Towards Abuse Detection and Prevention in IaaS Cloud Computing

Jens Lindemann
University of Hamburg, Germany
Email: jens.lindemann@informatik.uni-hamburg.de

***Abstract*—Cloud computing is frequently being used to host online services. Abuse of cloud resources poses an important problem for cloud service providers. If third parties are affected by abuse, bad publicity or legal liabilities may ensue for the provider. There is an unsatisfactory level of protection against abuse of cloud offerings at the moment.**

In this paper, we analyse the current state of abuse detection and prevention in IaaS cloud computing. To establish what constitutes abuse in an IaaS environment, a survey of acceptable use policies of cloud service providers was conducted. We have found that existing intrusion detection and prevention techniques are only of limited use in this environment due to the high level of control that users can exercise over their resources. However, cloud computing opens up different opportunities for intrusion detection. We present possible approaches for abuse detection, which we plan to investigate further in future work.

I. Introduction

Cloud computing is being used by more and more organisations. Gartner [1] estimates the market for public cloud services in 2013 to total 131 billion US dollars, up by 18.5 percent compared to 2012. It provides a flexible way of hosting online services without the need to invest in an own IT infrastructure. However, cloud services can also be abused either by malicious users or by attackers who have gained control of a user's resources. An example of this could be seen in 2009, when a virtual machine inside Amazon's Elastic Compute Cloud (EC2) was used as a command and control server for the Zeus botnet [2].

Abuse of cloud services is considered to be one of the top nine threats to cloud computing by the Cloud Security Alliance [3]. Despite this, even commercial cloud offerings currently lack sufficient abuse protection [4]. While there have been some research efforts to detect and prevent attacks *on* cloud resources [5]–[8], detecting abuse *of* cloud resources for malicious activities has seen only limited research (as shown in Figure 1). While the work by Doelitzscher et al. ([9], [10]) studies the detection of abuse in IaaS cloud environments in detail, one of their approaches [10] concentrates on the application of anomaly detection techniques on data available from the cloud management system. This leads to the intrusion detection system (IDS) having a rather restricted view of the VMs, as only data such as start and stop times of instances is available. Furthermore, anomaly detection techniques are known to be subject to considerable limitations. Another approach, whose effectiveness is not well understood so far, involves monitoring deviations from 'normal' behaviour as defined by the VM's *user* [9], i. e. the customer.

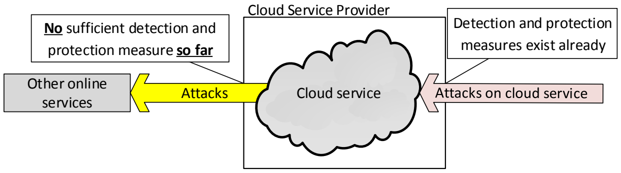
Cloud Service Provider
No sufficient detection and protection measure so far
Detection and protection measures exist already
Other online services
Attacks
Cloud service
Attacks on cloud service

Fig. 1. Research on in- and outbound intrusion detection for cloud computing

Failing to sufficiently protect cloud resources from abuse allows them to be used for malicious activities potentially affecting third parties. The cloud service provider may be rendered responsible for such activities. If third parties are affected by abuse, bad publicity or legal liabilities may ensue.

Our planned research will assess what security measures currently exist against abuse of IaaS cloud computing resources. It will further conceive concepts for abuse detection and prevention, which will be implemented and evaluated. The results will not only be useful to better quantify the risk stemming from the abuse of cloud services, but they will also provide insights regarding the potential and the limitations of cloud abuse detection and prevention techniques in practice.

This paper aims to analyse the current state of intrusion detection and prevention in IaaS cloud environments. It does not aim to provide finished solutions for improvement of intrusion detection and prevention systems in this scenario, but to show possible ways of improving it, which will be examined further in future work.

The remainder of this paper is structured as follows: Section II outlines fundamentals of cloud computing and relates them to the problem. In Section III, we establish what types of activities would be considered abuse based on a study of acceptable use policies of cloud service providers. In Section IV, we analyse abuse detection in IaaS cloud computing, whereas abuse prevention is analysed in Section V. In Section VI, we discuss how events detected by the abuse detection system should be reported. Section VII discusses privacy aspects of abuse detection, before the paper is concluded in Section VIII.

II. Cloud Computing

The NIST distinguishes three service models of cloud computing [11]:

- Software as a Service (SaaS),
- Platform as a Service (PaaS),

- and Infrastructure as a Service (IaaS).

The SaaS model gives users only limited flexibility: it provides access to pre-made applications offered by the provider only. A well-known example of an SaaS offering is Microsoft Office 365. If these services have no unknown vulnerabilities, abuse is fairly easy to detect and prevent on the application level as the feature set offered to users is known and can be restricted by the provider.

PaaS clouds are more difficult to protect because these allow customers to run self-developed applications. As customers are not in direct control of the operating system, host-based intrusion detection systems could be deployed to detect abuse. Detection could also be achieved by looking at calls to the platform's API. Abuse could be prevented by imposing suitable restrictions on this API. The customer may be hosting SaaS services on the platform, which could itself be abused by end-users. Customers would be expected to secure their applications against such abuse, but it may also be detected and prevented by the provider at platform level.

IaaS gives customers significantly more freedom than other service models. It provides them with 'processing, storage, networks, and other fundamental computing resources where [they are] able to deploy and run arbitrary software, which can include operating systems and applications' [11, p. 3]. This is commonly achieved by allowing customers to control a virtual machine, as is the case in the commercial product Amazon EC2 [12] and the open-source frameworks Eucalyptus [13] and OpenStack [14]. Unlike in SaaS and PaaS environments, abuse possibilities of virtual machines in IaaS clouds are comparable with those of physical machines. Detecting and preventing abuse in IaaS is therefore considerably harder than in SaaS and PaaS environments.

Abuse of services is not limited to cloud computing. It is also an issue for providers of dedicated root servers, a traditional hosting model which has been offered for many years, where customers pay for the exclusive use of a server. An important difference between dedicated root servers and cloud services is the price and the elasticity of resources. In comparison, dedicated servers are much more expensive. Moreover, virtual machines in IaaS clouds are much more ephemeral, i. e. they can be created, deployed and removed within seconds. Therefore, abusive behaviour of IaaS cloud services may not only have a much larger impact, it is also much more difficult to track down and prevent.

For the aforementioned reasons, we will concentrate on IaaS clouds in the following.

III. Abuse in Cloud Computing

We consider abuse to be the usage of cloud resources for illegal purposes or activities generally considered to be network abuse. It is irrelevant who actually commands the resources to conduct abuse – this could either be a malevolent, but legitimate customer of the cloud service provider, or somebody who has taken control of a customer's resources.

To establish which activities cloud providers see as abuse and might want to protect against, we conducted a survey of the acceptable use policies of the IaaS offerings of Amazon [15], Google [16], Microsoft [17], Rackspace [18], Softlayer [19], HP [20] and ProfitBricks [21]. An acceptable use policy describes which activities customer's are (not) allowed to conduct within the provider's cloud environment.

We have found the level of detail to vary widely between the surveyed policies. Google's very short policy forms the one extreme, merely giving a rather high-level description of disallowed use. Rackspace's policy is the other extreme case – it explicitly mentions in great detail many types of disallowed use (most of which fall in the high-level categories mentioned in Google's policy).

The results of the survey are shown in Table I. All of the surveyed policies generically disallow illegal activities and sending unsolicited mass e-mail or other messages. All providers also ban explicitly either viruses, 'malware of any kind' or 'harmful content'. Some also specify other types of malware as disallowed, such as trojan horses or worms. Barring Google, all providers also mention unauthorised access to other systems and faking mail headers. All providers but Microsoft explicitly disallow intentional interference with a system. Other activities banned by some providers include activities harmful to the provider's operations, fake IP addresses in network headers and scanning for vulnerabilities of a system. Interestingly, the only German provider in our Survey, ProfitBricks, also forbids the operation of IRC servers, as well as anonymisation, streaming, download and peer-to-peer services and even *linking* to such services from within its cloud, while Rackspace bans 'content that compromises national security'. Softlayer on the other hand does not allow users to *receive* unsolicited e-mail.

The consequences for users in case of a violation of the acceptable use policies vary between providers. Amazon, ProfitBricks, Rackspace and HP state removal of content or disabling of resources violating the agreements or terminating the services. 'SoftLayer reserves the right to take all actions it deems appropriate to comply with applicable laws.' Rackspace and Amazon threaten to report to or cooperate with authorities, in the case of Amazon even 'appropriate third parties.' ProfitBricks threatens users with a penalty of EUR 5100 plus additional compensatory damages. ProfitBricks and Softlayer explicitly state that customers themselves are responsible for ensuring that their use of the services is compliant with applicable law. Rackspace even holds users responsible for violations by 'anyone using [the customer's] services (with permission or unauthorized)'. Google and Microsoft do not mention any consequences for violations in their respective policy.

In the following, we will have a closer look at some of the malicious activities we found mentioned in the acceptable use policies. For most of these, detection techniques already exist. However, existing techniques typically concentrate on detecting inbound attacks. Often, they would also assume that all messages forming part of an attack originate from a single source IP address. However, in a cloud, it would be easy for attackers to spread out their activities over multiple VMs. A cloud service provider would still be able to correlate these messages according to the customer who owns the VMs.

TABLE I. SURVEY OF ACCEPTABLE USE POLICIES

Prohibited activity	Amazon	Google	Microsoft	Rackspace	Softlayer	HP	ProfitBricks
illegal activities (generically)	✓	✓	✓	✓	✓	✓	✓
sending unsolicited mass e-mail	✓	✓	✓	✓	✓	✓	✓
distributing malware of any kind			✓		✓		
distributing harmful content	✓					✓	✓
distributing viruses	✓	✓	implied	✓	✓	✓	✓
unauthorised access to other systems	✓		✓	✓	✓	✓	✓
fake mail headers	✓		✓	✓	implied	✓	✓
intentional interference with a system	✓	✓		✓	✓	✓	✓
activities harmful to the CSP's operations	✓	✓	✓			✓	
fake IP addresses in network headers			implied	✓		implied	✓
scanning for vulnerabilities of a system	✓			✓		✓	✓
IRC servers, anonymisation, streaming, download and P2P services, linking to these							✓
content that compromises national security				✓			
receive unsolicited e-mail					✓		

A. Portscanning

In a portscan, messages are sent to ports on remote hosts to check whether they are open. While this is not an attack in itself, it is often a precursor to one as it can be used to check for vulnerable network services which can subsequently be attacked. An attacker may either scan many ports on a specific remote host or a specific port on many remote hosts. Techniques for detecting inbound portscans on the network level exist (among others [22], [23]) and should be adaptable for detection of outbound scans in a cloud environment.

B. Distribution of Malware

Cloud resources can be used to distribute malware. This may take place in different forms, one of them being through placing it on websites hosted within a cloud. Malware could also be spread by sending it attached to e-mail messages, typically of the spam variety (see Sect. III-D) or by exploiting vulnerabilities on network hosts in order to directly install the malware on the victim machine (see Sect. III-F).

C. (Distributed) Denial of Service

A denial of service (DoS) attack has the goal of compromising the availability of a network service. This is typically achieved by overloading the victim machine by means of sending a large number of requests. If multiple machines are involved in sending the requests, the attack is classified as a distributed denial of service (DDoS) attack. DoS attacks are visible at the network level and could be detected by sensors at the boundary of user VLANs (see Sect. IV). Numerous approaches for detecting inbound DoS attacks exist already [24] and should be adaptable to outbound detection.

D. Sending Unsolicited E-Mail

Cloud infrastructures can easily be used to send a large number of unsolicited e-mail messages (i. e. spam). This may or may not include forged sender information. Techniques for detecting spam messages exist (e. g. [25]), but are typically deployed at the mail server used for either sending or receiving the messages. For our scenario, these would have to be adapted to either monitor messages at the network level or to monitor the mail server software deployed by a customer from outside the VM.

E. Forged Headers

Incorrect sender information in the header of network messages (e. g. a wrong sender IP address in an IP packet or an incorrent sender address in an e-mail) may pose a problem for the receiver and/or the 'sender' of the messages. On the one hand, fake information makes it harder for the victim of an attack to identify the attacker and block the attack – if the attacker constantly changes the 'sender' information, it cannot be used to formulate a rule for blocking attack traffic. On the other hand, forged sender information can be used for amplification attacks, in which the attacker sets the sender information to resemble those of the victim. The attacker then sends requests to a server, which will send the response messages (which will typically be larger than the request) to the victim, who did not request them. DNS servers are particularly popular as amplifiers, as small DNS requests may sometimes generate very large response messages, especially if DNSSEC is being used [26].

F. Exploiting Security Vulnerabilities

Cloud environments can also be used to exploit security vulnerabilities on other network hosts. If the pattern of activity for exploiting a vulnerability is known, it should not only be possible to detect inbound, but also outbound attacks.

G. Botnets

Botnets consist of a large number of compromised hosts which can be commanded by the botnet 'owner' to launch attacks against other network hosts [27]. The compromised hosts are referred to as 'bots' and are typically controlled by a command and control server, as described by Strayer et al. [27], although a peer-to-peer structure is also possible [28]. Both bots as well as command and control servers could be run within cloud environments. A survey of some existing botnet detection techniques has been composed by Feily et al. [29].

While botnets are not specifically listed in the surveyed acceptable use policies, they are still something that providers should aim to protect themselves against as they are typically being used for conducting (in the case of bots) or supporting (in the case of command and control servers) other types of abuse.

1) Command and Control Servers: Command and control servers are used to control a botnet. They take commands from the botnet 'owner' and communicate these to the individual bots, which will then execute the commands. If any information is requested from the bots, it will be collected by the command and control server, where they can be collected by the botnet 'owner'. A command and control server of the Zeus botnet has been found to be operating within Amazon's EC2 in 2009 [2].

2) Bots: VM instances may also be used as bots. This may either be due to a benevolent customer's VM getting infected with malware, but also due to a malevolent user deliberately installing the bot software.

The bot software itself may be detectable by looking at the inner workings of the VM: it will be present in memory and on the virtual hard drive and may also be present in the OS's process list (although many bots try to disguise themselves by manipulating the OS process information). A bot may also be detected by detecting its symptoms: it will communicate with a command and control server and may be used to launch attacks (such as DoS or spamming as described above).

IV. Detecting Abuse in an IaaS Environment

In the previous section, we have shown which types of activity would be considered abuse in an IaaS environment. We will now review existing security techniques in order to evaluate the potential of their application for cloud computing abuse detection. The treatment will specifically highlight opportunities and limitations in order to identify gaps in research.

Classical intrusion detection systems cannot be directly employed for the detection of abuse in cloud environments. On the one hand, these systems are typically geared towards detecting intrusions, i. e. attacks originating from outside the monitored network targeting machines within it. Furthermore, detection software running on the monitored host itself (i. e. within virtual machines) cannot be relied upon in this scenario, as users could easily tamper with or disable it due to having full control of the VM. Its behaviour will thus have to be monitored from the outside.

The possible placement of different IDS sensor types for abuse detection in an IaaS infrastructure with isolated virtual networks for each user is illustrated in Figure 2.

One way this can be achieved is by monitoring network traffic originating from VMs using a network-based IDS, such as Snort [30]. Although these are mostly used to detect intrusions *into* a network or system, they can also be used to detect attacks launched *from within* the network, provided suitable signatures (in case of signature-based systems) or training data (in case of anomaly-based systems) are available to the system. This is referred to as 'outbound intrusion detection' [31] or 'extrusion detection' [25].

If user networks are not isolated from each other, network-based IDS would have to be placed either with each individual VM **(1b)** or at the border of the provider's network to the Internet **(1c)**. While the IDS would be simpler to deploy in the latter case, it would be unable to detect attacks originating from a VM within the cloud environment targeting another VM within it.

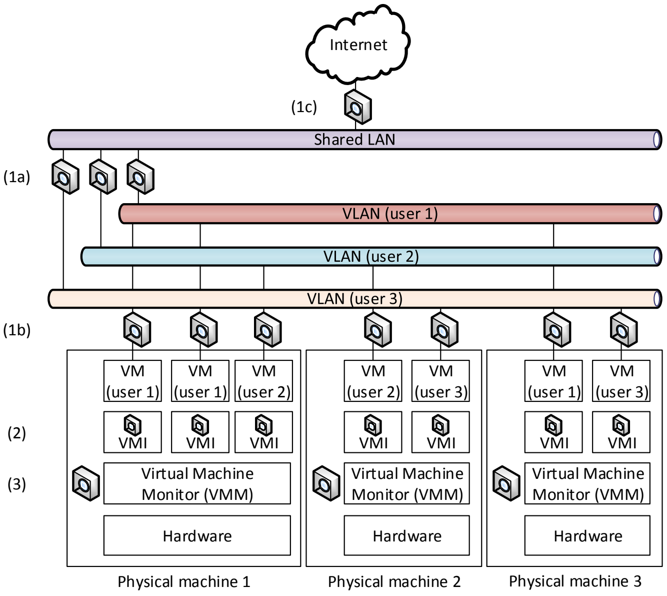

Fig. 2. Placement of sensors

In an environment where each user has a virtual network (VLAN) to himself which is isolated from other customers' networks (as recommended by Hao et al. [32] and Chowdhury et al. [33]), the IDS could instead be placed at the border of each individual user's virtual network with the public part of the network **(1a)**. While the placements described above (1b, 1c) would also be possible, (1a) provides the best compromise between visibility of malicious traffic and ease of deployment. Compared with deploying an IDS with each individual VM, it would not be possible to detect attacks a customer's VM launches on another VM on that customer's virtual network, but responsibility for preventing this can easily be left to the customer as no third parties are affected by such attacks.

While network-based IDS are relatively simple to deploy, they can only examine the network traffic – which might not always be clearly classifiable as legitimate use or abuse, e. g. due to encryption. Host-based IDS on the other hand cannot be deployed within a VM as a malicious user could easily disable or modify these. The technique of 'virtual machine introspection' (VMI), as presented by Garfinkel and Rosenblum [34], allows to monitor the inside of a VM without a user being able to interfere **(2)**. A VMI-based IDS can directly inspect the machine state and could be used to detect malicious software running on the host. This could include malware used to hijack a VM as well as attack tools willingly executed by a malicious user. It is important to note that no assumptions must be made regarding the semantics or trustworthiness of the VM when using VMI on customer VMs [35]: Users will be in full control of the virtual machine and can manipulate any of the software running within. This applies even to users using VM images supplied by the provider.

As a consequence, VMI techniques are challenging to use in situations where the user is not cooperating with the operator of the VMI system: They require knowledge about the memory structures used by the operating system running within the VM. While there are relatively few Windows kernel versions being used, there is a huge number of different Linux kernels (which will potentially use different memory structures) due to frequent updates and the possibility of users compiling custom kernels. Existing VMI software, such as LibVMI [36], typically requires the Linux kernel's *System.map* file, which contains the kernel symbol table and enables the VMI software to find system information in memory. However, a provider cannot assume a malevolent user to provide the correct version of this file. While there are memory analysis techniques which enable memory analysis without prior knowledge of the location of data structures in memory, these are relatively slow [37] and would make real-time monitoring infeasible. The lack of satisfactory solutions means that this is a fruitful area of future research.

Another source of data, which could be used for intrusion detection, is information available from the virtual machine monitor **(3)**. One example of data provided by the VMM is the creation and destruction time of a VM, as used for intrusion detection by Doelitzscher et al. [10].

Some of the presented techniques may also be of use for detecting abuse of machines in large networks of companies or universities. Especially if the use of personal devices is permitted, there will be a large heterogeneity of devices, with the organisation having no control over both soft- and hardware of these devices. This limits outbound intrusion detection to analysing the network traffic. On devices actually controlled by the organisation, traditional host-based sensors may be used if the user's rights are sufficiently restricted to prevent tampering with the sensor.

V. Prevention of Abuse in an IaaS Environment

In addition to the *detection* techniques as outlined in the previous section, research should also encompass *prevention* of abuse. It is currently unclear how an abuse prevention system could be constructed and how effective it would be in preventing abuse in cloud computing. Both technical as well as non-technical measures may play a role in preventing abuse.

A. Technical Measures

1) Intrusion Prevention Systems (IPS): Intrusion prevention systems (IPS) exist as an extension of intrusion detection systems. Similar to IDS, existing implementations cannot be directly employed for preventing abuse in cloud computing. Thus, an 'abuse prevention system' is needed as a new class of prevention system. Similar to IPS, it would be desirable for this system to automatically take corrective action on detection of abusive activity by the abuse detection system. Possible reactions might be to shut down the system or restrict its network connection.

False positive reports by the abuse detection system relating to perfectly legitimate events are a challenge for automatic prevention. If a VM is shut down or taken off the network following a false positive, customers would very likely be dissatisfied and might decide to move their business to a competitor. Additionally, the provider might be in breach of a service level agreement and incur financial penalties. Allowing an abuse prevention system to take automatic action is thus feasible only if an IDS report can almost certainly be attributed to abuse.

2) Filtering of Network Traffic: Another technical measure to prevent abuse would be a firewall. This could block network transmissions according to its rule-set. Unlike an intrusion prevention system, it will not take into account any information provided by IDS sensors when deciding which network transmissions are to be blocked. One example of malicious traffic that could easily be filtered is traffic with fake header information (see Sect. III-E). A cloud service provider can easily prevent its customers from sending IP packets containing fake sender IP addresses by filtering network packages if their sender address does not match the customer's network, as described in RFC 2827 [38]. Similar filtering could be set up for other protocols.

3) Logging: Alternative to an outright prevention of abuse, the system could also trigger an extensive logging procedure to allow later forensic investigations of the suspicious activities. While this would allow abuse to take place at first, it at least makes legal prosecution possible at a later stage. This could also help providers in case they were held legally responsible for abuse conducted by their customers.

Compared to constant comprehensive logging of all activity within the cloud, this approach would reduce storage requirements considerably. Also, user privacy (cf. Sect. VII) is improved, as detailed activity logs will only be created for customers suspected of conducting abuse.

B. Non-technical Measures

1) Acceptable Use Policies: Acceptable use policies form part of the contract between customer and provider and can help to deter abuse. Especially if certain behaviour is not illegal, but the provider still does not want it to be conducted within its environment, it is necessary to formalise this behaviour as disallowed. A policy cannot outright prevent any abuse from happening, but it makes clear to users that a certain type of use will not be tolerated and allows a provider to take necessary steps for stopping such use, i. e. by terminating the user's services. An acceptable use policy can also make it easier to implement technical prevention measures, as users who subsequently complain about the blocked type of use not working can be referred to the policy which they have accepted as part of their contract. For a survey of the typical contents of acceptable use policies see Sect. III.

2) Account Verification: In the past, many – especially smaller – cloud service providers offered trial accounts and did not require any verification apart from an e-mail address. As e-mail addresses are very easy to obtain, it was possible to automatically create a large number of accounts with such providers. These accounts could then be used for malicious purposes. Such an attack was demonstrated by Salazar and Ragan at the 2014 RSA Conference USA [39]. This type of attack can easily be prevented by requiring a higher level of authentication for new accounts, e. g. by asking for a phone number and authenticating it by means of sending a text message containing a verification code. A study by Thomas et al. [40] has shown that phone verification significantly increases the price of an account on the black market, leading to the conclusion that it is indeed harder to create such accounts.

We surveyed the same providers as in Section III and found that for these large providers, verification measures beyond a simple e-mail confirmation are in place. Signing up for an account with Amazon, Microsoft Azure, ProfitBricks, Rackspace and Softlayer requires providing a phone number, which can be used for phone verification. Amazon, Microsoft and Google additionally require new customers to provide their billing information even if they do not currently intend to use any billed services. HP occasionally asks users to send copies of their utility bills as proof of address.

3) Financial Incentivisation: Another measure to deter abuse would be to implement an incentive system. This could be implemented in form of a deposit made by a customer before starting to use the services. If users are subsequently found to be abusing the services, the service is terminated and the provider gets to keep the deposit. Szefer and Lee propose such a system using Bitcoin to make the deposit, which they call 'BitDeposit' [41].

VI. Reporting

As not all abusive activity can be prevented by the techniques mentioned in Section V, reports by the abuse detection system may have to be manually examined by a cloud service provider's staff. Manual examination of each individual event would be too costly, as there will potentially be a large number of events reported by the IDS – especially in large cloud environments. Security metrics will have to be devised, aggregating events to allow easier identification of malicious users or abused VMs.

A possible way of doing this would be to assign a *reputation* to each individual VM based on events reported by the IDS. An event which can almost certainly be attributed to abuse should have a higher impact on the reputation than an event which could also be attributed to legitimate use. If an event of the same type occurs again and again, this should have a higher impact than an event which occurs only once.

Assigning a reputation to individual users by further aggregating the events of all their VMs might also be helpful. If one of a customer's VMs has a very low reputation whereas all others have a high reputation, a hacked VM rather than a truly malicious user may be the cause. This user's reputation should still be relatively high. If all of a user's VMs had a low reputation, their reputation should also be low. This would signal that there is either a general security problem with all of the user's VMs (e. g. due to an exploitable vulnerability in some piece of software used within all of these) or that the user himself is malicious.

VII. Privacy Considerations

Careful consideration should also be given to what impact abuse detection and prevention have on privacy of customers and users of a customer's services. A cloud service provider trying to detect and prevent abuse of its infrastructure needs to look at both network traffic and behaviour of (virtual) machines. While network traffic either between different VMs within the cloud or with machines outside the cloud may or may not be encrypted by the customer, information will normally have to be unencrypted while it is being processed on the (virtual) machine itself. Using virtual machine introspection, a cloud service provider would be able to see this information in its unencrypted form, giving rise to privacy concerns if this technique is employed with wrong intentions.

To establish how cloud service providers currently handle data within customer VMs and storage as well as communications flowing through their network, we surveyed the same providers as in Section III. We found that there was not a lot of information to be found about this aspect of privacy – neither in the terms of use, the acceptable use policies, nor the privacy policies. Also, for all surveyed providers, the privacy policy was not specific to the cloud offerings. Policies seemed to be tailored to fit all online operations of the respective companies including information about aspects such as cookies used on the website, not giving much detail about which customer data would be accessed by the providers for operational purposes, such as security monitoring. While Microsoft also provides a white paper on 'Protecting Data and Privacy in the Cloud' [42] and has a section on privacy and security in its service terms [43], these are mostly concerned with how data is being protected from third parties and does not detail whether and how data is being accessed by Microsoft for security monitoring. HP and Amazon rather 'cloudily' reserve the right to investigate any suspected violation of their acceptable use policy and report this to law enforcement agencies, again not detailing what information would actually be accessed in this case and what would still be off-limits.

Our survey has shown that providers currently lack awareness of the privacy issues connected to them being able to access not only their customers' network communications (which customers could protect by encrypting it), but also any data being processed in VMs and saved in the cloud.

VIII. Conclusion

We have motivated the need for detecting abuse in IaaS cloud computing and have shown that this has so far not been addressed satisfactorily by research. The results of our planned research may serve as a valuable building block in the ongoing struggle to improve the security of IaaS cloud services, on which many service providers may rely on in the future. The results will not only be useful to better quantify the risk stemming from the abuse of cloud services, but they will also provide insights regarding the potential and the limitations of cloud abuse detection and prevention techniques in practice.

References

[1] Gartner, "Gartner Says Worldwide Public Cloud Services Market to Total $131 billion," Press release, February 28th 2013, http://www.gartner.com/newsroom/id/2352816.

[2] C. Babcock, "Zeus Bot Appears in EC2 Cloud, Detected, Dismissed"," InformationWeek, December 11th 2009, http://www.informationweek.com/cloud-computing/zeus-bot-appears-in-ec2-cloud-detected-d/229203959.

[3] Top Threats Working Group, "The notorious nine – cloud computing top threats in 2013," Cloud Security Alliance, Tech. Rep., February 2013, https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf.

[4] BAE Systems Detica, "Security Research Blog – botCloud: an emerging platform for cyber-attacks," October 28th 2012, http://baesystemsdetica.blogspot.de/2012/10/botcloud-emerging-platform-for-cyber_785.html.

[5] S. Roschke, F. Cheng, and C. Meinel, “Intrusion Detection in the Cloud,” in *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2009, Chengdu, China, 12-14 December, 2009*. IEEE, 2009, pp. 729–734. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/DASC.2009.94

[6] S. N. Dhage, B. B. Meshram, R. Rawat, S. Padawe, M. Paingaokar, and A. Misra, “Intrusion detection system in cloud computing environment,” in *Proceedings of the ICWET ’11 International Conference & Workshop on Emerging Trends in Technology, Mumbai, Maharashtra, India, February 25 - 26, 2011*, B. K. Mishra, Ed. ACM, 2011, pp. 235–239. [Online]. Available: http://doi.acm.org/10.1145/1980022.1980076

[7] J. Arshad, P. Townend, and J. Xu, “An automatic intrusion diagnosis approach for clouds,” *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 286–296, 2011.

[8] I. Gul and M. Hussain, “Distributed cloud intrusion detection model,” *International Journal of Advanced Science and Technology*, vol. 34, pp. 71–82, 2011.

[9] F. Doelitzscher, C. Reich, M. Knahl, and N. L. Clarke, “Incident detection for cloud environments,” in *The Third International Conference on Emerging Network Intelligence (EMERGING)*, 2011, pp. 100–105.

[10] F. Doelitzscher, M. Knahl, C. Reich, and N. L. Clarke, “Anomaly Detection in IaaS Clouds,” in *IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom 2013, Bristol, United Kingdom, December 2-5, 2013, Volume 1*. IEEE Computer Society, 2013, pp. 387–394. [Online]. Available: http://dx.doi.org/10.1109/CloudCom.2013.57

[11] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” NIST, Special Publication 800-145, September 2011.

[12] “Amazon Elastic Compute Cloud (EC2),” http://aws.amazon.com/ec2/.

[13] “Eucalyptus website,” http://www.eucalyptus.com/.

[14] “Openstack website,” http://www.openstack.org/.

[15] Amazon Web Services, “AWS Acceptable Use Policy,” November 2nd 2011, http://aws.amazon.com/de/aup/.

[16] Google, “Google Cloud Platform: Google Cloud Platform Acceptable Use Policy,” https://cloud.google.com/terms/aup?csw=1.

[17] Microsoft, “Microsoft Azure Service Terms,” http://azure.microsoft.com/en-us/support/legal/services-terms-nov-2014/, September 2014.

[18] Rackspace, “Legal Information – Global Acceptable Use Policy,” August 1st 2014, http://www.rackspace.com/information/legal/aup.

[19] Softlayer, “Legal – Aceptable Use Policy,” http://www.softlayer.com/acceptable-use-policy.

[20] HP, “Acceptable Use Policy,” http://www.hpcloud.com/acceptable-use-policy.

[21] ProfitBricks, “Allgemeine Geschäftsbedingungen – Rechtliche Hinweise,” https://www.profitbricks.de/agb.

[22] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, “Fast Portscan Detection Using Sequential Hypothesis Testing,” in *2004 IEEE Symposium on Security and Privacy (S&P 2004), 9-12 May 2004, Berkeley, CA, USA*. IEEE Computer Society, 2004, pp. 211–225. [Online]. Available: http://dx.doi.org/10.1109/SECPRI.2004.1301325

[23] S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical Automated Detection of Stealthy Portscans,” *Journal of Computer Security*, vol. 10, no. 1/2, pp. 105–136, 2002. [Online]. Available: http://content.iospress.com/articles/journal-of-computer-security/jcs154

[24] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, “Denial-of-Service Attack-Detection Techniques,” *IEEE Internet Computing*, vol. 10, no. 1, pp. 82–89, 2006. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/MIC.2006.5

[25] R. Clayton, “Stopping Spam by Extrusion Detection,” in *CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*, 2004. [Online]. Available: http://www.ceas.cc/papers-2004/172.pdf

[26] A. Cowperthwaite and A. Somayaji, “The futility of DNSSec,” in *Proceedings of the 5th Annual Symposium on Information Assurance, Albany, NY, USA*, June 2010.

[27] W. T. Strayer, R. Walsh, C. Livadas, and D. E. Lapsley, “Detecting Botnets with Tight Command and Control,” in *LCN 2006, The 31st Annual IEEE Conference on Local Computer Networks, Tampa, Florida, USA, 14-16 November 2006*. IEEE Computer Society, 2006, pp. 195–202. [Online]. Available: http://dx.doi.org/10.1109/LCN.2006.322100

[28] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, “Peer-to-Peer Botnets: Overview and Case Study,” in *First Workshop on Hot Topics in Understanding Botnets, HotBots’07, Cambridge, MA, USA, April 10, 2007*, N. Provos, Ed. USENIX Association, 2007. [Online]. Available: https://www.usenix.org/conference/hotbots-07/peer-peer-botnets-overview-and-case-study

[29] M. Feily, A. Shahrestani, and S. Ramadass, “A Survey of Botnet and Botnet Detection,” in *The Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009, 18-23 June 2009, Athens/Glyfada, Greece*, R. Falk, W. Goudalo, E. Y. Chen, R. Savola, and M. Popescu, Eds. IEEE Computer Society, 2009, pp. 268–273. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/SECURWARE.2009.48

[30] “Snort website,” http://www.snort.org/.

[31] S. Mandujano and A. Galván, “Outbound intrusion detection,” *Proceedings of the International Computer, Communications and Control Technologies, CCCT*, vol. 4, pp. 68–73, 2004.

[32] F. Hao, T. V. Lakshman, S. Mukherjee, and H. Song, “Secure Cloud Computing with a Virtualized Network Infrastructure,” in *2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud’10, Boston, MA, USA, June 22, 2010*, E. M. Nahum and D. Xu, Eds. USENIX Association, 2010. [Online]. Available: https://www.usenix.org/conference/hotcloud-10/secure-cloud-computing-virtualized-network-infrastructure

[33] N. M. M. K. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2009.10.017

[34] T. Garfinkel and M. Rosenblum, “A Virtual Machine Introspection Based Architecture for Intrusion Detection,” in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA*, 2003, pp. 191–206.

[35] M. Christodorescu, R. Sailer, D. L. Schales, D. Sgandurra, and D. Zamboni, “Cloud security is not (just) virtualization security: a short paper,” in *Proceedings of the first ACM Cloud Computing Security Workshop, CCSW 2009, Chicago, IL, USA, November 13, 2009*, R. Sion and D. Song, Eds. ACM, 2009, pp. 97–102. [Online]. Available: http://doi.acm.org/10.1145/1655008.1655022

[36] vmitools Google Code Wiki, “LibVMIIntroduction – An introduction to LibVMI,” https://code.google.com/p/vmitools/wiki/LibVMIIntroduction.

[37] Z. Lin, J. Rhee, X. Zhang, D. Xu, and X. Jiang, “SigGraph: Brute Force Scanning of Kernel Data Structure Instances Using Graph-based Signatures,” in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*. The Internet Society, 2011. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/11/pdf/3_3.pdf

[38] P. Ferguson and D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,” IEEE, Tech. Rep. RFC 2827 (BCP 38), May 2002.

[39] O. Salazar and R. Ragan, “CLOUD NINJA – Catch Me If You Can!” Presentation at RSA Conference, February 24-28, 2014, San Francisco, USA. http://www.rsaconference.com/writable/presentations/file_upload/ht-r01-cloud-ninja-catch-me-if-you-can_.pdf.

[40] K. Thomas, D. Iatskiv, E. Bursztein, T. Pietraszek, C. Grier, and D. McCoy, “Dialing Back Abuse on Phone Verified Accounts,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, G. Ahn, M. Yung, and N. Li, Eds. ACM, 2014, pp. 465–476. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660321

[41] J. Szefer and R. B. Lee, “BitDeposit: Deterring Attacks and Abuses of Cloud Computing Services through Economic Measures,” in *13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013, Delft, Netherlands, May 13-16, 2013*. IEEE Computer Society, 2013, pp. 630–635. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CCGrid.2013.102

[42] Microsoft, “Protecting Data and Privacy in the Cloud,” http://go.microsoft.com/?linkid=9694913&clcid=0x409.

[43] ——, “Online Services Terms,” January 2015, http://www.microsoftvolumelicensing.com/Downloader.aspx?DocumentId=8248.