Federated Cloud Security Architecture for Secure and Agile Clouds

Weiliang Luo, Li Xu, Zhenxin Zhan, Qingji Zheng, and Shouhuai Xu

Abstract Cyber threats against clouds have evolved rapidly. Traditional reactive cyber defense technologies are not effective and sufficient to protect federated clouds. This chapter introduces the novel federated cloud security architecture that includes proactive cloud defense technologies for secure and agile cloud development. The federated security architecture consists of a set of seamlessly integrated systematic security mechanisms at the application layer, the network layer and the system layer in federated cloud computing environments. Features of the architecture include: (1) it is centered on proactive cyber defense; (2) it facilitates to detect early warning cyber attacks against at one layer and deploy early warning signs of attacks to other layers for countermeasures; (3) it uses command and control (C2) to coordinate both in-cloud and cross-cloud defense activities via federated cloud security centers.

1 Introduction

When cloud consumers decide whether or not to adopt cloud computing, cloud security is the most important factor [5, 26, 39, 40]. This is reasonable because the problem of securing cloud systems is much more challenging than the problem of securing an individual enterprise Information Technology (IT) system. First, a cloud system can be seen as the "merger" of many enterprise IT systems that share the same hardware and possibly software infrastructure. As a consequence, traditional perimeter isolation between the physical enterprise IT systems may not be competent for isolating the cloud-hosted virtual enterprise IT systems. Second, a cloud system is harder to defend than a physical enterprise IT system because it

W. Luo (✉) • L. Xu • Z. Zhan • Q. Zheng • S. Xu
University of Texas at San Antonio, San Antonio, TX, USA
e-mail: wluo@cs.utsa.edu; lxu@cs.utsa.edu; zzhan@cs.utsa.edu; qzheng@cs.utsa.edu; shxu@cs.utsa.edu

K.J. Han et al. (eds.), *High Performance Cloud Auditing and Applications*,
DOI 10.1007/978-1-4614-3296-8_7, © Springer Science+Business Media New York 2014

must be open to all users of the cloud-hosted enterprise IT systems. In other words, a cloud system has a much larger attack surface, including a larger population of authorized users. As a consequence, a successful attack launched by an authorized user against the cloud can cause damage to all users of the cloud.

Given the scale and complexity of cloud systems, purely reactive cyber defense is far from sufficient. Moreover, threats against clouds will evolve rapidly. In order to adequately defend cloud systems against dynamic cyber threats, proactive defense should be widely deployed because it can offer defenders some inherent situational awareness and early warning capabilities. Toward this goal, we propose a cloud security architecture that can seamlessly integrate together various proactive defense mechanisms. To our knowledge, this challenging and important problem has not been systematically explored.

In this chapter, we propose a novel federated cloud security architecture to detect dynamic cyber threats (i.e., new and possibly zero-day attacks) against clouds *as early as possible* so that the defender can have enough time to deploy countermeasures. The architecture is centered on proactive defense, and consists of a set of seamlessly integrated systematic security mechanisms at the application layer, the network layer, and the system layer. More specifically, the architecture can be characterized as follows:

- *Application-Layer Situation Awareness*: We describe how application-layer proactive cyber defense mechanisms can be deployed to defend the dynamic abuse of clouds. Although some of the integrated mechanisms were known, the others are proposed here for the first time.
- *Network-Layer Situation Awareness*: We describe how network-layer proactive defense mechanisms can be deployed to detect new cyber attacks against clouds. In particular, we illustrate how to blend honeypots into the IP address space of production cloud systems, which will make it harder for the attackers to identify and bypass the honeypots.
- *System-Layer Situation Awareness*: We describe how proactive defense mechanisms can be deployed deep in the software stack to defend stealthy attacks against clouds. This includes the use of honeytokens to detect stealthy attacks.
- *Cloud C2 Security Center Situation Awareness*: The architecture uses a cloud C2 security center to coordinate the in-cloud and cross-cloud defense activities. As a result, new attacks detected at one layer may lead to countermeasures that can be deployed at the other layers, and clouds can rapidly share information about newly detected attacks as well as the countermeasures.

To our knowledge, this is the first work that systematically explores cloud security architectures, which will play a crucial role in the cloud computing era.

The rest of the chapter is organized as follows. Section 2 presents an overview of cloud security. Section 3 describes cloud system and cloud security threats. Section 4 introduces the novel federated security architecture that consists of cross-layer and cross-cloud security components. Section 5 describes the federated cloud C2 security modules. Section 6 discusses the usefulness of the federated cloud C2 centers. Finally, Section 7 presents our conclusions.

2 Cloud Security

We briefly review cloud security [40] and related prior work based on layers at which the defense mechanisms are deployed. For detecting malicious websites as early as possible, there have been studies on combating the abuse of cloud resources to host botnet command and control (C&C) severs [22] and to crack passwords [37]. The C&C is used for the botnet operation in this chapter. There have been studies on combating spam campaigns in cloud-like settings [10, 13, 20, 32, 43, 49, 52]. The problem of detecting malicious websites is relatively new and has not been thoroughly investigated, despite some interesting initial studies [12, 14]. A competent cloud security architecture should adequately address the problem of malicious websites.

The network-layer proactive defense presented in this chapter advocates the use of honeypots in some novel ways. Previous studies related to honeypots mainly focused on analyzing honeypot-observed probe activities [25] and characterizing honeypot-observed attacks [1–3, 15, 16, 28, 35, 36, 47].

For dealing with APT-like sophisticated and stealthy attacks, there have been studies such as how to cope with the exploitation of VM image management to distribute malicious VM images [30]. More advanced studies have focused on enhancing VMM security [8, 42, 44, 50]. Overall, proactive defense at the system layer is an important problem that is not yet well understood.

Despite the aforementioned studies on various aspects of cloud security, there is no prior work that systematically explores comprehensive security architectures for securing clouds against dynamic cyber attacks. The proposed security architecture fills the void.

3 Cloud System and Cloud Security Threats

In this section we discuss the cloud system model and the associated threat model. The cloud system model is general enough to accommodate the so-called public clouds, private clouds, community clouds, and social clouds [6, 54]. This means that the threat model and the security architecture that we will present later are equally applicable to these kinds of clouds.

3.1 Cloud System Model

As shown in Fig. 1, a cloud provides both computational (including hardware and possibly software) infrastructure and storage services to its users. Cloud users may include enterprise users, home users, and mobile users. Enterprise users may outsource their data and computing tasks to the cloud, while allowing their

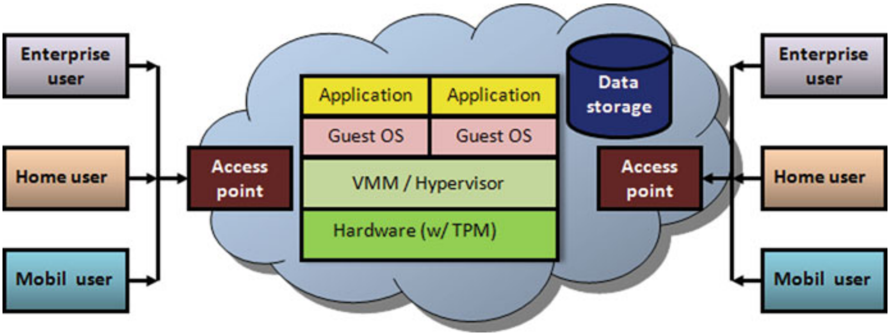

Fig. 1 Cloud system model

employees and business partners to access their data. Home users may outsource their data (e.g., emails, files and documents) to the cloud. Mobile users may access the cloud-hosted data and exploit the cloud to accomplish some compute-intensive and data-intensive tasks. Cloud users utilize cloud resources and services through some *access points*, which are managed and maintained by the cloud in question.

Cloud resource management [4] is centered on the concept of virtual machine (VM), which is an abstraction that eases the allocation of physical computing resources to cloud users. Each VM runs its own guest Operating System (OS), and may run one or more software applications of a cloud user. Multiple VMs, which may be allocated (or rented) to one or more cloud users, run on top of a Virtual Machine Monitor (VMM) or Hypervisor that manages resources of a physical computer. It is also possible that some physical computers may not need to run VMM at all because the entire physical computer is rented to a single cloud user. Each physical computer may have an associated Trusted Platform Module (TPM) [48], which may or may not be further virtualized so that the virtual machines can use the virtualized TPMs [11] (one virtual TPM per VM). A cloud can offer data storage services to its users. The outsourced data can be unstructured, semi-structured or structured. Unstructured data can be simply bitstrings or files. Semi-structured data can be emails. Structured data can be a relational database.

3.2 Cloud Security Threats

Threats against clouds may be launched by anyone, who may or may not be a legitimate participant in the above cloud system model. We assume that the cloud vendors (including their employees) are trusted not to attack the cloud users. While orthogonal to the focus of the present chapter, we mention that there have been studies on assuring cloud data storage integrity [7, 21, 55] and cloud data query integrity [24, 33, 56]. These mechanisms can be incorporated into our security architecture, in a plug-and-play fashion, to deal with potentially malicious cloud vendors. We assume that the cloud users may be compromised or malicious. For example, a cloud user may launch a malicious VM instance to attack the VMM or other VMs running on top of the same VMM, or may abuse a cloud VM to conduct malicious activities such as hosting malicious websites. More specifically, we focus on the following classes of powerful threats against clouds.

- *Abusing clouds to host malicious websites to launch drive-by download attacks*: Cloud resources have been, and will continue to be, abused to host malicious websites, which can launch drive-by download attacks against the vulnerable browsers and therefore the user-end computers. Cloud resources have been, and will continue to be, abused to facilitate botnet C&C operations.
- *Spreading new malwares*: These attacks often exploit software vulnerabilities that may not have been publicly available, and are powerful because it can be very difficult to detect these attacks.
- *Stealthy Advanced Persistent Threats (APT)*: These attacks do not attempt to infect as many computers as possible, but rather attempt to compromise targeted computers through vulnerabilities deep in the software stacks. These attacks are powerful because they often can bypass application-layer and network-layer defense, and may only be dealt with at the lower system layer.

Our primary goal is to detect new and possibly zero-day attacks belonging to the above classes as early as possible so that the defenders have enough time to deploy countermeasures.

4 Federated Agile Cloud Security Architecture

Now we present the cloud security architecture, which can offer the defender inherent capabilities of situational awareness and early warning against dynamic cyber threats in federated cloud computing environments. The architecture is centered on proactive defense.

4.1 Architecture Description

Figure 2 highlights the security architecture, which consists of (1) defense components at the application layer, (2) defense components at the network layer, (3) defense components at the system layer, and (4) the C2 center that coordinates both the defense within a single cloud and the defense crossing multiple clouds. In what follows we briefly discuss the functions of these defense components, which are then elaborated in the following subsections.

Application-layer defense can naturally deploy traditional reactive defense mechanisms. For example, there have been many proposals for enhancing security of execution environments [17, 19, 46], which can be deployed at the cloud-

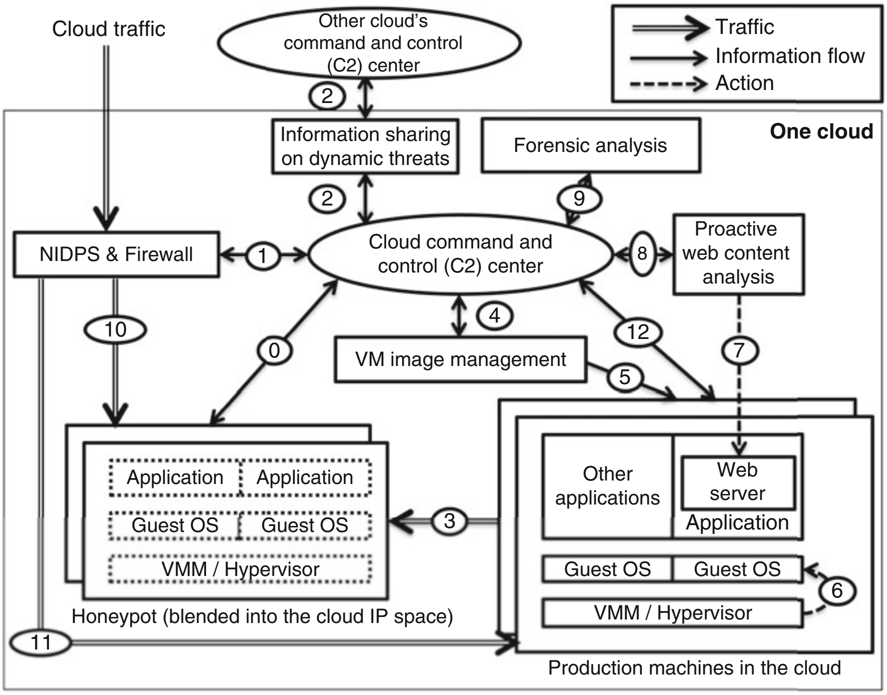

Fig. 2 Cloud security architecture with C2 security centers for coordinating in-cloud and cross-cloud defense activities in federated cloud computing environments

end to enhance the isolation between the applications. Nevertheless, we advocate application-layer proactive defense that can significantly benefit from the rich semantics of application-layer contents. Since websites have become a popular attack channel, we will focus on mitigating the abuses of websites to launch attacks. For this purpose, we propose using a *proactive web content analysis* component to identify malicious/compromised websites (Arrow 7). We elaborate application-layer proactive defense in Section 4.2.

It is well known that Network Intrusion Detection/Prevention System (NIDPS) & Firewall can possibly filter malicious traffic before it reaches the production machines (Arrow 11). However, effectiveness of NIDPS & Firewall depends on the accuracy of the detection and filtering algorithms and on the length of early warning time. We advocate fulfilling network-layer proactive defense by exploiting honeypots (http://www.honeynet.org/). Honeypots are vulnerable computers that can be used to monitor attacks [41] and collect malicious attack traffic and learn the evolution of cyber attacks (Arrow 10). The newly identified and learned attack vectors can be disseminated to NIDPS & Firewall (Arrow 0) for filtering attacks. Honeypots can also detect attack traffic originating from the cloud production machines (Arrow 3). We elaborate network-layer proactive defense in Section 4.3.

System-layer defense aims at detecting APT-like attacks against guest OS or even VMM as early as possible. System-layer defense is crucial because even network- and application-layer defenses together are not sufficient to defeat APT-like sophisticated attacks. We propose using enhanced *VM image management* component (Arrow 5) to enhance VM image security [51]. We also advocate taking advantage of VMMs to inspect guest OS kernels (Arrow 6), and taking advantage of recent progress in exploiting architectural features to enhance security of VMMs. We elaborate system-layer proactive defense in Section 4.4.

We propose letting each cloud run a C2 center. The C2 center shares updated threat information via the channel called *information sharing on dynamic threats* (Arrow 2), receives updated information from the defense components in the cloud (Arrows 0, 1, 4, 8, 9, 12), and coordinates activities of the components within the cloud (Arrows 0, 1, 4, 8, 9, 12). For example, the C2 center learns new attacks from the *proactive web content analysis* component (Arrow 8), and instructs the *NIDPS & Firewall* component (Arrow 1) and the *VM image management* component (Arrow 4) to proactively adjust their defense. We elaborate the federated cloud C2 security modules in Section 5.

4.2 Application-Layer

Web servers have been abused to attack users' computers. Since hosting malicious websites will hurt the reputation of cloud vendors, the vendors have the incentive to rapidly identify malicious websites they host. A cloud vendor may choose to attain situational awareness and early warning by a third-party or by itself.

4.2.1 Third-Party Monitoring Cloud-Hosted Websites

A cloud vendor can outsource the monitoring of the websites it hosts to a third party. There are three kinds of proactive mechanisms for this purpose, each of which has its own strengths and weaknesses.

- *Static approach*: Detect malicious websites via static analysis of website contents or even URLs themselves [18, 27]. This approach is very efficient, but often has limited success against sophisticated attacks such as obfuscation.
- *Dynamic approach*: Detect malicious websites by analyzing their runtime behaviors. This can be done by using various client honeypots, such as Capture-HPC [38] and PhoneyC [31]. This approach can cope with obfuscation but also resource consuming.
- *Hybrid approach*: Achieve the best of the above two approaches [12]. It uses a front-end static analysis tool to identify suspicious websites and a back-end dynamic analysis tool to further examine the websites.

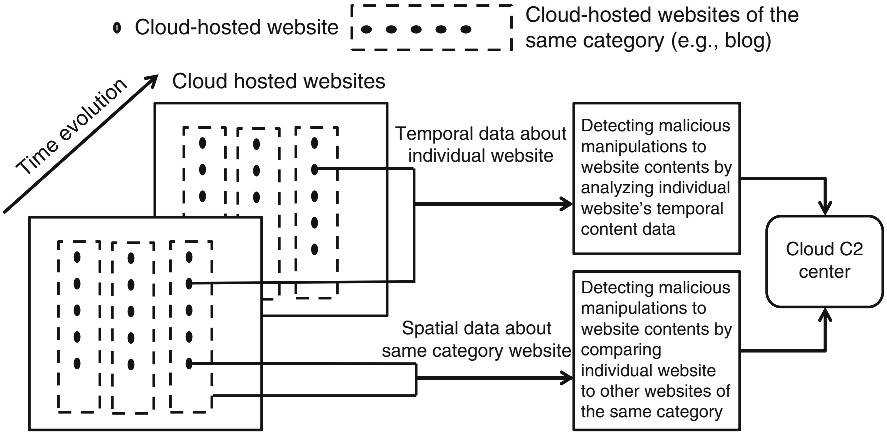
Cloud-hosted website
Cloud-hosted websites of the same category (e.g., blog)
Time evolution
Cloud hosted websites
Temporal data about individual website
Spatial data about same category website
Detecting malicious manipulations to website contents by analyzing individual website's temporal content data
Detecting malicious manipulations to website contents by comparing individual website to other websites of the same category
Cloud C2 center

Fig. 3 Cloud vendor proactively monitors malicious websites it hosts

4.2.2 Cloud Vendor-Side Websites Monitoring

A cloud vendor can analyze the website contents by using the aforementioned static, dynamic, and hybrid methods. This approach has the drawback that the analysis must be conducted as frequently as possible. Unfortunately, it is not practical to analyze each website every minute or even every few seconds for near real-time auditing that is necessary in order to detect attacks as early as possible. We propose the following new method by which the cloud vendor (more precisely, the C2 center) can be triggered to analyze the website contents (i.e., analysis on demand rather than periodically analyzing the website contents).

We consider two cases that depend on how websites hosted in clouds become malicious. In the first case, malicious websites are set up by malicious cloud users, which are accommodated in our threat model. In the second case, websites are set up by honest cloud users but later compromised by attackers for malicious purposes. To detect malicious websites in the first case, the vendor can, as illustrated in Fig. 3, classify websites into categories such as news, blog, searching, shopping etc. The websites belonging to an individual category would exhibit similar characteristics; whereas different categories of websites would exhibit different characteristics. This allows the cloud vendor to build patterns for each category of websites. The vendor could use the pattern of a specific category to detect an anomaly of a specific website belonging to the category. This would allow the vendor to detect malicious websites that were set up by honest cloud users but compromised by the attacker to launch further attacks. To detect malicious websites in the second case, the vendor can, as illustrated in Fig. 3, analyze the temporal data associated with individual website, possibly including the IP addresses from which accesses are initiated and the IP addresses the website redirects access to.

Compared with the aforementioned proactive defense based on third parties, the above proactive defense by the cloud vendor has the following advantages. First, attacks could be detected significantly earlier because deeper analysis can be automatically triggered when certain system calls are invoked to inject malicious code into the website content. This can be done more efficiently by the third-party approach that blindly analyzes all websites in a "brute-force" fashion. To be specific, let us look at the following concrete example. Suppose an attacker attempts to embed <iframe src="www.malicioussite.com"> into an innocent website www.innocentsite.com, which will cause the browsers pointing to www.innocentsite.com to automatically access www.malicioussite.com and therefore get compromised. Since the code injection will invoke some file I/O system calls, these invocation operations can automatically alert the cloud vendor with the potential attack. Similarly, a malicious website set up by a legitimate but malicious user may be detected at the setup time because it exhibits different characteristics from other non-malicious websites belonging to the same category. Therefore, cloud vendors can enforce defense that is impossible for third parties to implement because they do not have access to the data. Second, newly detected attacks can be more easily communicated to the cloud C2 center, which can take appropriate actions upon receiving reports on malicious websites.

4.2.3 Effectiveness of Application-Layer Proactive Defense

Effectiveness of application-layer proactive defense depends on several factors. First, proactive website content analysis should scale up to the large number of websites. This is why we advocate that each cloud monitors its own websites. Second, proactive website content analysis should be able to trace redirections automatically because attackers have been abusing redirections to hide the locations of the actual web servers that host malicious malwares. This problem is not trivial because redirections can be made sophisticated enough to defeat most static tracing methods.

4.3 Network-Layer

Compromised computers/VMs can attack the vulnerable computers/VMs in the same or different cloud. Network-layer proactive defense can offer situational awareness and early warning against such dynamic threats. Honeypots are vulnerable computers that can be used to monitor attacks [41]. Although the idea of using honeypots for detecting new attacks is not new, we propose enhancing this proactive defense by blending the honeypot IP addresses into the production network IP address space, rather than running honeypots in a separate unused address space, to make it more difficult for the attackers to figure out IP addresses of the honeypot. We also propose enhancing the effectiveness of this proactive defense with more personalized or diversified honeypots.

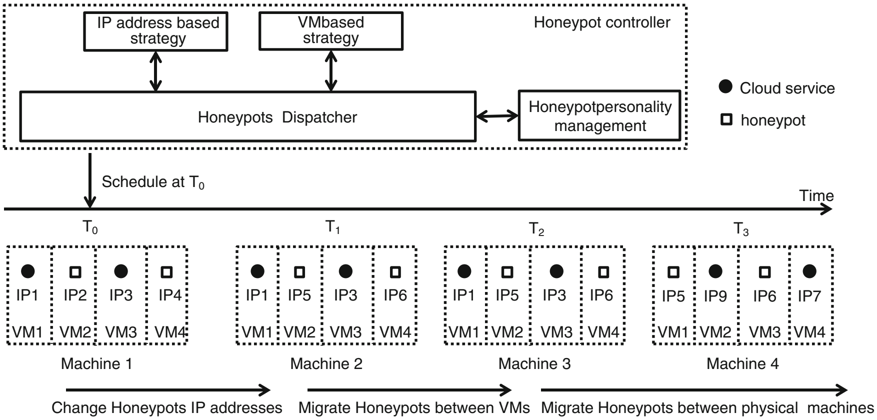

Fig. 4 Dynamic honeypots blended into the IP address space of production cloud

4.3.1 Exploiting Honeypots to Rapidly Learn New Attacks

Low-interaction honeypots can provide limited interactions with attackers, and may be easily recognized and bypassed by sophisticated attackers. In order to detect new and possibly zero-day attacks as early as possible, we advocate the use of high-interaction honeypots that use real but vulnerable operating systems to trap cyber attacks. High-interaction honeypots can collect sophisticated malicious attack traffic that would allow the C2 center to generate countermeasures for other components. For example, patterns of the observed malicious code can be extracted as NIDPS signatures. Since high-interaction honeypots often consume much more resources than low-interaction honeypots, we advocate the hybrid deployment of low-interaction and high-interaction honeypots so as to achieve better trade-off between effectiveness and efficiency. Basically, low-interaction honeypots can serve as a front-end tool for rapidly filtering known attacks, while high-interaction honeypots can serve as a back-end tool for interacting with unknown attacks that are not recognized by low-interaction honeypots.

As mentioned above and highlighted in Fig. 4, we propose the following enhancements to honeypot-based detection of new attacks. “IPx” stands for IP address x, “VMy Machine z” stands for the y-th VM on the z-th physical computer. First, we propose blending honeypot IP addresses into the production cloud network IP address. Traditionally, honeypots are set up in unused IP address space, which however can be easily recognized and bypassed by sophisticated attackers. We propose using a honeypot dispatcher to manage the dynamic IP addresses and VMs that are used for honeypot purposes. For example, at time T_0, VM2 of IP address 2 corresponding to physical computer 1 is used for honeypot purpose, which is migrated at time T_1 to VM2 of IP address 5 corresponding to physical computer 2, migrated at time T_2 to VM4 of IP address 6 corresponding to physical computer 3, and migrated at time T_3 to VM1 of IP address 7 corresponding to physical computer 4. Second, we propose to diversify the "personalities" of honeypots to achieve high fidelity in detecting new attacks. This is important because without diversifying honeypot personalities, a sophisticated attacker can recognize honeypots by analyzing the characteristics of the honeypots. This capability is independent of, and therefore can be incorporated with, the capability of recognizing honeypots via their IP addresses. Without the above enhancements, honeypots can be relatively easily evaded and cannot effectively detect new attacks.

4.3.2 Effectiveness of Network-Layer Proactive Defense

Effectiveness of network-layer proactive defense depends on the following factors. First, how well are the honeypots blended into the IP address space of the production cloud system? Traditional honeypots are often deployed at consecutive, unused IP spaces, which are relatively easy to be identified and bypassed by sophisticated attackers. Second, how often are the honeypots re-allocated or re-deployed? Even if high-interaction honeypots' IP addresses are well blended into the cloud system, attackers may still identify them because, after all, even high-interaction honeypots may not run semantically meaningful applications. This highlights the importance of re-deploying high-interaction honeypots periodically. The re-deployment operation should be conducted before attackers figure out the dynamic honeypot IP address spaces.

4.4 System-Layer

APT-like attacks might be able to bypass the application-layer and network-layer proactive defense. In order to counter such sophisticated attacks, we need system-layer proactive defense that is deployed deep into the software stack, namely at the levels of guest OS or even VMM. For this purpose, we not only need to enhance guest OS and VMM security, but also need to use new proactive approaches to detect such powerful attacks as early as possible.

4.4.1 Enhancing Guest OS and VMM Security Proactively

In order to accelerate the creation of software environment in clouds, guest OS is launched from VM image. This unfortunately also expands the attack surface of guest OS because malicious VM image directly leads to malicious guest OS. For example, a malicious VM image intentionally crafted by an attacker can cause a guest OS to run with rootkits, backdoor or manipulated kernel [30]. Therefore, cloud vendors should implement a VM image management system that enforces VM image access control, image provenance information tracking, and security patch [51]. Provenance data can be collected during the lifetime of VM images to establish a chain of trust starting at a root (e.g., Amazon or IBM). If the VMM is trusted, secure loading of guest OS is straightforward because the VMM can measure the code executed in the bootstrapping process of guest OS.

Once successfully launched, it is important to ensure runtime security of guest OS. This remains an open problem [34, 53]. There have been two promising approaches to assure runtime security of guest OS: *in-guest* protection and *VMM-supported* protection. The in-guest protection approach is similar to traditional OS protection, and depends on some defense mechanisms (e.g., anti-malware software) that run inside the guest OS itself. As long as the defense mechanisms can deal with dynamic attacks, this approach is still useful. This is possible because, for example, the anti-malware software can be behavior-based rather than signature-based. The VMM-supported protection approach is more powerful because it exploits VMM to inspect the states of the VMs. Although complete inspection and verification of system state is infeasible, it is possible to verify specific properties such as control-flow integrity in the guest OS kernel [34].

In the above, we assumed that VMMs are trusted. Security of VMMs is somewhat debatable and it is ideal not to trust VMMs. Nevertheless, there have been studies how to weaken (if not eliminate) the trust in VMMs, for example, by running VMs or applications on (almost) naked hardware [9,44] and exploiting hardware to protect VMs from malicious VMMs [45]. There also have been studies on enhancing VMM security [8,23,29,42,50]. Future progress in these aspects can be incorporated into the cloud security architecture in a plug-and-play fashion.

4.4.2 Using Honeytokens to Detect APT-Attacks Against Guest OS

APT-like attacks may still be able to penetrate the application-layer and network-layer proactive defense and the enhanced guest OS defense. We therefore propose using the following methods to detect such attacks as early as possible. For example, attackers may exploit zero-day vulnerabilities in the guest OS to launch stealthy attacks, which can be very difficult to observe from inside of the guest OS or even from inside of the VMM. This suggests that we need to detect such powerful attacks from outside of the guest OS or even from outside of the VMM. One approach is to run a guest OS on top of a VMM, where both guest OS and VMM do not have any known vulnerabilities (from the perspective of the cloud vendor or defender). The guest OS can set up a set of user accounts, and the VMM can incorporate the inspection mechanisms mentioned above. Each user account has access to a set of honeytokens such as: (1) an email address (e.g., nobody@nowhere.com) that is never published or supposed to receive any emails; (2) a username/password for another account at possibly another computer that is never supposed to be accessed; (3) fake data containing specific keywords that may be of interest to APT-like attacks.

Even if the attacks bypassed all of the defense mechanisms mentioned above, they could still be detected by this system-layer proactive defense as follows. When the email account nobody@nowhere.com receives any email, possibly spam or email with a malicious attachment, we can conclude that the guest OS has been compromised. Similarly, when the specific fake account is accessed, we can conclude that the guest OS has been compromised.

Finally, if an APT-like attack does not encrypt the data it steals from the compromised guest OS, the VMM could detect the compromise by inspecting whether the special keywords appear in the outgoing packets; if an APT-like attack does encrypt the data it steals from the compromised guest OS, this fact of encrypted outgoing packets could still trigger the VMM's inspection mechanism for further investigation. The latter is possible because the vendor can make the outgoing traffic encrypted using some cryptographic keys that are known to the defender. When the VMM inspection mechanism cannot decrypt the outgoing packets that were encrypted by the attacker using its own cryptographic key, the VMM can conclude that the guest OS has been compromised.

4.4.3 Effectiveness of System-Layer Proactive Defense

System-layer proactive defense has a great potential in detecting APT-like stealthy attacks. The effectiveness of system-layer proactive defense depends on the following factors. First, how effective can the runtime attack-detection mechanisms be? This is because modern guest Oss are often complex and it is challenging to detect runtime security violations. Second, how effective can VMM-based inspection of guest OS state be? This largely depends on the degree at which the semantic gap (i.e., VMM may not understand the semantics of guest OS objects and operations) can be bridged. Third, what is the optimal trade-off between the power, security, and performance of VMM? This will affect the services VMMs can provide to support guest Oss and applications, and the trustworthiness of VMMs in terms of their capability in detecting sophisticated attacks.

5 Federated Cloud C2 Security Modules

The security architecture uses cloud C2 security centers to coordinate in-cloud and cross-cloud defense activities in federated cloud computing environments. In particular, it facilitates rapid deployment of countermeasures against newly detected attacks and rapid sharing of such information between clouds. In this section we elaborate the C2 center component and discuss its usefulness in fulfilling situational awareness and early warning.

The cloud C2 security center has the following four modules: information aggregation, information fusion and analysis, information feedback, and information sharing. These modules are pictorially illustrated in Fig. 5, which also highlights the usefulness of federated cloud C2 security centers.

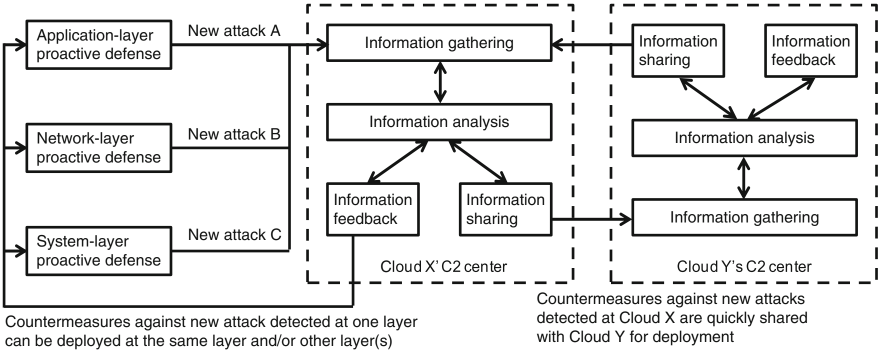
Application-layer proactive defense
Network-layer proactive defense
System-layer proactive defense
New attack A
New attack B
New attack C
Information gathering
Information analysis
Information feedback
Information sharing
Cloud X' C2 center
Information sharing
Information feedback
Information analysis
Information gathering
Cloud Y's C2 center
Countermeasures against new attack detected at one layer can be deployed at the same layer and/or other layer(s)
Countermeasures against new attacks detected at Cloud X are quickly shared with Cloud Y for deployment

Fig. 5 One scenario of cloud C2 security centers coordinating in-cloud and cross-cloud defense activities

5.1 Information Aggregation Module

This module collects dynamic attack information from the NIDPS & Firewall, the honeypot, the VM image management component, the system-layer (guest OS and VMM) security monitoring mechanisms, the proactive website analysis component, and the forensic analysis component. The gathered information may have different structures and semantics. For example, application-layer information may be specific to guest OS processes (e.g. image path, creator, I/O operations, network operations); network-layer information may identify the communication peers (e.g. IP addresses, port number, protocol, application payloads); system-layer information may be specific to guest OS and VMM operations. There might be some semantic gaps between the information gathered at different layers. This highlights the importance of identifying effective cross-layer methods to bridge the potential semantic gaps so as to maximize the utility of the gathered information.

5.2 Information Fusion and Analysis Module

This module learns and characterizes dynamic cyber threats. This module is the core of C2 centers, and interacts with the other modules by possibly instructing them to adapt their functions. This module can instruct the information aggregation module to collect more information about some specific events. Also, this module can adopt countermeasures received from other cloud C2 centers, and can adapt them before actually instructing the information feedback module to deploy them. Various (adversarial) machine learning methods can be applied. Ideally, the information analysis engine can not only extract effective detection methods against the newly identified attacks, but also can help pin down the root cause of the newly identified attacks. For example, application-layer analysis may lead to effective methods for detecting malicious websites despite the fact that they may evolve rapidly (e.g., a website may be malicious only for the period of minutes); network-layer analysis may lead to the identification of dynamic malicious IP addresses or attack signatures that can be deployed at the NIDPS & Firewall.

5.3 Information Feedback Module

This module instructs the defense mechanisms at appropriate defense points within the cloud to deploy the newly learned countermeasures. The countermeasures may be deployed at the NIDPS & Firewall to filter malicious attack traffic, may be accommodated to adjust the configuration of honeypots, and may be incorporated to enhance the security of VM images. The countermeasures can instruct the system-layer security monitor mechanisms to pay special attention to some activities, can order the proactive web content analysis component to enhance its analysis methods, and may ask the forensic analysis engine to provide more information about certain attacks. Note that the above information gathering, information analysis, and information feedback formulate a proactive defense control loop within a single cloud.

5.4 Information Sharing Module

This module shares newly learned countermeasures with other cloud C2 centers. This can be fulfilled through the component called *information sharing on dynamic threats* that is responsible for sharing information between the C2 centers of different clouds. The rapid sharing of newly detected attacks can substantially mitigate the damage because, for example, a zero-day attack can be detected after it compromises VMs in one cloud, rather than after it compromises VMs in many clouds. The C2 centers effectively formulate a higher-layer of defense network, which is necessary for defending dynamic cyber threats.

6 Usefulness of Federated Cloud C2 Security Centers

Figure 5 elaborates the usefulness of federated cloud C2 security centers. The usefulness can be understood from the following perspectives. First, as illustrated in Fig. 5, countermeasures newly developed at Cloud X can be rapidly disseminated to Cloud Y through their respective cloud C2 security centers via cross-cloud information sharing. The C2 centers that receive the new countermeasures can coordinate their deployment within their own clouds through the information feedback module, which receives instructions from the information analysis module.

Second, as illustrated in Fig. 5, a new attack detected at a particular layer can lead to countermeasures that not only can be deployed at the application layer, but also can be deployed at the network layer and/or the system layer. For example, proactive website content analysis can trace to (perhaps after redirection hops) some malware that exploits a zero-day vulnerability, which can be quickly analyzed using an automated dynamic analysis tool. Suppose, for example, new attack *A* is detected at the application layer by proactive web content analysis. The countermeasures can be naturally deployed at the application layer to, for example, blacklist not only the websites that host the malware, but also the websites that automatically redirect to the malicious websites that host the malwares. The latter is important because even if the malicious websites that host the malwares are shut down by the defender, the attacker can quickly use the same intermediate websites to redirect to newly compromised websites for hosting malwares. Moreover, the ways attack *A* exploits the operating system vulnerabilities could lead to detection methods at the system layer to block attack *A* from success. This is possible even before the vulnerability is patched because the patch may become available only after a significant amount of time. Third, although not illustrated in Fig. 5, it is possible that C2 center's information analysis module can detect new attacks that are not detected by the defense at individual layers. For example, some new attacks may only be detected after correlating information obtained at the application layer defense mechanisms, the information obtained at the network layer defense mechanisms, and even the information obtained at the system layer defense mechanisms. Extremely sophisticated attacks may only be detected by such cross-mechanisms defense techniques.

It is important to note that effectiveness of C2 centers depends on the following factors. First, how can accurate and detailed information about attacks be gathered? This directly affects the quality of the collected raw data and ultimately bounds the usefulness of the analysis results. Second, how good are the information analysis algorithms? The better the machine learning algorithms, the more useful the extracted detection models. Third, how quick can it be to extract the patterns for detecting new and possibly zero-day attacks and how quick can it be to deploy the countermeasures? This ultimately determines the effectiveness of both in-cloud proactive defense and cross-cloud proactive defense. The above open problems should be adequately addressed in future studies.

7 Conclusion

We have developed a federated cloud security architecture and cloud C2 security modules to mitigate cloud threats against federated clouds. Cyber situational awareness and early warning feature selection functionalities are included as security components that are controlled by the cloud C2 security centers. Since cloud users access cloud-based applications through web browsers, web-security related data are collected at the application-layer, the network-layer and the system-layer.

The cloud C2 security centers aggregate cyber situational awareness data to extract early warning signs. The cloud C2 security center coordinates both in-cloud and cross-cloud defense activities. The proactive cloud security mechanisms may mitigate cloud threats and enhance cloud security. Therefore, the proposed cloud defense technologies may be appropriate for secure and agile clouds.

Acknowledgements This material is based upon work partially supported by the Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-09-01-0165 and the Air Force Research Laboratory (AFRL) Visiting Faculty Research Program (VFRP) extension grant LRIR 11RI01COR. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agency.

References

1. Almotairi, S.I., Clark, A.J., Dacier, M., Leita, C., Mohay, G.M., Pham, V.H., Thonnard, O., Zimmermann, J.: Extracting inter-arrival time based behaviour from honeypot traffic using cliques. In: Proceedings of the 5th Australian Digital Forensics Conference, Perth, pp. 79–87 (2007)
2. Almotairi, S., Clark, A., Mohay, G., Zimmermann, J.: Characterization of attackers' activities in honeypot traffic using principal component analysis. In: Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing, NPC'08, Shanghai, pp. 147–154. IEEE Computer Society, Washington, DC (2008)
3. Almotairi, S., Clark, A., Mohay, G., Zimmermann, J.: A technique for detecting new attacks in low-interaction honeypot traffic. In: Proceedings of the 4th International Conference on Internet Monitoring and Protection, ICIMP'09, Venice, pp. 7–13. IEEE Computer Society, Washington, DC (2009)
4. An, K.: Resource management and fault tolerance principles for supporting distributed real-time and embedded systems in the cloud. In: Proceedings of the 9th Middleware Doctoral Symposium of the 13th ACM/IFIP/USENIX International Middleware Conference, MIDDLEWARE'12, Montreal, pp. 4:1–4:6. ACM, New York (2012). doi:10.1145/2405688.2405692
5. Anderson, T.E.: weforum.org, Exploring the future of cloud computing: riding the next wave of technology-driven transformation. http://goo.gl/BeR45 (2010)
6. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010). doi:10.1145/1721654.1721672
7. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS'07, Alexandria, pp. 598–609. ACM, New York (2007). doi:10.1145/1315245.1315318
8. Azab, A.M., Ning, P., Wang, Z., Jiang, X., Zhang, X., Skalsky, N.C.: Hypersentry: enabling stealthy in-context measurement of hypervisor integrity. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS'10, Chicago, pp. 38–49. ACM, New York (2010). doi:10.1145/1866307.1866313
9. Azab, A.M., Ning, P., Zhang, X.: Sice: A hardware-level strongly isolated computing environment for x86 multi-core platforms. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS'11, pp. 375–388. ACM, New York (2011). doi:10.1145/2046707.2046752
10. Benczur, A.A., Csalogany, K., Sarlos, T., Uher, M., Uher, M.: Spamrank – fully automatic link spam detection. In: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, AIRWeb'05, Chiba (2005)

11. Berger, S., Cáceres, R., Goldman, K.A., Perez, R., Sailer, R., van Doorn, L.: vtpm: virtualizing the trusted platform module. In: Proceedings of the 15th Conference on USENIX Security Symposium – Volume 15, USENIX-SS'06, Vancouver. USENIX Association, Berkeley (2006)
12. Canali, D., Cova, M., Vigna, G., Kruegel, C.: Prophiler: a fast filter for the large-scale detection of malicious web pages. In: Proceedings of the 20th International Conference on World Wide Web, WWW'11, pp. 197–206. ACM, New York (2011). doi:10.1145/1963405.1963436
13. Chellapilla, K., Maykov, A.: A taxonomy of javascript redirection spam. In: Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web, AIRWeb'07, Banff, pp. 81–88. ACM, New York (2007). doi:10.1145/1244408.1244423
14. Choi, H., Zhu, B.B., Lee, H.: Detecting malicious web links and identifying their attack types. In: Proceedings of the 2nd USENIX Conference on Web Application Development, WebApps'11, Portland, pp. 121–132. USENIX Association, Berkeley (2011)
15. Clark, A., Dacier, M., Mohay, G., Pouget, F., Zimmermann, J.: Internet attack knowledge discovery via clusters and cliques of attack traces. J. Inf. Assur. Secur. **1**(1), 21–32 (2006)
16. Conti, G., Abdullah, K.: Passive visual fingerprinting of network attack tools. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC'04, Washington DC, pp. 45–54. ACM, New York (2004). doi:10.1145/1029208.1029216
17. Dai, W., Jin, H., Zou, D., Xu, S., Zheng, W., Shi, L.: Tee: A virtual drtm based execution environment for secure cloud-end computing. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS'10, Chicago, pp. 663–665. ACM, New York (2010). doi:10.1145/1866307.1866390
18. Garera, S., Provos, N., Chew, M., Rubin, A.D.: A framework for detection and measurement of phishing attacks. In: Proceedings of the 2007 ACM Workshop on Recurring Malcode, WORM'07, pp. 1–8. ACM, New York (2007). doi:10.1145/1314389.1314391
19. Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M., Boneh, D.: Terra: A virtual machine-based platform for trusted computing. ACM SIGOPS Oper. Syst. Rev. **37**(5), 193–206 (2003). doi:10.1145/1165389.945464
20. Gyongyi, Z., Garcia-Molina, H.: Web spam taxonomy. In: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, AIRWeb'05, Chiba (2005)
21. Juels, A., Kaliski, B.S., Jr.: Pors: proofs of retrievability for large files. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS'07, Alexandria, pp. 584–597. ACM, New York (2007). doi:10.1145/1315245.1315317
22. Kartaltepe, E.J., Morales, J.A., Xu, S., Sandhu, R.: Social network-based botnet command-and-control: emerging threats and countermeasures. In: Proceedings of the 8th International Conference on Applied Cryptography and Network Security, ACNS'10, Beijing, pp. 511–528. Springer, Berlin/Heidelberg (2010)
23. Klein, G., Elphinstone, K., Heiser, G., Andronick, J., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Kolanski, R., Norrish, M., Sewell, T., Tuch, H., Winwood, S.: sel4: formal verification of an OS kernel. In: Proceedings of the 2009 ACM SIGOPS 22nd Symposium on Operating Systems Principles, SOSP'09, Big Sky, pp. 207–220. ACM, New York (2009). doi:10.1145/1629575.1629596
24. Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L.: Dynamic authenticated index structures for outsourced databases. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, SIGMOD'06, Chicago, pp. 121–132. ACM, New York (2006). doi:10.1145/1142473.1142488
25. Li, Z., Goyal, A., Chen, Y., Paxson, V.: Towards situational awareness of large-scale botnet probing events. IEEE Trans. Inf. Forensics Secur. **6**(1), 175–188 (2011). doi:10.1109/TIFS.2010.2086445
26. Luna Garcia, J., Langenberg, R., Suri, N.: Benchmarking cloud security level agreements using quantitative policy trees. In: Proceedings of the 4th ACM Workshop on Cloud Computing Security Workshop, CCSW'12, Raleigh, pp. 103–112. ACM, New York (2012). doi:10.1145/2381913.2381932

27. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: learning to detect malicious web sites from suspicious urls. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'09, Paris, pp. 1245–1254. ACM, New York (2009). doi:10.1145/1557019.1557153
28. Mahoney, M.V., Chan, P.K.: Learning nonstationary models of normal network traffic for detecting novel attacks. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'02, Edmonton, pp. 376–385. ACM, New York (2002). doi:10.1145/775047.775102
29. McCune, J.M., Li, Y., Qu, N., Zhou, Z., Datta, A., Gligor, V., Perrig, A.: Trustvisor: efficient TCB reduction and attestation. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy, SP'10, Oakland, pp. 143–158. IEEE Computer Society, Washington, DC (2010). doi:10.1109/SP.2010.17
30. Meer, H., Arvanitis, N., Slaviero, M.: defcon.org, Clobbering the cloud. http://goo.gl/42hRL (2009)
31. Nazario, J.: usenix.org, PhoneyC: a virtual client Honeypot. http://goo.gl/euYt0 (2009)
32. Niu, Y., Chen, H., Hsu, F., Wang, Y.M., Ma, M.: A quantitative study of forum spamming using context-based analysis. In: Proceedings of the 2007 Network and Distributed System Security Symposium, NDSS'07, San Diego (2007)
33. Pang, H., Zhang, J., Mouratidis, K.: Scalable verification for outsourced dynamic databases. Proc. VLDB Endow. **2**(1), 802–813 (2009)
34. Petroni Jr., N.L., Hicks, M.: Automated detection of persistent kernel control-flow attacks. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS'07, Alexandria, pp. 103–115. ACM, New York (2007). doi:10.1145/1315245.1315260
35. Pham, V.H.: eurecom.fr, Honeypot traces forensics by means of attack event identification. http://goo.gl/wGPlV (2009)
36. Pouget, F., Dacier, M.: Honeypot-based forensics. In: Proceedings of the 2004 AusCERT Asia Pacific Information Technology Security Conference, AusCERT'04, Gold Coast (2004)
37. securityfocus.com, Zeus botnet finds hold in Amazon cloud. http://goo.gl/rFjzF (2009)
38. Seifert, C., Steenson, R.: honeynet.org, Capture – Honeypot Client (Capture-HPC). http://goo.gl/u7qJZ (2006)
39. Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., Sekar, V.: Making middleboxes someone else's problem: network processing as a cloud service. In: Proceedings of the 2012 ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '12, Helsinki, pp. 13–24. ACM, New York (2012). doi:10.1145/2342356.2342359
40. Somorovsky, J., Heiderich, M., Jensen, M., Schwenk, J., Gruschka, N., Lo Iacono, L.: All your clouds are belong to us: security analysis of cloud management interfaces. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW'11, Chicago, pp. 3–14. ACM, New York (2011). doi:10.1145/2046660.2046664
41. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesly Longman, Boston (2002)
42. Steinberg, U., Kauer, B.: Nova: a microhypervisor-based secure virtualization architecture. In: Proceedings of the 5th European Conference on Computer Systems, EuroSys'10, Paris, pp. 209–222. ACM, New York (2010). doi:10.1145/1755913.1755935
43. Stone-Gross, B., Holz, T., Stringhini, G., Vigna, G.: The underground economy of spam: a botmaster's perspective of coordinating large-scale spam campaigns. In: Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats, LEET'11, Boston, pp. 25–32. USENIX Association, Berkeley (2011)
44. Szefer, J., Keller, E., Lee, R.B., Rexford, J.: Eliminating the hypervisor attack surface for a more secure cloud. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS'11, Chicago, pp. 401–412. ACM, New York (2011). doi:10.1145/2046707.2046754
45. Szefer, J., Lee, R.B.: Architectural support for hypervisor-secure virtualization. In: Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS'12, London, pp. 437–450. ACM, New York (2012). doi:10.1145/2150976.2151022

46. Ta-Min, R., Litty, L., Lie, D.: Splitting interfaces: making trust between applications and operating systems configurable. In: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI'06, Seattle, pp. 279–292. USENIX Association, Berkeley (2006)
47. Thonnard, O., Dacier, M.: A framework for attack patterns' discovery in honeynet data. Digit. Investig. **5**, S128–S139 (2008). doi:10.1016/j.diin.2008.05.012
48. trustedcomputinggroup.org, TPM specifications version 1.2. http://goo.gl/0IWyy (2011)
49. Wang, Y.M., Beck, D., Jiang, X., Roussev, R., Verbowski, C., Chen, S., King, S.T.: Automated Web patrol with strider HoneyMonkeys: finding Web sites that exploit browser vulnerabilities. In: Proceedings of the 2006 Network and Distributed System Security Symposium, NDSS'06, San Diego. The Internet Society, San Diego (2006)
50. Wang, Z., Jiang, X.: Hypersafe: a lightweight approach to provide lifetime hypervisor control-flow integrity. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy, SP'10, Oakland, pp. 380–395. IEEE Computer Society, Washington, DC (2010). doi:10.1109/SP.2010.30
51. Wei, J., Zhang, X., Ammons, G., Bala, V., Ning, P.: Managing security of virtual machine images in a cloud environment. In: Proceedings of the 1st ACM Workshop on Cloud Computing Security, CCSW'09, Chicago, pp. 91–96. ACM, New York (2009). doi:10.1145/1655008.1655021
52. Wu, B., Davison, B.D.: Cloaking and redirection: a preliminary study. In: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, AIRWeb'05, Chiba, pp. 7–16. Chiba (2005)
53. van Doorn, L.: Trusted computing challenges. In: Proceedings of the 2007 ACM Workshop on Scalable Trusted Computing, STC'07, Alexandria, pp. 1–1. ACM, New York (2007). doi:10.1145/1314354.1314356
54. Xu, S., Yung, M.: Socialclouds: concept, security architecture and some mechanisms. In: Proceedings of the 1st International Conference on Trusted Systems, INTRUST'09, Beijing, pp. 104–128. Springer, Berlin/Heidelberg (2010). doi:10.1007/978-3-642-14597-1_7
55. Zheng, Q., Xu, S.: Fair and dynamic proofs of retrievability. In: Proceedings of the 1st ACM Conference on Data and Application Security and Privacy, CODASPY'11, San Antonio, pp. 237–248. ACM, New York (2011). doi:10.1145/1943513.1943546
56. Zheng, Q., Xu, S., Ateniese, G.: Efficient query integrity for outsourced dynamic databases. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop, CCSW'12, Raleigh, pp. 71–82. ACM, New York (2012). doi:10.1145/2381913.2381927