After We Knew It:

Empirical Study and Modeling of Cost-effectiveness of Exploiting Prevalent Known Vulnerabilities Across IaaS Cloud

Su Zhang
Kansas State University
zhangs84@ksu.edu

Xinwen Zhang
Samsung Research America
xinwen1.z@samsung.com

Xinming Ou
Kansas State University
xou@ksu.edu

ABSTRACT

Infrastructure as a Service (IaaS) cloud has been attracting more and more customers as it provides the highest level of flexibility by offering configurable virtual machines (VMs) and computing infrastructures. Public VM images are usually available for customers to customize and launch. However, the 1 to N mapping between VM images and running instances in IaaS makes vulnerabilities propagate rapidly across the entire public cloud. Besides, IaaS cloud naturally comes with a larger and more stable attack surface and more concentrated target resources than traditional surroundings. In this paper, we first identify the threat of exploiting prevalent vulnerabilities [1] over public IaaS cloud with an empirical study in Amazon EC2. We find that attackers can compromise a considerable number of VMs with trivial cost. We then do a qualitative cost-effectiveness analysis of this threat. Our main result is a two-fold observation: in IaaS cloud, exploiting prevalent vulnerabilities is much more cost-effective than traditional in-house computing environment, therefore attackers have stronger incentive; Fortunately, on the other hand, cloud defenders (cloud providers and customers) also have much lower cost-loss ratio than in traditional environment, therefore they can be more effective for defending attacks. We then build a game-theoretic model and conduct a risk-gain analysis to compare exploiting and patching strategies under cloud and traditional computing environments. Our modeling indicates that under cloud environment, both attack and defense become less cost-effective as time goes by, and the earlier actioner can be more rewarding. We propose countermeasures against such threat in order to bridge the gap between current security situation and defending mechanisms. To our best knowledge, we are the first to analyze and model the threat with prevalent known-vulnerabilities in public cloud.

[1] in our experiments, we treat vulnerabilities with 30% or higher prevalence as prevalent vulnerabilities

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

http://dx.doi.org/10.1145/2590296.2590300.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*Statistical methods*; K.6.1 [**Management of Computing and Information Systems**]: Project and People Management—*Strategic information systems planning*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection ;

General Terms

Security, Measurement, Management

Keywords

Cloud Computing, Vulnerability Management, Game Theory, Virtual Machine Images, Patching Management

1. INTRODUCTION

Public cloud delivers computing resources with service-oriented, multi-tenant, and pay-as-you-go manner. According to the forms of offered resources, cloud computing can be Software as a Service (SaaS) such as Google Apps which provide individual applications to cloud customers (or users), Platform as a Service (PaaS) such as Microsoft Azure which offers a platform with a set of pre-configured software and programming environment, and Infrastructure as a Service (IaaS) such as Amazon Web Services (AWS) which allows users to run a number of virtual machines (VMs). In general, users in IaaS have the highest level of flexibility, e.g., to deploy their own infrastructures, systems, and applications according to their business requirements, by completely controlling and customizing their VMs. At the same time, IaaS users have more responsibility to secure their infrastructures and systems [1, 2].

Problem: In this paper, we do a comprehensive analysis on threats from VM images used in public cloud, based on an empirical study of known vulnerabilities in Amazon Machine Images (AMIs), which are VM images running on AWS Elastic Compute Cloud (EC2). Typically in an IaaS cloud, users can choose either private images (uploaded or customized by themselves) or public images (uploaded by others) to run their VMs. The public images could be uploaded by various types of publishers, including IT companies, open-source communities, and individuals. AWS is currently leading the market among IaaS providers. There are more than 6,000 public images published on EC2.

Security issues of public images in AWS have been reported in previous research work [1, 3]. Bugiel et al. [3] scanned a number of public images and found out that image publishers may leave unwanted information (e.g. passwords, keys, and other credentials) in their images and form backdoors in the cloud. They proposed several operational solutions to these issues. Balduzzi et al. [1] did

a similar but more comprehensive experiments on EC2 by scanning a larger number of images and identified more security issues including software vulnerabilities. However, these work do not provide any further analysis and modeling of attack and defense in cloud environment, which have very different cost-effectiveness properties compared with traditional computing environment according to our study in this paper. Although risk assessment approaches [4, 5, 6, 7, 8, 9, 10, 11] have been largely applied over traditional network surroundings, both Grobauer et al. [12] and Shrobe [13] have pointed out that prevalent vulnerabilities should be considered as a cloud-specific threat, under the homogeneous system environments in public cloud. However, they do not systematically analyze the impact of exploiting prevalent vulnerabilities in the cloud.

Even though security bulletins have been setup by Amazon to notify users about vulnerability information, previous experience has told us that significant effort is needed to bridge the gap between the provided service and current security situation. Specifically, we find that Amazon security bulletin usually releases critical vulnerability information more than two weeks later than original release date, e.g., by software vendors or community (cf. Section 2 for our study result). The exploit window could be even longer since there is no guarantee that every cloud user will and will be able to apply the update with the release, even though he is notified. Also, a cloud provider may not be able to identify all known vulnerabilities on its platform. For known vulnerabilities, this attack window is way longer than it should be [14]. Besides, the prevalence of individual vulnerabilities has not been considered when publishing security bulletins. For example, Amazon only use CVSS score [15] to indicate the severity of vulnerabilities, which is indicative for individual vulnerabilities on traditional in-house servers. However, threat from the prevalence of individual vulnerabilities should be re-evaluated under cloud environment. A prevalent image with known vulnerabilities can be instantiated by a large number of users in cloud, therefore it may generate large number of security holes for attackers. Attackers can do penetration test over public images, from where they can identify prevalent known vulnerabilities of running VMs and launch the same attack repeatedly to different instances. If the prevalent vulnerabilities indeed spread over the cloud, the attacker obtains an ideal cost-effective vehicle by exploiting the vulnerabilities to a large number of VMs. Therefore, with the new computing model of public cloud, it is easier for attackers to launch attacks through prevalent vulnerabilities.

On the other side, cloud also provides an ideal venue to deploy defense mechanisms in large-scale. For example, with the homogeneous cloud environment, automatic patching becomes more efficient than in traditional in-house environment. A number of patching frameworks have been proposed towards known security holes in cloud [16, 17]. However, there is no empirical study and analysis on the cost and gain effectiveness of defending in cloud.

Contributions: Consider these two aspects, we believe that public IaaS cloud introduces a completely new venue to consider attack and defense strategies, to maximum each side's benefit with minimum costs. For the first in this line of research, we empirically analyze the cost and effectiveness for exploiting known vulnerabilities under two different environments (traditional in-house and public IaaS cloud). We take AWS in our study since we can find more publicly available information than other IaaS providers. We first identify with real data analysis that prevalent known vulnerabilities are very common in AWS AMIs, and demonstrate with real penetrations test that attack with these vulnerabilities is very trivial by malicious cloud users. We then statically analyze that both attack and patch are more cost-effective in cloud than under traditional environment. By statically we mean our analysis is over one time spot. To further investigate the relationship and strategy of attackers and defenders in cloud environment, we map these scenarios into a two-player game theoretic model. Our model indicates that the current security of public cloud needs significant improvement. We then construct risk-gain analysis to simulate the evolution of the cost-effectiveness from defenders and attackers under different circumstances. Our results show that cloud defender should be more responsive and proactive when hardening cloud platform as the attack surface increases dramatically compared to traditional computing environment. Moreover, our model illustrates that both attack and defense are more time-sensitive in cloud as they become less cost-effective as time goes by. We then propose countermeasures according to the evaluation results. Figure 1 summarizes our contributions in this paper.

Roadmap: Section 2 states our approach and finds of identifying security vulnerabilities and threats from prevalent vulnerabilities in AWS. We conduct cost-effectiveness analysis for both attackers and defenders in Section 3. We construct a game theoretical model and a risk-gain analysis in Section 4. In Section 5 we provide countermeasures based on the results of our model. We discuss several limitations of our modeling in Section 6. We present related work in Section 7 and conclude this paper in Section 8.

2. EMPIRICAL STUDY: METHODOLOGY AND FINDS

2.1 Background

Amazon EC2 We do our experiments on public images over Amazon EC2. As a leading IaaS cloud provider, Amazon EC2 provides a platform by allowing different principals sharing their images publicly. Open source organizations like BitNami [2] and Ubuntu, IT companies such as Oracle and Amazon itself, and arbitrary number of individual contributors have published over 6,000 public images. Like potential attackers, we do penetration test over these images by launching corresponding VMs in order to analyze the weakness of running instances in the cloud.

Nessus Vulnerability Scanner Nessus [3] is a commercial vulnerability scanner developed from an open source product. It checks against configuration settings of a host and outputs a detailed report including security vulnerabilities, warnings, and system information, which can be from 50 to hundreds of pages. Therefore it is usually difficult for cloud administrators and users to read reports one by one in order to understand all security details in the cloud.

National Vulnerability Database (NVD) NVD [4] is an open database maintained by National Institute of Standards and Technology (NIST), which is regarded as one of the most comprehensive open vulnerability databases. Each entry in NVD is indexed by a Common Vulnerability Exposure(CVE), which is associated severity base score with a set of characteristics for that vulnerability. The base score is called "Common Vulnerability Scoring System (CVSS) base score" ranging from 0 to 10. The score indicates the overall severity of the vulnerability (the higher the worse).

2.2 Methodology

Penetration test over public images is a straightforward approach to identify prevalent vulnerabilities. Figure 2 illustrates our overall methodology. When scanning available public images on Amazon

[2] http://bitnami.org/
[3] http://www.tenable.com/products/nessus
[4] http://nvd.nist.gov/

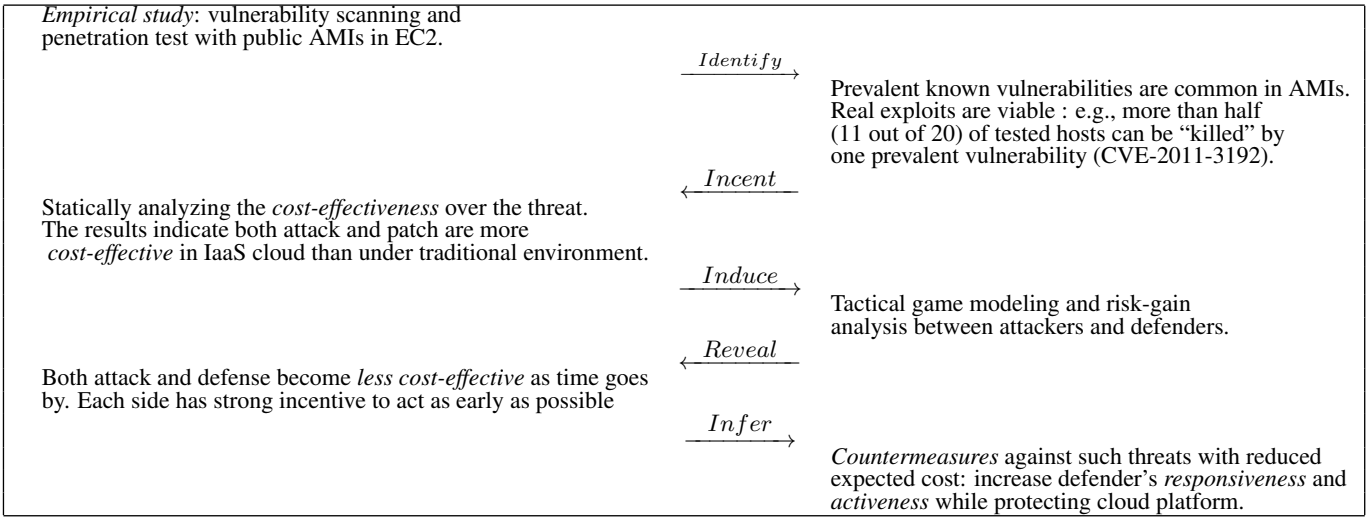


Figure 1: Contribution map of this paper.

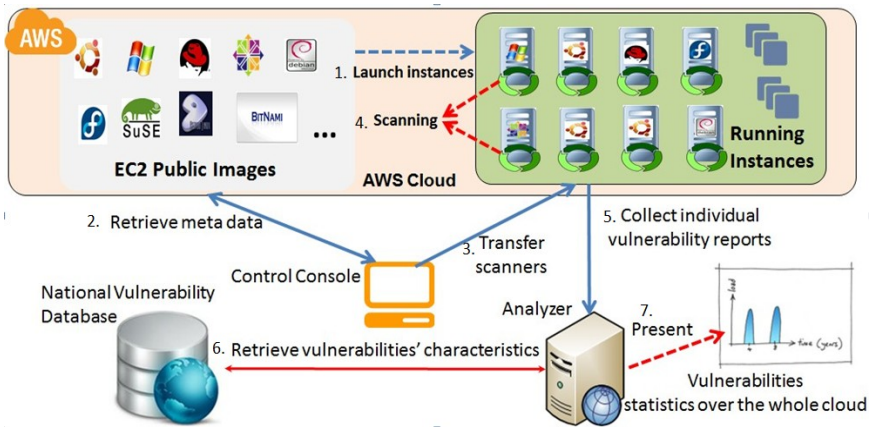


Figure 2: Methodology of our empirical study.

EC2, we first select a number of representative images to investigate, then launch one instance for each selected image. We then adopt a dedicated scanning server to transfer Nessus vulnerability scanner to each instance, and start scanning by running our script on each target instance [5]. After the scanning is complete, our script transfers all scanning reports to the scanning server. We retrieve the characteristics of each vulnerability by looking it up at the NVD. Based on the distribution of vulnerabilities and their characteristics, we obtain a single vulnerability report of all launched instances.

We launch and scan 80 public images in EC2. The selection of these images is based on the distribution of the operating system (OS) types and versions of public AMIs, with the assumption of the similar distribution of launched VM instances in the cloud.

2.3 What We Find

A considerable amount of prevalent vulnerabilities exist in AMIs. Similarly to what other researchers have found [3, 1], our scanning reveals a large number of vulnerabilities existing in public AMIs. Besides, we have identified several prevalent ones among all of these detected vulnerabilities. Table 1 lists the top prevalent vulnerabilities from our scanning. The prevalence indicates the probability of the vulnerability's existence among all images we have scanned. We find out that most (8 out 9) of them are critical vulnerabilities (with a CVSS score 7-10) by NVD standard, most (8 out of 9) of them can be accessed remotely, most (8 out 9) of them can be easily accessed, and most of them (7 out of 9) can be utilized by attackers to crush corresponding applications completely.

Attackers can identify prevalent vulnerabilities without scanning individual VM instances. Amazon EC2 allows users to select public images based on platforms (OS types, versions, and pre-installed applications). Figure 3 shows the public images distribution based on OS types [6]. As we can see, more than half of the images are Ubuntu based. A closer look into the Ubuntu images indicates that more than half of them are either 10.04 or 12.04. Therefore under this circumstance, an attacker can keep monitoring newly released vulnerabilities affecting these prevalent OSes and application frameworks. The attacker can also leverage known vulnerabilities that have not been patched by the publishers of the

[5]Thanks for Amazon's approval for our scanning and penetration tests

[6]The data was collected in September 2012, which may change with new releases of AMIs.

Table 1: Windows between Original Release and Amazon Announcement of Prevalent Known Vulnerabilities

CVE	CVSS Base Score	Prevalence	Original Release	Amazon Announce	Attack Window in (days)
CVE-2012-4244	7.8	0.59	09/14/2012	09/28/2012	14
CVE-2012-3955	7.1	0.58	09/14/2012	N/A	> 26
CVE-2012-3817	7.8	0.52	07/25/2012	08/07/2012	13
CVE-2012-2807	10	0.49	09/07/2012	N/A	> 33
CVE-2012-2337	7.2	0.46	05/18/2012	07/30/2012	73
CVE-2011-3102	10	0.45	05/16/2012	N/A	> 117
CVE-2012-1033	5.0	0.45	02/08/2012	06/22/2012	135
CVE-2012-1667	8.5	0.45	06/05/2012	06/22/2012	17
CVE-2012-2110	7.5	0.34	04/19/2012	05/03/2012	15

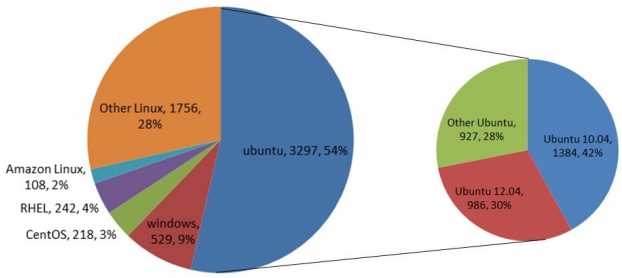
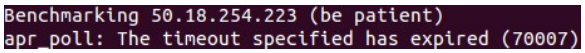


Figure 3: Public images distribution by OS in Amazon EC2.

100% of requests can be serviced before DoS attack.



The server stops responding after DoS attack.

Figure 4: Benchmarking results of a server before and after launching apache killer.

AMIs or the administrators of running instances, due to the patch window gap that we have observed in EC2 (explain shortly).

As a result, statistical analysis of OS and application distributions can help attackers in identifying the weaknesses and prevalent vulnerabilities in the cloud. This provides a scope of target victims and reduces the cost for large scale scanning and penetration. Attackers can roughly understand the overall potential weakness by simply noticing the latest vulnerabilities associated with the most prevalent OSes and applications installed in public images.

The patch window is long enough for attackers to exploit. We study several critical vulnerabilities and find that the gap between their original releases and Amazon's notifications is usually longer than two weeks (cf. Table 1). Attackers could easily launch 1-day exploit repeatedly in the entire cloud. The length of exploit window depends on the activities of cloud stakeholders (cloud provider and customers) such as the date of notification and their hardening and patching mechanisms. Moreover, not all known vulnerabilities can be easily detected by the cloud provider. As we have noticed, a large amount of exploitable vulnerabilities have not been notified by Amazon after a long time of their original releases. Therefore, attackers have enough time to prepare and launch attacks. Even worse, existing study has shown that more than 40% of small companies (under $50M revenue) do not have patch management [18] deployed in cloud, which consists of a considerable amount of current IaaS customers [19].

Running VMs in IaaS cloud offers more stable attack surfaces. VMs in IaaS cloud are more stable than traditional endpoints from an attacker' perspective. First of all, the IP range of each cloud provider is stable and can be predicted easily. Attackers could identify the location of their target VMs by playing several tricks [20]. Besides, a vast number of EC2 users are service providers with high availability requirement [19]. Therefore their applications and port configurations are relatively easy to detect. Attackers could reuse configuration information obtained previously to launch large scale attacks afterwards (for new vulnerabilities on the same or similar applications and systems). However, this does not work well under traditional in-house environments since the IP addresses of end hosts are changing more frequently, and most in-house servers are behind firewalls, and it is much more costly for an attacker to launch large scale attacks in order to locate a large number of victims under such heterogeneous environment.

2.4 Case Study: Penetration Testing on VMs in EC2

To confirm the viability of exploiting with prevalent vulnerabilities in EC2, we conduct a penetration test towards running VMs launched from vulnerable AMIs upon Amazon's approval. We first identify a prevalent vulnerability CVE-2011-3192, which is referred as "Apache Killer". We note that this vulnerability was not detected by Nessus in our scanning but it exists in 11 out of 20 AMIs [7] that we investigated with Ubuntu 10.04, most of which have been published for more than one year. Surprisingly, no security advisory on Amazon has been published for this vulnerability. We simply launch another instance in EC2 as an attacker with Metasploit [21] installed. By following the online instruction, we simply setup Metasploit with the number of packets sent to the target VM for DoS attack. We successfully crashed the Apache server running on all of the target VMs by sending 400 packets. The attack can be defended by running one line command (`sudo apt-get update`) to patch the vulnerability.

In order to verify the DoS attack, we use ApacheBench to test the response of the target server. As shown in Figure 4, before the exploit, all requests are served by the server; while after the attack, the ApacheBench could not receive any response, indicating the server is completely crashed.

We observe that the attack is very easy to launch with little interaction from the attacker. Therefore, crushing a large number of

[7] For safety reason we omit the AMI IDs here.

web service hosts is trivial from the attacker's perspective if any one of these vulnerable AMIs is widely used.

3. STATIC COST-EFFECTIVENESS ANALYSES

Our empirical study has demonstrated that homogeneous settings in popular public cloud not only enhance the efficiency of computing power, but also bring new economic considerations for both attackers and defenders. Towards a first study on this, we do a comprehensive cost-effectiveness analysis by comparing exploiting prevalent vulnerabilities in public IaaS cloud and traditional in-house computing environments. While we refer a single *attacker* in both cases, a *defender* refers to service owners in traditional case and all cloud stakeholders (both cloud platform provider and cloud customers) in IaaS.

Assumptions: Our analysis is based on the assumption that VM images are publicly available and used by cloud customers, but we do not require either each image or certain percentage of images are instantiated in the cloud. We further assume that prevalent types of images (OS types, versions, and application frameworks) are also prevalent in the VMs of the cloud.

Results: Our analysis reveals that *both attack and defense are more cost-effective in cloud than in traditional in-house environment*. Attack surface under cloud environment has been enlarged with an increased density of potential victims. Moreover, attack cost has been decreased in cloud because the homogeneous nature of public cloud platforms reduces the effort required for target locating and vulnerability reconnaissance. On the other hand, cloud stakeholders (providers and customers) can manage patch with batch processing, which can patch larger attack surface per unit time than that in traditional environment.

3.1 Cost-effectiveness Analysis for Attacker

3.1.1 Cost of Attacker

A cyber attack usually involves the following costs [22, 13]: (1) locating target victims, (2) identifying vulnerabilities of victims, (3) choosing vulnerabilities , (4) obtaining exploits, and (5) dealing with defense mechanisms. For target victims in the cloud and traditional in-house environment, the costs (3) and (4) are the same. Therefore our analysis focuses on (1), (2), and (5), and our results indicate that IaaS cloud provides dramatically lower costs for attackers in these aspects.

Identifying victims. Under traditional environment, attackers could obtain target IP addresses in a straightforward way (e.g. by looking up DNS server). However, the external firewall deployed by most in-house servers may make the IP addresses untraceable. For certain types of threats like botnet or non-targeted DoS attacks by cyber terrorists, continuous (in terms of IP address) nodes with weak defending mechanisms but stable and high bandwidth are on the top of their target list. Consider that most bots in popular botnets such as "Conficker" have small bandwidth only [23], we believe high quality bots in cloud are very appealing and can significantly increase the competitive strength of a bot master in botnet market, thus give strong incentive for attackers.

Consider a botnet master that needs to harvest N bots with a certain vulnerability v. Assuming for each reachable host, the probability of having v is ρ_v. Ideally, the search space of the vulnerable hosts under traditional environment is the whole IP address space (3,706,452,992), e.g., by generating random target IP addresses to exploit. Consider the factors that not every IP is assigned a host,

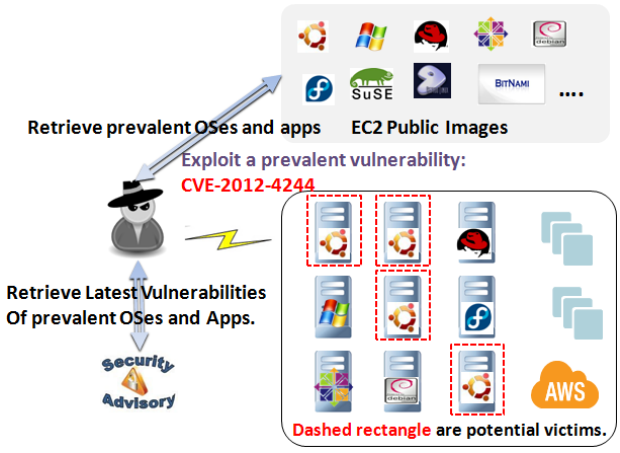


Figure 5: Attacks under IaaS cloud

and not each host is accessible, let δ_i be the probability that a single IP address is reachable in the Internet. Therefore the attacker needs to have at least $N/\rho_v\delta_i$ tries. However, under public IaaS cloud environment, the exploring range is significantly shrunk as the cloud provider offers the location and IP range publicly. For EC2, the total IP addresses is around 1,500,000 [24]. Besides, most of these IPs are located in a centralized manner as the IP addresses of VMs on the same data center are usually assigned continuously [20]. With the high density of VMs running in a single data center, launching exploit to the cloud usually has much higher hit ratio δ_c. Therefore the attacker needs $N/\rho_v\delta_c$, where $\delta_c \gg \delta_i$, which indicates that the attacker needs dramatically less cost in cloud.

Identifying vulnerabilities. Under traditional environment, if the attacker wants to utilize known vulnerabilities to exploit a host, he may have to scan over the target machine, which can be easily blocked by firewalls. Researchers have proposed several passive scanning approaches in order to bypass IDS or firewall [25, 22], which may lower the scanning cost but still take a considerable amount of time and rely on some other assumptions (e.g., host administrators never modify packet headers).

On the other side, this vulnerability scanning cost can be reduced dramatically in public cloud environment (cf. Figure 5). As shown in our study in Amazon EC2, attackers could obtain the information of VM images (OS and applications installed) by browsing public image description pages. A brute force scanning on all images can help the attacker to decide the distributions of systems and applications in VMs, although in a rough manner. This information can reduce the cost to identify existing vulnerabilities of VMs running in the cloud. Furthermore, the attacker can keep tracking newly-released vulnerabilities associated with these prevalent OSes or applications in public images. Once a new vulnerability is released, it may exist on a large number of VMs in the cloud. Consider the usual patching window gap that we have observed in the last section, the attacker has plenty of time to develop and launch exploits, e.g., to harvest bots with vulnerable VMs. Therefore, identifying known vulnerabilities over the cloud is dramatically faster than that under traditional environment.

Dealing with hardening mechanisms of hosts. Customers on IaaS cloud usually have limited hardening support from the cloud provider, e.g,. Amazon EC2 only provides each instance an external firewall called security group, but no patching management. At

the same time, a large number of cloud customers are small-sized service providers [18], and usually do not have strong motivation of hardening their systems as large companies. This results in a weak link for the cloud provider. Once an attacker has managed to exploit a prevalent vulnerability among the VMs of these small companies, a large scale of attack can result in loss for both the cloud customers and the cloud provider.

However, under traditional environment, an enterprise level service provider usually has dedicated team to maintain their platforms, which are usually hardened with several layers of firewalls in order to protect their data and infrastructure. The in-depth defense mechanisms increase the difficulty level for an attacker to compromise the server. It is extremely hard for an attacker to compromise a large number of hosts at the same time.

Therefore, we conjecture that compromising or bypassing hardening systems costs less under public cloud than that in traditional environment. Consider the cloud provider as a special service provider. Since it provides high flexibility of customizing infrastructure to its customers, its own defense mechanism is less tightly controlled compared to traditional in-house service providers, which makes it much easier to penetrate.

3.1.2 Gains of Attacker

An attacker could access confidential information for social or commercial benefits. Besides, the attacker could gain from the loss of his competitors by disrupting or disabling their services. These gains are the same under both cloud and traditional environment. One cloud specific gain is that upon compromising, high-quality bots on the cloud are denser than that in traditional computing environment with higher bandwidth and availability, which makes cyber terrorists easier to identify their targets.

3.1.3 Summary of Cost-effectiveness for Attacker

Considering similar gains of compromising a fixed set of hosts, the cost of the attacker is lowered by launching large scale attacks in an IaaS cloud, with lower costs in identifying enough number of vulnerable hosts, identifying exploiting vulnerabilities, and dealing with hardening mechanisms. Furthermore, exploiting prevalent vulnerabilities in the cloud usually brings the attacker more competing benefits with higher quality of bots than exploiting targets individually under traditional environment. Therefore, *the cost-effectiveness ratio for an attacker is lower in public cloud than that in traditional computing environment; that is, it is more economically efficient for an attacker to launch attacks in cloud.*

3.2 Cost-effectiveness Analysis for Defender

We refer the single term defender as all stakeholders that benefit from defending attacks, including the cloud provider and all of its customers. While facing attacks, the visible cost paid by the defender is the hardening cost, and the gain is the loss of being exploited by attackers, or the commercial benefits from the services that otherwise are disrupted or disabled by attacks.

3.2.1 Costs for Defender

Hardening cost against known vulnerabilities is mainly from patching [26]. The cost per unit by patching in-house hosts is more pricey than batch patching over the cloud, since the batch processing lowers the hardening cost in cloud than in house servers [16].

3.2.2 Loss (or Gains) of Defender

Avoiding potential exploit effectiveness is the gain from the cloud provider's perspective. Exploiting effectiveness has a considerable overlap with an attacker's potential gains. Specifically, classical losses including that of service availability, data integrity, and confidentiality are the same for the defender in both cloud and traditional environment. Most of these losses are transferred to the attacker's benefit. However, there are cloud specific losses caused by large scale attacks, including neighborhood loss, user reputation loss for services, cloud provider reputation loss, and cloud utility misuse.

Neighborhood loss. As aforementioned, an attacker can lookup the IP range of a cloud provider's data center easily. The attacker could rent a VM and launch a large scale exploits to the VMs in the same data center. The attacker does not need to know the exact IP address of his target. Instead, all VMs on the same data center with the same vulnerability can be exploited. This expanded attack surface causes exponentially higher loss than that in traditional computing environment.

Reputation loss for cloud customers. Cloud customers usually are web service providers, and can lose their reputation from their own users upon being compromised. Even though this type of loss is invisible and indirect, it may completely affect the end users' confidence in continuing their services. Threats from prevalent vulnerabilities enlarge such fears as a large number of services on the same cloud platform may exist.

Reputation loss of cloud provider. Even worse than user's reputation loss for cloud customers, the cloud provider's reputation can dramatically drop given a considerable amount of their VMs are compromised. Typically, the healthy including safety level of a cloud provider impacts the number of its users. A customer based survey [27] indicates that a cloud provider's reputation is the most important factor when a customer chooses which provider to go with.

Cloud utility misuse. Once an attacker has managed to deploy bots on one type of VMs in public cloud, he potentially could create a botnet with a large number of machines, which can be powerful enough for crushing other services over the Internet. This further enlarges the cloud provider's reputation loss. As for monetary loss, existing study has pointed out that a DDoS attack could cause up to $19M/hour loss for availability-sensitive services like E-banking. For each DDoS attack, the cost can be up to $100M [28].

3.2.3 Summary of Cost-effectiveness for Defender

The cloud provider can patch prevalent vulnerabilities with a cheaper unit cost than patching in-house servers individually. At the same time, the effectiveness of exploiting prevalent vulnerabilities in IaaS cloud is exponentially higher than the same attacks under traditional environment, consider much denser potential victims with the same vulnerabilities in cloud. Furthermore, the defender has extra cloud-specific losses such as cloud provider's reputation loss and cloud utility misuses. Therefore, our conclusion is that *the cloud defender has much lower cost-effectiveness ratio than in traditional computing environment, which indicates that with the same cost spent by the defender, he achieves more economic benefit in cloud.*

4. TACTICAL GAME MODELING BETWEEN ATTACKER AND DEFENDER

Above cost-effectiveness analysis statically considers the costs and gains for both attackers and defenders. However, in real world several factors impact the relative costs and benefits of each side, and thus both rational attackers and defenders adjust their behaviors by considering these dynamic factors to achieve maximum benefits. Among these, the time-since-release has been considered as

one of the main affecting factors that impacts the effectiveness of exploiting known vulnerabilities. This comes from an assumption that more VMs are patched for a given vulnerability as time goes by. Therefore, the sooner the attacker acts, the larger number of victim hosts can be hit with the same cost. On the other side, the sooner the defender acts, he can patch more VMs thus prevent more loss with lower cost. Moreover, patching a more prevalent vulnerability (by means of the vulnerability distribution in images and VMs) results in more cost-effectiveness ratio for both the attacker and defender, since it costs more for the attacker to identify vulnerable victims, and brings less gain for the defender to patch the vulnerability.

Therefore, we believe the dynamic cost-effectiveness ratios result in a game-based tactics between the attacker and defender. In this section, we construct a game theoretic model in order to illustrate the actions that rational attackers and defenders should take. We further map different cost-effectiveness scenarios into cost density functions to show their evolutions. Our model indicates that both the attacker and defender have stronger incentive to act earlier, and their actions become less cost-effective as time goes by. After certain moment, the defender only needs to maintain the security level (the prevalence of the vulnerability) as the patching cost may exceed the cost from residual risk. The attacker may also lose the motivation of launching further attacks after certain point as the attack gain may not be able to compensate the attack cost due to the drop of the vulnerability prevalence. Therefore the threat from prevalent vulnerabilities can be greatly mitigated as long as the defender patches security holes in a timely and proactive manner. However, cloud customers should be advised to protect their systems against targeted attacks as this is not a cloud specific threat.

4.1 Game Theory Background

An N-player game can be represented as a function G ($S_1, S_2, \ldots, S_N, u_1, u_2, \ldots, u_N$), where S_i $(0 < i < N)$ is a strategy set $(s_{i1}, \ldots, s_{im})$ for player i, and s_j $(s_j \in S_i)$ is a complete strategy available for player i. Player i has a probability distribution $P_i = (p_{i1}, \ldots, p_{im})$, where p_{ik} is the probability of s_{ik} being adopted by player i. The payoff for player i is $u_i(S_1, \ldots, S_n)$ $(1 < j < n)$, where S_j is the strategy adopted by user j. For an N-player game theory, the expected payoff for player i is:

$$v_i(p_1, \ldots, p_n) = \sum_{m_1=1}^{M_1} \cdots \sum_{m_n=1}^{M_n} \left[\prod_{k=1}^{n} P_{km_k} \right] u_i(S_{1m_1}, \ldots, S_{nm_n}), \quad (1)$$

where M_j is the pure strategy numbers available to player j. For a 2-player game, the expected payoff of player 1 is:

$$v_1(p_1, p_2) = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} P_{1m_1} P_{2m_2} u_1(S_{1m_1}, S_{2m_2}), \quad (2)$$

where p_1 and p_2 are two sets of probability distributions adopted by the two players, respectively. Each distribution consists of a number of probabilities (sum up to 1), each of which indicates the chance of a strategy adopted by the player. S_{1m1} and S_{2m2} represent the strategies adopted by p_1 and p_2, respectively.

4.2 Game Theory Modeling

We consider player 1 as the attacker and player 2 as the defender. Player 1 has two strategies: attack (S_{11}) or stay idle (S_{12}). Player 2 also has two strategies: patching (S_{21}) or stay unpatched (S_{22}). P_{ij} indicates the probability of S_{ij} being adopted. We say that K_1 is a proactive action adopted by each player, meaning attack and patch for the attacker and the defender, respectively. K_2 means a passive action: stay idle for the attacker and leave the platform unpatched for the defender. Given each of the two players has two possible strategies, there are four conditions as follows.

- Both players choose K_1. The cloud defender needs to pay cost $(-CP)$ in order to patch his platform in a timely manner. On the other side, the attacker has to pay the cost $(-AC)$ of exploiting but without gaining from the hardened platform.
- When both players choose K_2, obviously both get 0.
- When the attacker chooses K_1 and the defender chooses K_2, the attacker gains $(+AG)$ from exploiting by paying attack cost $(-AC)$. The defender suffers the cost of being exploited $(-CD)$.
- When the attacker chooses K_2 and the defender chooses K_1, the attacker gets 0 and the defender pays patch cost $(-CP)$ to keep the platform up-to-date.

We use P_S and P_A to denote the probability of being proactive for the defender and the attacker, respectively. Given the four possible conditions, their expected payoffs (V_A and V_S) in the game are:

$$\begin{aligned} V_A &= -AC \times P_A P_S + 0 \times (1 - P_A) \times (1 - P_S) \\ &\quad + AG \times P_A \times (1 - P_S) + 0 \times (1 - P_A) \times P_S \\ &= AG \times P_A \times (1 - P_S) - AC \times P_A P_S \end{aligned} \quad (3)$$

$$\begin{aligned} V_S &= -CP \times P_A P_S + 0 \times (1 - P_A) \times (1 - P_S) \\ &\quad - CD \times P_A \times (1 - P_S) - CP \times (1 - P_A) \times P_S \\ &= -CD \times P_A \times (1 - P_S) - CP \times P_S \end{aligned} \quad (4)$$

The equations indicate that the expected payoffs of both players depend on both of their determinations of being proactive. Without exploiting intention, the attacker does not gain anything. When being more aggressive, he has an increased potential gain (when facing an unconscious defender) with the cost of launching attacks. A passive defender may end up losing nothing given the attacker is passive as well. However, this assumption is unrealistic as cyber attacks are ubiquitous. The defender (especially under cloud environment) should have a reasonable expectation on the density of attacks per unit time in order to balance the tradeoff between hardening cost and risk properly. Visualizing the game between the attacker and the defender can assist cloud stakeholders to better understand current security situation and make hardening plans accordingly.

4.3 Tactical Modeling between Attacker and Defender

We consider the events of instantiating images by different customers in an IaaS cloud are independent, and the instantiation rate is a relatively stable value given the large number of customers. Therefore the instantiation of images with each prevalent vulnerability can be modeled with an exponential distribution, and the probability density function (PDF) can be expressed in Equation 5, where t is time and λ is the arrival rate of instantiation events in the cloud. A larger λ means a denser event and higher risk density of the vulnerability. Therefore, the prevalence of the vulnerability determines the value of λ, and the PDF can be regarded as a risk density function. The risk density keeps decreasing as time goes by. This is because less and less vulnerable targets available (either

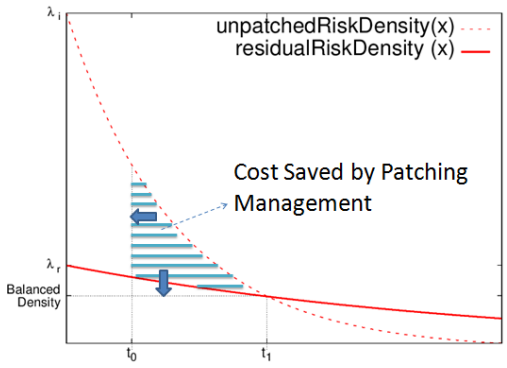


Figure 6: Cost density distribution for cloud defender.

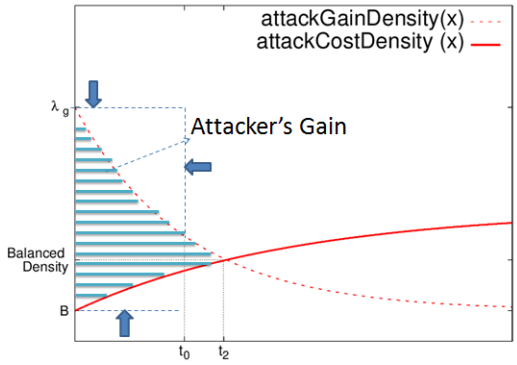


Figure 7: Cost and gain density distribution for cloud attacker.

patched by the VM users or already exploited by the attacker) to the attacker.

$$f(t,\lambda) = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (5)$$

4.3.1 Tactical Modeling for Defender

Figure 6 illustrates the cost-effectiveness from the defender's perspective with two different strategies. A defender with little or no security awareness has a higher rate parameter (risk density) than a defender with appropriate patching management. The value of t_0 indicates the exploit window of a prevalent vulnerability. Therefore starting from t_0, the defender can choose to deploy patch to the vulnerability. Then the risk density is dropped to the patched risk density curve. A rational defender has a lower risk density because the sum of residual risk and patch cost is regarded lower than the unpatched risk cost density. At the moment t_1, the two curves have a point of intersection. Starting from t_1, the defender only needs to maintain the security level. This is because the patch cost density has exceeded the residual risk density[8]. The value of t_1 is decided by the two rate parameters with Equation 6.

$$\begin{aligned} \lambda_1 e^{-\lambda_1 t_1} &= \lambda_2 e^{-\lambda_2 t_1} \\ t_1 &= \frac{\ln^{\lambda_i} - \ln^{\lambda_r}}{\lambda_i - \lambda_r} \end{aligned} \quad (6)$$

Therefore, the risk that could possibly be reduced by a rational defender can be expressed with Equation 7 and is marked in Figure 6, where λ_i and λ_r are risk density rates of before and after the patch has been deployed. As indicated in Figure 6, minimizing the value of t_0 and rate parameter (λ_2) of residual risk density can maximally reduce the risk.

$$\begin{aligned} R(\lambda_i, \lambda_r) &= \int_{t_0}^{t_1} \lambda_i e^{-\lambda_i t} dt - \int_{t_0}^{t_1} \lambda_r e^{-\lambda_r t} dt \\ &= (1 - e^{-\lambda_i t})\Big|_{t_0}^{t_1} - (1 - e^{-\lambda_r t})\Big|_{t_0}^{t_1} \end{aligned} \quad (7)$$

4.3.2 Tactical Modeling for Attacker

The density of the attacker's potential gain through exploiting is similar as but slightly lower than the unpatched cloud loss due to being exploited, since some loss like neighborhood loss cannot be gained by the attacker. Therefore, the attacker's potential gain conforms to a PDF as well. The attack cost has both maximum and minimum values given a fixed number of to-be exploited targets. The maximum value is paid while brute-forcing over the whole cloud platform (if there are not enough hosts that can be compromised in the whole platform) and the minimum cost is paid while exploiting each target with minimum cost (each attack succeeds at its first attempt). The cost is negatively correlated to the exploiting effectiveness because exploiting vulnerabilities with higher density costs less than utilizing sparsely distributed security holes. Therefore, the attack cost can be mapped to a variant of a cumulative probability function (CPF) shown in Equation 8. The only difference between the attack cost model and CPF is the constant B, which represents the basic cost of each attack, e.g., target reconnaissance and vulnerability detection. λ_e refers to the cost density of failed attack attempts.

$$F(t,\lambda_e) = \begin{cases} 1 - e^{-\lambda_e t} + B & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (8)$$

Figure 7 indicates the attacker's cost-effectiveness as time goes by. Consider a defender with strong security awareness which deploys patch at t_0, the attacker's gain is then decreased dramatically. If the defender is unconscious, attacks could last until t_2 as the gain through attacks after t_2 cannot compensate the attacker's cost of launching these exploits. Equations 9 indicates the attacker's gain by exploiting unpatched ($t_p = t_2$) and well hardened ($t_p = t_0$) platform, respectively. λ_g means the density of attacker's gain and λ_c means the attack cost density. In general the attacker loses the motivation of attacks after t_0 or t_2 whichever comes first. t_2 can be calculated through solving Equation 10. Given t_0 and t_2, the attacker's gain reduced by a rational defender can be obtained through Equations 9 and is the marked area in Figure 7.

$$\begin{aligned} G(\lambda_g, \lambda_c) &= \int_0^{t_p} \lambda_g e^{-\lambda_g t} dt - \int_0^{t_p} (1 - e^{-\lambda_c t}) dt - t_p \times B \\ &= (1 - e^{-\lambda_g t})\Big|_0^{t_p} - (t + \frac{1}{\lambda_c})\Big|_0^{t_p} \\ t_p &= \begin{cases} t_2 & t_0 \geq t_2 \\ t_0 & t_0 < t_2 \end{cases} \end{aligned} \quad (9)$$

$$\lambda e^{-\lambda_g t_2} = (1 - e^{-\lambda_c t_2}) \quad (10)$$

4.3.3 Two-player Game

The game between the attacker and defender can be seen in Figure 8. At the starting point, both attack and defense are impacting. However, both of them become less cost-effective as time goes

[8] residual risk density refers to the risk density left after the patch has been deployed

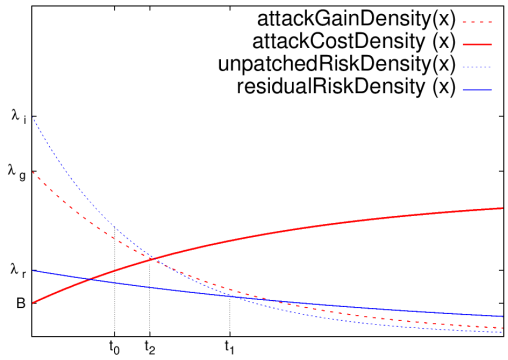


Figure 8: Game between attacker and defender in cloud.

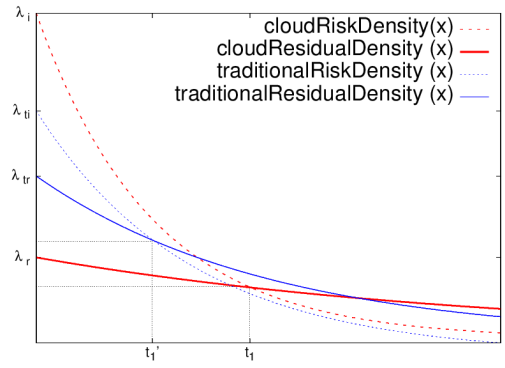


Figure 9: Cost-effectiveness comparison between traditional and cloud environments.

by. There are three noteworthy moments. t_1 is the moment when the attacker's cost and gain get balanced, after that a rational attacker stops launching attacks. t_2 is the moment when the cloud defender's cost is minimized and further patching costs higher than the residual risk. t_1 is not necessarily less than t_2. When $t_1 < t_2$, a rational attacker stops attacking but the cloud defender continues patching as the expected potential loss is greater than patching cost, and the cloud defender stops hardening its platform until t_2. When $t_1 \geq t_2$, the defender's cost and gain get balanced first, and he stops patching but maintains the security level. However, a rational attacker continues launching attacks under this circumstance as he can still obtain more than attack cost. When $t_1 \geq t_2$, we say there is a range for both the attacker and the defender can be satisfied. If $t_1 < t_2$, the attacker stops launching attacks as the expected gain cannot compensate attack cost. The defender only needs to maintain the security level under this circumstance. After t_0 the defender has patch available to the vulnerability. If the defender deploys patch, both t_1 and t_2 could arrive earlier up to t_0 because both rate parameters of the exploiting effective density and attacker gain density drop. Therefore, not only could an smaller t_0 reduce the attacker's gain and exploiting effectiveness, it could also end the game between attacker and defender earlier.

4.3.4 *Cost-effectiveness Comparison between Cloud and Traditional Computing Environment*

In order to compare attacks under cloud and traditional computing environment, and investigate how much risk can be reduced by rational defenders, we model the cost-effectiveness from both the attacker's and the defender's perspective. We use the risk density functions (Equations 7 and 9) to answer the following questions:

- How much risk can be reduced by a rational defender in the cloud?
- How much risk can be reduced by a rational defender in traditional in-house environment?
- How much more gain can the attacker obtain when facing an unconscious cloud defender than a rational defender?
- How much more gain can the attacker obtain when facing an unconscious defender than a rational defender in traditional environment?

In Section 3 we have analyzed that prevalent vulnerabilities lead to lower patching expense but higher potential loss in cloud than under traditional environment. Therefore under the cloud environment, the risk density is higher but the patch cost density is lower than those under traditional environment. Figure 9 illustrates the comparison. For cloud environment, the rate parameters before and after patch deployed are denoted by λ_i and λ_r. While λ_{ti} and λ_{tr} represent rate parameters under traditional environment (before and after patch deployed, respectively). As we can see, the defender in the cloud can achieve a better stable security level than in traditional because of the lower cost in patching. The time t_1' does not need to be greater than t_1 [9]. If the cloud defender can handle hardening work appropriately, it can achieve a lower security level with shorter period of time. Figure 9 also tells us that it is urgent for the cloud defender to harden the cloud platform as the gap between the potential loss and hardening cost is dramatically enlarged compared to that under traditional environment. Without patch deployed, the potential loss in the cloud is exponentially higher , i.e., $R(\lambda_i, \lambda_r) \gg R(\lambda_{ti}, \lambda_{tr})$.

4.4 Summary

Through dynamic cost-effectiveness analysis with gaming modeling, we have observed that both attack and defense are more cost-effective in the cloud, and they become less cost-effective as time goes by. Three factors determines the game between the two parties: the defender's willingness (P_S), responsiveness (t_0), and activeness (λ_r). In a nutshell, the cloud defender should be willing to harden its cloud platform in a timely and proactive manner.

5. COUNTERMEASURES

Our gaming modeling indicates that in order to reduce the risk with prevalent known vulnerabilities in cloud, which is the cost marked in Figure 6, two parameters (t_0 and the rate parameter of residual risk density) need to be minimized.Which means that the defender should be more responsive and proactive to known vulnerabilities. Minimizing t_0 means eliminating 1-day exploits, i.e., keeping all instances up-to-date and maintaining running instances without severe known security holes. As this is difficult to achieve, keeping all public images up-to-date may be more feasible. We propose the following three countermeasures for cloud providers and customers.

Patching public VM images. Public VM images like AMIs should be up-to-date when being launched by users. The cloud provider should setup policy in order to make sure unpatched images should not be launched. Two options are available in order to achieve this

[9] t_1 and t_1' represent the moment when the attacker's cost and gain get balanced under traditional and cloud environment respectively

requirement: the provider can either force a user to update the image before it is launched, or the provider or image publishers can update public images offline periodically [16]. This could reduce the boot time at user end during launching.

Maintaining running instances. Every time when launching an image, the image should be required to check against a configuration file in order to make sure all default apps are up to date. The configuration file can be provided by the image publisher. Cloud users can customize the configuration file on their own. Cloud users should also be responsible for the applications installed by themselves.

Give patching priority to prevalent vulnerabilities. As Figure 6 indicates, higher prevalence is more time sensitive. Therefore, considering the prevalence (along with impact factors) of vulnerabilities is needed when making patching plans. The moment (t_d) of deploying a patch to a specific vulnerability can be inferred from a preset threshold of tolerable risk. Given similar impact, a higher prevalent vulnerability usually has a smaller t_d.

Shuffling cloud infrastructure smartly. Introduce a new defensive mechanism, which could make the configuration of the cloud platform as an animation rather than a static picture. Similar to patching vulnerabilities periodically, configurations (e.g.IP address, topology or applications) can be changed time to time. This type of moving target defense paradigms [29, 30, 31] could significantly mitigate security holes on the cloud.

6. LIMITATIONS AND DISCUSSIONS

Our model has a number of limitations. First of all, we only consider prevalent vulnerabilities in large-scale cloud. We do not provide an aggregated metric indicating the security level of the whole cloud platform and individual VM systems. Cloud stakeholders can use our model to further calculate a metric for finer-grained modeling. The calculation can be based on environmental factor or impact factor of identified vulnerabilities.

Secondly, we do not have accurate statistics regarding cloud customers' system information of running VMs, e.g., exactly how many VMs are launched for a particular image, and how many users usually patch their VMs in timely manner. In our model we take a simple estimation by assuming that the prevalence of instances is similar to that of public images.

7. RELATED WORK

Security issues regarding the public images in Amazon EC2 have been studied. Sensitive information leak has been detected by Bugiel et al. [3]. They also suggested several solutions for various cloud specific threats. More comprehensive experiments over EC2 have been conducted by Balduzzi et al. [1]. They scanned a larger number of public AMIs and found more security issues like software vulnerabilities and malwares. However, neither of them evaluate potential threat from prevalent known software vulnerabilities, which we believe is the most straightforward and efficient way for attackers to intrude an IaaS cloud.

Game theory has been used for modeling attacks and defenses. Activities (exploiting and hardening) between attackers and defenders perfectly conform to a 2-player game. Yan et al. [32] model a game between DDoS attackers and defenders. Khirwadkar [33] constructs a game theoretic model between attackers and defenders by using Fictitious-Play approach in order to make sure the two parties are not under complete information environment. Game theory based analysis regarding network security is surveyed in [34].

Another line of work is to visualize economic incentive from defender's perspective [35, 14, 36, 37, 26, 28], which helps the defender appropriately allocate resources to security-related tasks. Studer et al. [35] evaluate DDoS attacks from both technical and economic points of view, and provide evaluation on monetary loss due to these attacks in addition to the economic appraisal by [28]. Frei et al. [37] visualize time lengths between vulnerability disclosure date, patch date, and exploit date, and believe these time periods represent the current status of security industry. However, they do not consider the date of patch deployment, which is captured in our model to evaluate the dynamic threat of individual platforms or systems. Richardson et al. [26] conduct a survey indicating that 62.3% respondents apply patch after a security incident happens. A report by Mellberg [18] indicates that only 59% of small companies ($\leq$ \$50M revenue) have patch management which conforms to the number investigated by Richardson et al. [26]. This motivates our modeling study in this paper since a large number of IaaS users are individuals and small companies [19].

8. CONCLUSION

We identify the threat of exploiting prevalent vulnerabilities in IaaS cloud with an empirical study and real penetration test in Amazon EC2. We pinpoint that such threat exponentially increases the risk level of cloud due to two factors: the prevalent vulnerabilities can spread quickly on public cloud as one image could potentially be instantiated by a large number of users, and the nature of the cloud enables more cost-effective attacks than traditional in-house computing environment. We analyze the cost and effectiveness of exploiting and defending prevalent vulnerabilities under traditional and cloud environments. Our results indicate that both cloud attackers and defenders have lower cost-effectiveness ratio, which enables a game-like tactical scenario between them. To further illustrate the influence of dynamic cost-effectiveness nature, we build a 2-player game theoretic model and a risk-gain analysis to capture the risks associated with two types (rational or unconscious) of attackers and defenders. Our result reveals that both attack and defense become less cost-effective in cloud as time goes by, which suggests the defender should be more responsive and proactive under cloud environment. We stir up them with a number of possible countermeasures against such threat.

9. ACKNOWLEDGEMENTS

Su Zhang and Xinming Ou were partially supported by the Air Force Office of Scientific Research award FA9550-12-1-0106 and U.S. National Science Foundation awards 0954138 and 1018703. Any opinions, findings and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the above agencies.

10. REFERENCES

[1] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro, "A security analysis of amazon's elastic compute cloud service," in *Proc of the 27th Annual ACM SAC*, 2012.

[2] *AWS Customer Agreement.* http://aws.amazon.com/agreement/: Amazon Web Services LLC., 2012.

[3] S. Bugiel, S. Nürnberger, T. Pöppelmann, A. Sadeghi, and T. Schneider, "Amazonia: when elasticity snaps back," in *Proc of the 18th ACM CCS*, pp. 389–400, 2011.

[4] S. Zhang, D. Caragea, and X. Ou, "An empirical study on using the national vulnerability database to predict software vulnerabilities," in *Database and Expert Systems Applications*, pp. 217–231, Springer, 2011.

[5] H. Huang, S. Zhang, X. Ou, A. Prakash, and K. Sakallah, "Distilling critical attack graph surface iteratively through minimum-cost sat solving," in *Proceedings of the 27th Annual Computer Security Applications Conference*, pp. 31–40, ACM, 2011.

[6] J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. R. Rajagopalan, and A. Singhal, "Aggregating vulnerability metrics in enterprise networks using attack graphs," *Journal of Computer Security*, vol. 21, no. 4, pp. 561–597, 2013.

[7] S. Zhang, X. Ou, A. Singhal, and J. Homer, "An empirical study of a vulnerability metric aggregation method," in *The 2011 International Conference on Security and Management (SAM'11), special track on Mission Assurance and Critical Infrastructure Protection (STMACIP'11)*, 2011.

[8] S. Zhang, X. Ou, and J. Homer, "Effective network vulnerability assessment through model abstraction," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 17–34, Springer, 2011.

[9] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of system-level vulnerability metrics through actual attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 825–837, 2012.

[10] W. P. WD, F.-X. A. THA, K. M. THA, and D. Date, "Posecco-prototype for vulnerability autonomous assessment and remediation," 2013.

[11] H. Holm, *A Framework and Calculation Engine for Modeling and Predicting the Cyber Security of Enterprise Architectures*. Kth royal institute of technology, dissertation trita-ee, issn 1653-5146; 2014:001, KTH Royal Institute of Technology, 2014. ISBN 978-91-7595-005-1.

[12] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *Security & Privacy, IEEE*, vol. 9, no. 2, pp. 50–57, 2011.

[13] H. Shrobe, "What if we got a do-over?," in *Proc of the 2012 ACM Workshop on CCSW*, ACM.

[14] L. Bilge and T. Dumitras, "Before we knew it: an empirical study of zero-day attacks in the real world," in *ACM CCS*, 2012.

[15] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0," in *Published by FIRST-Forum of Incident Response and Security Teams*, pp. 1–23, 2007.

[16] W. Zhou, P. Ning, X. Zhang, G. Ammons, R. Wang, and V. Bala, "Always up-to-date: scalable offline patching of vm images in a compute cloud," in *Proc of the 26th ACSAC*, 2010.

[17] L. Litty and D. Lie, "Patch auditing in infrastructure as a service clouds," in *ACM SIGPLAN Notices*, vol. 46, pp. 145–156, ACM, 2011.

[18] J. Mellberg, *Take patch management to the next level*. https://www.brighttalk.com/webcast/8113/54861: Secunia, 2012.

[19] *Customer Success. Powered by the AWS Cloud.* http://aws.amazon.com/solutions/case-studies/: Amazon Web Services LLC., 2008.

[20] D. Talbot, "Vulnerability seen in amazon's cloud-computing," tech. rep., MIT Tech Review, 2009.

[21] D. Maynor, *Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research*. Syngress, 2007.

[22] A. Ghosh, S. Noel, and S. Jajodia, "Mapping attack paths in black-box networks through passive vulnerability inference," tech. rep., DTIC Document, 2011.

[23] S. Shin and G. Gu, "Conficker and beyond: a large-scale empirical study," in *Proceedings of the 26th ACSAC*, 2010.

[24] *Amazon Web Services Discussion Forums.* https://forums.aws.amazon.com/forum.jspa?forumID=30: Amazon Web Services LLC.

[25] R. Lippmann, D. Fried, K. Piwowarski, and W. Streilein, "Passive operating system identification from tcp/ip packet headers," in *Workshop on Data Mining for Computer Security*, p. 40, Citeseer, 2003.

[26] R. Richardson, "15th annual 2010/2011 computer crime and security survey," tech. rep., Computer Security Institute, 2011.

[27] P. Koehler, A. Anandasivam, M. Dan, and C. Weinhardt, "Cloud services from a consumer perspective," *AMCIS 2010 Proc*.

[28] F. Consulting, *DDoS: A Threat You Can't Afford To Ignore*. Jan. 2009.

[29] R. Zhuang, S. Zhang, S. A. DeLoach, X. Ou, and A. Singhal, "Simulation-based approaches to studying effectiveness of moving-target network defense," in *National Symposium on Moving Target Research*, 2012.

[30] R. Zhuang, S. Zhang, A. Bardas, S. A. DeLoach, X. Ou, and A. Singhal, "Investigating the application of moving target defenses to network security," in *Resilient Control Systems (ISRCS), 2013 6th International Symposium on*, pp. 162–169, IEEE, 2013.

[31] I. Unruh, A. G. Bardas, R. Zhuang, X. Ou, and S. A. DeLoach, "Compiling abstract specifications into concrete systems–bringing order to the cloud," tech. rep., Kansas State University, 2013.

[32] G. Yan, R. Lee, A. Kent, and D. H. Wolpert, "Towards a bayesian network game framework for evaluating ddos attacks and defense," in *ACM CCS*, 2012.

[33] T. Khirwadkar, *Defense against network attacks using game theory*. PhD thesis, University of Illinois, 2011.

[34] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, IEEE.

[35] R. Studer, "Economic and technical analysis of botnets and denial-of-service attacks," *Communication systems IV*, p. 19, 2011.

[36] T. Forbath, P. Kalaher, and T. O'Grady, "The total cost of security patch management," tech. rep., Wipro Product Strategy & Architecture, 2005.

[37] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale vulnerability analysis," in *Proc of the 2006 SIGCOMM workshop on Large-scale attack defense*, ACM.