The Role and Security of Firewalls in IaaS Cloud Computing

Jordan Cropper, Johanna Ullrich, Peter Frühwirt, Edgar Weippl
SBA Research
Vienna, Austria
Email: (firstletterfirstname)(lastname)@sba-research.org

Abstract—**Cloud computing is playing an ever larger role in the IT infrastructure. The migration into the cloud means that we must rethink and adapt our security measures. Ultimately, both the cloud provider and the customer have to accept responsibilities to ensure security best practices are followed. Firewalls are one of the most critical security features. Most IaaS providers make firewalls available to their customers. In most cases, the customer assumes a best-case working scenario which is often not assured. In this paper, we studied the filtering behavior of firewalls provided by five different cloud providers. We found that three providers have firewalls available within their infrastructure. Based on our findings, we developed an open-ended firewall monitoring tool which can be used by cloud customers to understand the firewall's filtering behavior. This information can then be efficiently used for risk management and further security considerations. Measuring today's firewalls has shown that they perform well for the basics, although may not be fully featured considering fragmentation or stateful behavior.**

I. Introduction

The concept of cloud computing is almost as old as the idea of distributed networks itself, but has only in the last decade become a truly feasible option for consumers and enterprises [1]. What began as organisations attempting to generate side revenue from idling data centres has itself become enormous business and arguably one of the most important changes in general computing since the mainframes of the past evolved into the now ubiquitous personal computer. Cloud computing has brought enormous changes in the landscape of computation. The flexibility it provides, that 1000 hours on one machine costs the same as an hour on 1000 machines, allows companies, research bodies and even private individuals to enjoy as necessary a level of computing power previously accessible only to organisations with large budgets and equally large workloads. It would not be undue to claim that cloud computing may bring with it the next great revolution in the world of computation.

Part of this outsourcing of computation, whether on a lower level with Infrastructure as a Service (IaaS), or on a higher level as with Platform/Software as a Service (PaaS / SaaS), is a shift in security responsibility. Cloud providers and customers making use of cloud computing both have significant parts to play in keeping cloud computing secure. Many cloud providers offer default, basic security products along with recommendations for running client level security solutions.

With security being a major factor in whether organisations choose to embrace or reject the cloud, it is important to consider how both traditional and novel security concerns are addressed in a cloud environment. The firewall, long the cornerstone of network security, has an important role to play in cloud services, but the change of environment brings with it challenges that may not be fully addressed by the current generation of cloud firewalls.

In a traditional networking environment, e. g., a corporate environment, a perimeter firewall protects the internal nodes from attacks from the outside; the firewall as well as the internal nodes are managed by the same instance and considered as benign. Cloud computing however blurs this concept of internal and external. External parties are able to easily rent virtual instances residing within the perimeter and run their applications there, whilst the vendor remains unaware, i. e., there is no information on used protocols, ports, etc. in terms of networking. This leads to the conclusion that a new way of handling firewalls is necessary in the environment of cloud computing.

We investigate the role and security of firewalls in Infrastructure-as-a-Service (IaaS) cloud computing using the examples of the most popular providers *Amazon EC2*, *IBM Softlayer*, *Microsoft Azure*, *Google Compute Cloud* and *Rackspace* as we believe them to be adequate representatives of the public cloud landscape. Our work is twofold:

Today's Firewalls: We investigated the architecture, default configuration and configuration parameters. We rented our own instances from the aforementioned cloud providers, investigated the default configuration, the possibilities of configuration and the relevant documentation. Furthermore, we probed firewall filtering by means of our firewall testing tool. This provides an insight into the quality of firewalls in current IaaS cloud computing.

Firewall Test Tool: The cloud computing field is changing rapidly. Vendors are constantly introducing new functionality or improving current offerings. Similarly, it is likely that firewalls' scope will also change in the near future. Thus, we developed a tool which enables cloud consumers to check their firewall on their own. We provide this tool on an open-source base to the public. Further, the tool is extendible with additional test scenarios.

Our results shed light on aspects which are not accessible to average cloud tenants, because providers refrain from disclosure of their infrastructure and applied technologies. Official documentation on firewalls is rare, either barely scratching the surface or in the style of recipes. We address our results' implications on securing user instances, and consider improvements for the provider as well as the instance owners.

Section II provides background on the three cloud service models *IaaS*, *PaaS* and *SaaS*; major *IaaS* vendors and a typology of firewall types. Section III debates the changes arising from cloud computing for the application of firewalls, and provides an overview on how major vendors operate their firewalls. As these vendors typically refrain from providing in-depth information on their infrastructure, we develop in Section IV an extensible tool and test cases for firewall probing based on previous architectural considerations. The succeeding Section V presents the gained results. Section VI discusses the results and infers practical advice for gaining secure virtual instances. The paper is concluded in Section VIII.

II. Background

In these subsections, we provide background information on the general cloud service models *IaaS*, *PaaS*, *SaaS*; an overview of today's cloud computing vendors and a typology on firewalls.

A. Cloud Service Models

Cloud computing is defined as *"a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."* by the *National Institute of Standards and Technology (NIST)* [2]. It encompasses the five characteristics of *on-demand self-service*, *broad network accesses*, *resource pooling* and *rapid elasticity*.

However, clouds appear in various ways and are classified into the three service models *SaaS*, *PaaS* and *IaaS*, as described in [2]. The service model assigns responsibilities to consumer and cloud provider in different combinations: While with *Software as a Service (SaaS)*, the consumer uses a provider's applications running in the cloud by means of a web browser or a certain client; in *Platform as a Service (PaaS)*, a cloud provides a platform including programming languages, libraries, etc. to run consumer-created applications. *Infrastructure as a Service (IaaS)* cloud providers however offer resources to the consumer where she is able to run her applications including an operating system of choice.

Figure 1 shows a representation of a cloud system consisting of four layers: *infrastructure*, *hypervisor*, *operating system* and *application*. As can be seen, the lower two layers are always the provider's responsibility, while the upper layers differ. Since in *SaaS* a provider basically offers an application for use, a consumer's responsibility for maintenance is very limited; in *PaaS* the application layer is split into the *platform*, maintained by the provider, and the developed app of the consumer. As *IaaS* allows the consumer to choose and configure an operating system, the responsibility of maintenance belongs with the consumer.

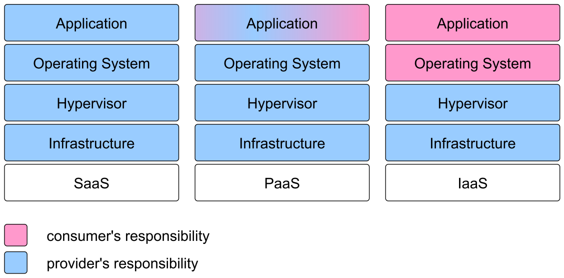

Fig. 1. Responsibility assignment of consumer and provider according to service model.

In the Infrastructure as a Service (IaaS) model, Virtual Machines (VMs, often referred to as Instances) are paid for by usage, generally in terms of uptime hours, GB of storage and bandwidth used. Users in the IaaS model have control of the whole VM from the kernel layer upwards: the choice of Operating System tends to be left to the consumer, although most cloud providers offer a number of pre-specified options, usually Linux based, for easy installation. The user may then operate the VM as though it was any other remote server, with administration over SSH, hosting relational databases or providing public facing Web services over the internet. Many providers allow for a number of instances to be easily connected inside a group, simulating a LAN in the cloud.

IaaS is the least abstracted level of cloud computing. The next level up, Platform as a Service (PaaS), offers the consumer a slightly more removed but still comparatively customisable cloud computing platform. The operating system is usually set or the choice is much more limited than with IaaS, and the vendor offers a number of underlying applications, APIs or other tools in order to simplify the use of the cloud platform. The definition of PaaS is somewhat broad, and certain PaaS products, such as parts of Microsoft's Azure cloud suite, span more than one category.

The final level we see is Software as a Service, SaaS. This involves offering a product directly from the cloud, though this does not preclude having certain locally installed elements to add extra features or provide a better service. Notable current SaaS examples include SalesForce, Dropbox and numerous others. It is worth noting that a web application offered with the SaaS model could easily be hosted on a IaaS platform, with numerous levels of abstraction inbetween.

B. Cloud Computing Vendors

The current cloud computing marketplace is a young market and still relatively diverse [3], but a number of larger vendors hold considerable market share. This paper looked at a number of different cloud computing vendors who all enjoy a considerable level of market share in the cloud computing space, summarised in Table I. Consistent with the origins and high initial setup costs of a cloud computing platform, they are all large, well recognised names in the technology sphere. Providers

TABLE I
A COMPARISON OF CLOUD COMPUTING VENDORS IN THE STUDY.

Provider	Service Type
Amazon Web Services	IaaS
IBM (Softlayer)	IaaS / PaaS
Microsoft Azure	IaaS / PaaS
Google	IaaS / PaaS
Rackspace	IaaS

are ranked in approximately decreasing size of market share, although the way in which cloud computing operates makes it extremely difficult to make accurate comparisons. Companies are often under no obligation to report the extent of their revenue which comes from cloud computing, and the use of different metrics makes direct comparison difficult. However it is universally acknowledged that Amazon is the largest player in the cloud computing space, with as much cloud revenue and capacity as the majority of the competition combined [4]. Many providers offer a number of service types or are difficult to classify. Google's Compute Engine uses an IaaS model much the same as Amazon's Elastic Compute Cloud (EC2), which shares little with Google App Engine, the company's other offering firmly in the PaaS area. Parts of the Azure cloud suite are also PaaS offerings, but their Virtual Machines are a standard IaaS product.

C. *Typology of Firewalls*

A Firewall is a component residing at the border of two networks which inspects traffic going from one to the other. For best security results, no traffic should bypass the device to guarantee inspection of the whole communication process. Reflecting the historic development of these security tools, firewalls are grouped into three types [5].

The most simple are *Packet Filters*. They hold a number of rules which define allowed or disallowed traffic; while the first is forwarded to the other network, the latter is dropped. Rules include source and destination address and ports identifying the service. The decision is based solely on the currently inspected packet and no connection context is maintained. The advantages of this general approach allow the firewall to handle a high level of service without any respective in-depth knowledge; this also allows the integration of newly developed services the firewall is not already aware of. Beyond, the simple architecture has only a limited need for resources in comparison to other types. On the other hand, they do not have full insight into the communication as a number of protocols are stateful, i. e., packets may be in general valid, but not at this certain point in communication.

Stateful Filters go beyond this approach and maintain a context for connections. This allows the use of previously seen packets as part of the decision regarding the current packet [6]. This way it is possible to guarantee that certain packet types are protocol-compatible; not only considering the message format, but also their point in the communication process, although at the cost of requiring more memory. Typically, real-world stateful filters also use simple packet filtering for the best effect.

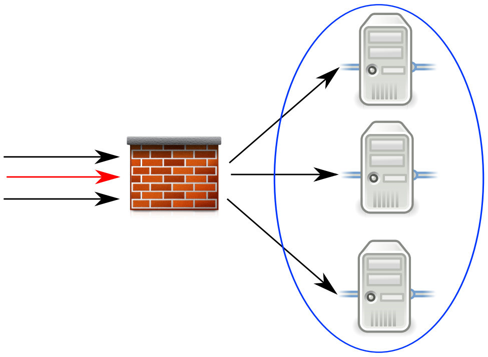

Fig. 2. Simplified diagram of a traditional network: data ingress/egress is restricted and protected by a firewall customised to the specific setup.

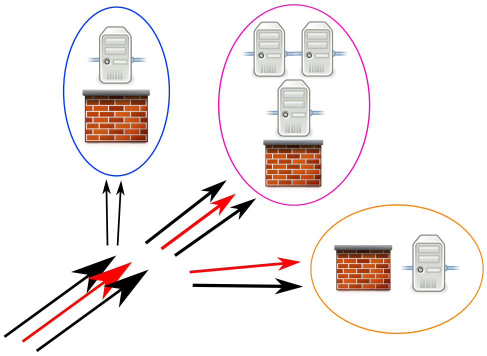

Fig. 3. A cloud network deals with different challenges. The firewalls, provided by the vendor, must be capable of protecting their individual groups of instances without advance knowledge of the expected traffic.

The third step in firewall evolution are *Application Layer Filters* which have special code for every application. While this allows even deeper inspections and is thus also considered to be more secure, the drawback is the specialized code for every application. This generally leads to the situations that only the best known services are available, without any extension for niche or novel protocols, and is even more costly in terms of computing. Currently, *Deep Packet Inspection* is all the rage, which also scans the packets' payloads, e. g. for malware.

For all kind of firewalls, the permissive and the restrictive approach can be applied. Describing the default behavior, a permissive firewalls allows all traffic to pass unless specified otherwise; restrictive configurations drop everything unless specified. Assuming in general everything as evil, the latter is far more common.

III. SECURING AN IAAS CLOUD

In this section, we discuss the challenges of positioning a firewall in clouds and how this is handled by major vendors.

A. Towards Firewalls in Clouds

Firewalls are amongst the older but still highly important tools in security, and continue to play an important role both in home networks and in enterprise [7]. They began as simple packet inspectors, looking at the source and destination to determine whether to pass or drop them. Modern firewalls are powerful, fully featured security tools capable of matching incoming traffic against complex rules to protect the vast array of modern networks against numerous threats and malicious actors. The main function of a firewall is to protect a network against external threats and restrict internal actors even after they may have been able to compromise a node inside the network. Firewalls inspect traffic passing into and out of the network and match it against rules set by the network administrator. Many firewalls can perform packet inspection, but it is not their task to detect malicious payloads: rather, they can block attempts to connect to a network by any means other than those explicitly allowed, and hinder exfiltration of information from inside the network.

Whilst remaining a critical security tool, firewalls have evolved as network threats have changed and Moore's Law has provided them with greater defensive resources yet simultaneously benefited their attackers. Firewalls have changed from the early packet filters to rely mostly on rulesets, packet inspection and intelligent behaviour. Most modern firewalls are stateful, keeping track of connections and rejecting packets that are not part of a pre-existing link. Many modern firewalls also work with intrusion detection systems, and are capable of packet inspection in order to drop packets with suspicious characteristics. The majority of firewalls are able to log events and raise alerts if certain conditions are met.

IaaS brings with it challenges in implementing this kind of firewall, and although this model is suitable for the client VM, external firewalls must be configured differently. They offer limited customisability beyond the blocking of certain ports, and many provide no form of logging at all. They provide only basic network security and do not include the more advanced features found on more traditional large scale firewall products. The fact that an IaaS provider cannot know ahead of time what the client intends to use their machines for means that traditional firewalls are unsuitable, as shown in the comparison of Figure 2 and Figure 3. This is not a problem faced by SaaS or (to a slightly more limited extent) by PaaS providers. In SaaS the provider knows exactly the needs and requirements of the application, and can tailor the security set up accordingly. The situation for PaaS is slightly more complex given that the vendor does not have complete knowledge of the use of the platform, but still knows what mechanisms are supplied for the client to use. As a result, the most interesting case is IaaS, as here the firewall providers must adapt the most to the changes brought by the cloud environment; flexibility may be one of cloud computing's most heralded advantages, but it may also present problems when we consider the security of the cloud environment. After all, securing any border without knowing in advance what is to be expected is considerably harder than when one has exact knowledge, as in SaaS, of what should be allowed in or out.

The distinction between the various types of cloud services must also consider the *location* of the cloud servers. Some large organisations may choose to host their own private cloud, accessible to employees but not the general public. This contrasts with public clouds, such as those offered by Amazon and other cloud providers, whose business it is to offer their computing power to paying customers. Private clouds tend to function on the SaaS or more rarely PaaS model, and can be subject to much stricter security protocols than public clouds, since both the users and uses of the cloud are known in advance, and the organisation has far greater control over the entire system.

Cloud vendors have approached this problem by providing a middle ground security product that can perform basic functions but little else. Generally, the firewalls are set to deny everything by default and individual services must be enabled by the client. The firewalls, in order to simplify operation, rarely provide any other features, with the exception that they allow and often encourage the restriction of access to certain IP address ranges. This ability to whitelist certain IPs and services, combined with a default deny approach results in a relatively user friendly and efficient security product, but one that lacks many of the features commonly found on local firewall products. Of course, the IaaS model allows the client to run their own security software on individual instances.

B. Default Setup at Major Vendors

Based on the considerations above, we investigated the availability of firewalls and their scope of functionality. Out of the providers in Table I, Rackspace and IBM (Softlayer) could not be tested as they do not offer a firewall, the subject of this paper. We looked at Amazon's Elastic Compute Cloud (EC2), Microsoft's Azure Cloud Platform and Google's Compute Engine.

Both EC2 and Azure provide user configurable firewalls around instances or groups of instances, referred to as Security Groups in EC2 and Endpoints in Azure, designed to offer a basic level of security. Usability is evident as both are easy to configure whilst offering good default security by blocking inbound traffic (except SSH) by default, preventing unsecured instances being exposed to the Internet unintentionally.

In terms of user experience, both EC2 and Azure provide simple web interfaces for adding and removing firewall rules. The Google Compute Engine web interface allows the user to enable or disable HTTP(S) traffic, but any further configuration must be done using the downloadable Google Cloud SDK via the terminal. This allows scriptable and programmatic control of the firewall, but makes configuration slightly more difficult for users who are less used to the command line interface. However, both types of interface allow the same level of configuration: rules can be added to allow certain ports and protocols, and a range of allowable IP or single addresses can be specified. All the interfaces surveyed used IPv4 rather than v6.

Provider	Firewall available	Web interface	Documentation available	Inbound/outbound separated	Configuration parameters	Default inbound config	Default outbound config
Amazon EC2	✓[a]	✓	✓	✓	transport layer protocol, port (single or range), source (anywhere or single IP)	SSH (TCP, 22) from anywhere [b]	* to anywhere
Google Compute Engine	✓	✗	✓	✗[c]	transport layer protocol, port (single or range), source (range)	ICMP SSH (TCP, 22) from anywhere RDP (TCP, 3389) from anywhere UDP and TCP (all ports) from same /16	* to anywhere (not configurable)
IBM (Softlayer)	✗						
Microsoft Azure	✓[d]	✓	✓	✗[e]	transport layer protocol port (single) source (range)	SSH (TCP, 22) from anywhere	* to anywhere (not configurable)
Rackspace	✗						

[a] called *security groups*

[b] Amazon however suggests to change the address to your own IP after instance launching.

[c] configuration of inbound only

[d] called *endpoints*

[e] configuration of inbound only

TABLE II
DEFAULT SETTINGS OF FIREWALLS FROM MAJOR CLOUD SERVICE PROVIDERS

Amazon provides separated configuration on inbound and outbound traffic, while it seems that others provide inbound configuration only and in general all kind of outbound traffic is allowed to pass. Amazon however provided only inbound configuration in the older EC2-classic configuration, so it might also be included in the future at the other providers. A overview comparison can be seen in Table II.

It must be finally noted that this paper concentrates on *vendor provided software firewalls*. Some providers, such as IBM Softlayer, offer the option of a dedicated hardware firewall, which is beyond the scope of this paper and has more in common with traditional enterprise networks than the cloud environment. Similarly, we are not looking at local, instance or OS level firewalls, an area covered by traditional firewall products such as iptables and other commerical offerings. We are primarily interested in the new challenges that the cloud brings to security, and how vendor provided perimeter firewalls can work towards mitigating these threats.

IV. FIREWALL TESTING TOOL

In this section, we present our firewall test tools. First, architectural aspects are considered, followed by our implementation and developed test cases.

A. Architectural Consideration

Although firewalls are offered by cloud providers, none of them requires deep insights into the architecture. The available documentation as indicated in Table II is of limited use due to being more like a handbook. From a consumer's perspective, the following is known: (1) We are able to configure ports and ranges, mostly from a web interface for a group or a single instance. (2) Every consumer is able to make a configuration for her own needs. However, the providers refrain from stating whether the consumer can configure partially a perimeter firewall this way or a hypervisor-based firewall. While Amazon's approach of security groups lets one think of a more de-centralized approach like the second alternative, Microsoft Azure naming of *end points* lets one rather believe in a configurable perimeter.

In the case of a hypervisor-based solution, we are further not able to infer whether the firewall is placed in front of the hypervisor, or afterwards directly before the guest operating system, see for both alternatives Figure 4.

While this seems minor at first sight, the order may have a serious impact: Hypervisors are not obliged to deliver packets on to the guests without any alteration. For example, they can decide to reassemble fragments before forwarding due to performance issues. One has to be aware, that the connection

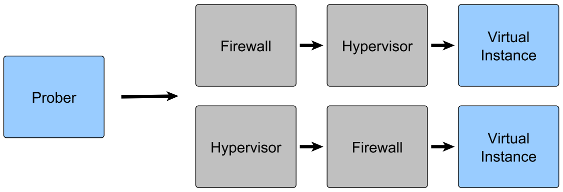

Fig. 4. Alternatives for hypervisor-based firewalls

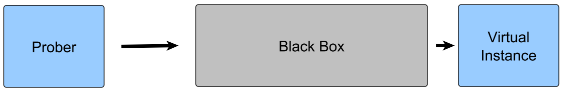

Fig. 5. Assumption for black box firewall testing in cloud environments

between hypervisor and guest is not constricted to typical network requirements like a MTU (Maximum Transmission Unit). Thus, in the first case the packets investigated by the firewall are how they travelled on the network, while they might have been altered by the hypervisor in the second alternative. This has impacts on the way firewall rules have to be defined as well as the complexity of the firewall mechanism itself, e. g., a hypervisor reassembling the fragments releases the firewall from doing so.

As a consequence of these uncertainties, we have chosen to establish a black box model including the firewall independent of its location in combination with the hypervisor functionality, see Figure 5. We are able to access the prober as well as the virtual instance for measurement and therefore define the following general test approach:

1) start the capturing tool on the receiver,
2) establishment of pre-conditions (e. g. performing a TCP handshake),
3) sending of test packet(s) from the prober to the virtual instance,
4) and observe whether packet has been captured by the sniffing tool.

We preferred this over invoking a response from the virtual instance as this would add more chances for failure. By means of this set-up and specific test scenarios, we aim to answer the following question:

- *Which aspects (e.g. protocols and respective fields) are filtered?*
  This issue encompasses the ISO/OSI layer the firewall is inspecting and which protocol header field is therefore included. Conceivably, we will heavily work with the network layer protocol IP and transport layer's TCP and UDP. This also include layer-dependent mechanisms, e. g., fragmentation.
- *Do cloud firewalls reveal stateful or stateless behavior?*
  A knowledge of firewall behavior enables consumers to estimate their extent and the residual measures to gain needed the required level of protection, e. g., by additional host firewalls. As today's state-of-the-art are stateful firewalls, it would be unusual to find plain packet filters. Thus, we will investigate the "extent of statefulness" of the tested implementations.

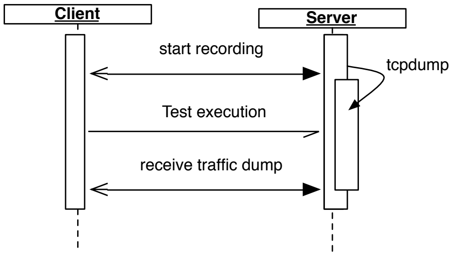

Fig. 6. Communications between the test client and the server component

- *Are application layer filters implemented?*
  Application layer filters would imply a large intervention into a consumer's traffic and restrict their free choice of ports. On the other hand, it would provide more control on what is going into the cloud. We limit our research however on application layer filters for HTTP as we believe that this would be the first choice beneath SSH and FTP for providers for implementation, due to the vast number of web servers in the cloud.

B. Implementation

In this section we present a feasible implementation of our evaluation approach. As a precondition we presume that at least one port is reachable by the testing client, which is used later as a feedback and communication channel between the test client and the server. Our approach consists of two components: A testing client that executes test cases against a cloud instance and a server component that is running on the cloud instance and records the traffic during test execution. A test case implements a concrete scenario outlined in Section IV-C. As a test oracle the test case can evaluate the network traffic that reached the server instance in order to decide if the test case was blocked by the cloud firewall. In best case the test case hides a randomised token, e.g. by adding it to the payload, within the test connections and seeks afterwards in the traffic dump for this token. If the token is present, the test case fails because the request has passed the cloud firewall.

Figure 6 gives a detailed view of the communication between the test client and the server component running on the cloud instance. In detail the client connects to the server component running on a cloud instance using a plain TCP connection. This channel is used for coordination and callbacks. A single test case is stored as a separate file in a directory. The client loads a test case from this directory and creates a test case instance. Afterwards the client tells the server component over the communication channel to start recording the network traffic using *tcpdump*. Subsequently the client executes the test case and request the tcpdump of this session from the server component. The test case instance now evaluates the success of this test iteration using the dump received from the server. The client repeats this steps till all test cases are executed. We released the implementation including all test cases[1].

C. Test Cases

Within this architecture, we implemented test cases according to Table III. With this test cases, we looked to determine the responses of the firewalls to specific inputs, in order to classify it and to examine how it responded to certain well known security threats. The test cases are separated in groups A to E depending on the functionality. Some test scenarios, e. g., 1 or 7, are benign scenarios to test for functionality and connection. For a detailed explanation, we refer to the test cases provided with our testing tool.

We chose to examine the firewalls' responses in the following areas:

A: Internet Protocol (IP)
This test cases cover basic aspects of the Internet Protocol and a variety of illegitimate field combinations. These scenarios test the extent of packet investigation on the network layer. At present, we limit the test cases to IPv4 as IPv6 is at the moment not widely supported. By now, IBM Softlayer and Rackspace – both not offering firewalls – are the only providers supporting IPv6 natively at their virtual instances.

B: Fragmentation
The firewalls' responses to fragmented packets, both normally generated and "malicious" overlapping packet fragments, were measured in a series of tests. The same operating system was used for both tests, and kernel level measuring tools were employed to ensure the firewall, not the OS, was responsible for the observed behaviour. Test Cases 8 to 10 are modelled according to [8] and transformed to IPv4. We used both fragmented ICMP and UDP packets, except for Azure where only UDP was used, as Azure does not permit ICMP.

C: Basic TCP and UDP
Group C scenarios cover basics aspects of the transport-layer protocols TCP and UDP, i. e., invalid source and destination ports as well as invalid checksums.

D: TCP Flagging
Group D contains a number of packets with illegitimate TCP flag combinations, but they are tested without a previously established connection. As part of this, we considered how far the firewalls could be regarded as stateful.

E: Stateful Behavior
Stateful behaviour, i. e., in combination with an established connection, is tested with test cases of group E. In general, all stateless test cases should be repeated within a connection. However, we performed our tests in an iterative manner and decided to refrain from repetitions as the test cases passed already successfully without any connection.

F: Application Layer
The last group F targets the application layers firewalls. We targeted HTTP as it seems to be one of the most heavily used protocols in clouds.

These areas have been well addressed by traditional firewall products. Since our cloud perimeter firewalls have only limited customisation options, it is not possible to test some of the modern attacks against firewalls, designed to evade intrusion detection systems that are often part of or function with modern firewalls. The absence of any form of firewall logging means that the testing took on a black box approach, with limited information available even with full access to the firewall configuration menus.

V. Firewall Responses

The results of our test cases are presented in Table IV. We used default configuration, and additionally opened two ports: First, we allowed TCP traffic for our testing tool's synchronization, and second we opened another for TCP and UDP traffic for probing. The responses of all three of the tested firewall products were found to be consistent with each other in the majority of the use cases.

A: Internet Protocol (IP)
The most problematic of these is that the Azure platform disallows the ICMP protocol (see test case 1), meaning certain test cases originally basing on ICMP had to be rewritten to using UDP as a transport layer protocol. However, test cases 2 to 6 imply that invalid IP headers are filtered en-route and firewalls perform well.

B: Fragmentation
While benign fragments as in test cases 7 and 11 passed the firewall, overlapping fragmentation without a terminating fragment (more fragment flags set to 0) are filtered by all the products tested. However, the responses differs with overlapping fragments which are terminated (test case 8). While Amazon and Google let them pass, Azures filters them. At least, their actions are consistent when the fragments are received in reverse order (test case 10). Looking at the packet captures from the receiver, we saw that fragmented packets are reassembled before reaching the virtual instance, i. e., by the firewall or the hypervisor. We believe that this is done for performance reasons.

C: Basic TCP and UDP
Test Cases 12 to 17 show that malformed headers of transport layer protocols are able to pass the firewall, and only in the case where the destination port equals zero are the packets unable to pass. The latter behavior however is obvious when following a port-based filter approach as there is no rule for this port.

[1] https://gitlab.sba-research.org/johanna/cloud-firewall-monitoring-tool

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Amazon	✗	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Google	✗	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Azure	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗

TABLE IV
RESULTS OF TEST CASES PER CLOUD PROVIDER (✓ FILTERED, ✗ UNFILTERED)

ID	*Name*
A	**IP**
1	Valid ICMP Request
2	Checksum invalid
3	Invalid packet length
4	Invalid header length
5	Reserved flag unequal zero
6	IP protocol number unequal ICMP, TCP or UDP
B	**Fragmentation**
7	Benign fragmentation
8	Overlapping Fragmentation
9	Overlapping fragmentation without terminating fragment
10	Overlapping fragmentation in reverse order
11	Tiny Fragments
C	**Transport Layer Protocols**
12	Invalid Source Port (TCP)
13	Invalid Source Port (UDP)
14	Invalid Destination Port (TCP)
15	Invalid Destination Port (UDP)
16	Invalid Checksum (TCP)
17	Invalid Checksum (UDP)
D	**TCP Flags**
18	Null Packet (no flags)
19	SYN, FIN
20	SYN, FIN, PSH
21	SYN, FIN, RST
22	SYN, FIN, RST, PSH
23	FIN
E	**Stateful behaviour**
24	SYN in established connection
25	ACK without ACK number
F	**Application layer**
26	Improper HTTP request

TABLE III
IMPLEMENTED TEST CASES

D: TCP Flagging

The firewalls of Amazon and Google allowed packets with strange or illegal flag combinations in any case [9], while Microsoft's filtered these packets. The null packets, i. e. no flags at all, could however pass in every case; likewise FIN packets without a previously established connection.

E: Stateful Behavior

All three firewalls allowed the sending of SYN flags in an established connection as well as packets containing the ACK flag without an acknowledgement number.

F: Application Layer

No sign of an application layer firewall was found.

Additional findings

Each platform disallows the use of certain protocols: Google's Compute Engine does not allow SMTP (port 25) or SMTP over SSL (465 and 587); Azure does not permit ICMP packets to be sent or recieved. In a similar way, EC2 also blocks the sending of SMTP mail by default, though this can be enabled by submitting a support request and using a set of Amazon APIs (SES).

None of the services tested responded to either crafted UDP packets or to a UDP scan using the Nmap port scanning tool. The result was the same no matter if the ports were opened or closed in the firewall. We see from this that the firewall does not send an ICMP "Port Unreachable" notification when the port is closed. This is expected for Azure, which disallows completely the use of ICMP, but perhaps less so for Amazon EC2 and Google Compute Engine.

VI. DISCUSSION AND IMPLICATIONS

All the firewalls tested work in the same way to perform basic security functions, and they all suffer from a lack of customisability and the ability to log events. The important role the firewalls play in securing these systems highlights how they have been adapted to the new environment of cloud computing, albeit at the cost of some features considered standard in other use cases.

Although the firewalls correctly implement best practices by discarding invalid IP packets, they fail to do so when considering the TCP flags. Packets with suspicious TCP flag combinations (such as SYN-ACK) can be immediately dropped, as there is no situation where they would be transmitted in normal usage. There appears to be no reason for the firewalls not to drop such packets in the same manner as with invalid IP packets. However, it is possible that the firewall rules are deliberately set to allow a wide range of packets, since the IaaS providers cannot know in advance exactly what each instance will be used for. Again, a greater level of customisability for the firewall rules would be beneficial here.

Whilst the nature of cloud computing means that it would be more difficult to enable firewall logging, we believe that this is potentially one of the larger features missing from the current generation of cloud based firewalls. If full logging was not possible, the collation of reduced statistics in the control panels for the respective cloud services would still be an improvement over the current situation.

We have seen how the firewalls examined behave in a number of real world conditions, and must thus consider how this fits into the wider framework of cloud security. Securing a cloud instance cannot be left solely to the perimeter firewall, a task for which it is neither designed nor suitable. Cloud security must utilise a defense-in-depth approach, briefly outlined here.

The perimeter firewall plays a critical role as the gatekeeper, permitting only the required traffic for the VM to perform its required task. In the case of a static webserver this could be only HTTP and SSH for administration; in the case of a number of instances in a large group deployed by a major corporation, tens or hundreds of cloud machines could have their own specific tasks. The perimeter firewalls allow control of the traffic into and between groups of instances. **Local firewalls** can be employed on the level of individual instances to offer fine-grained control currently lacking in the larger perimeter firewalls. Operating System firewall products such as the Linux iptables would be appropiate. They can also be paired with **intrusion detection** systems to watch for attempts at malicious access to the cloud instances. Finally, the endless task of keeping software patched and up to date contributes significantly to the security of the cloud ecosystem.

Cloud providers can and do play a major role in keeping the cloud secure, and are incentivised to do so: after all, an attack against a customer's cloud is also an attack against the provider. By providing security updates as soon as they are released and a strong, configurable cloud firewall, cloud vendors help to protect both their customers and themselves. We recommend that any cloud vendor not offering these basic measures should strongly consider doing so.

The three firewall products behaved, with the exception of minor variation imposed by and specific to each vendor, in almost exactly the same way. They all perform the basic job of the perimeter firewall and have the same advantages and limitations, but overall they perform well in the basic task of providing strong perimeter security for cloud instances. Default deny settings and the ability to easily restrict access by IP address makes the firewalls useful for providing a baseline level of security in the cloud.

As for the firewall products, we would recommend that customers be allowed finer control over the traffic that is blocked and permitted. We have identified that the firewalls also permit a number of strange combinations of TCP flags which should not be transmitted in normal usage and can most likely be safely blocked. Further packet inspectation capabilities, beyond the exisiting ability to block overlapping packets, would be a powerful new tool, although admittedly not as directly useful to many firewall users.

Finally, the addition of logging to the firewalls, perhaps as an optional component, would significantly enhance their usefulness and capabilities. It would be useful not only for general traffic statistics, but also to give customers a good indication of any security threats that their cloud platform may be facing. Automated log scanning is easily possible and would give customers a good overview of the details of the traffic hidden by presenting them only with bandwith usage and billing figures.

VII. Related Work

Related work can be divided into three major areas.

Firewalls: Firewalls have been used since the early days of the Internet and are well understood. A number of research papers cover aspects of firewalls in traditional network of the pre-cloud era. The scopes encompasses a wide variety of topics like firewall design, e. g. [10], [6], [11], or quantitative studies on configuration errors [12], [13] to name a few. Additionally, *Request for Comments (RFCs)* provide guidelines kept in a more practical way, see for example [14], [15].

General Cloud Computing Security: Most previous work on cloud computing security has focused on the engineering challenges and the business benefits of the cloud [16]. Of the studies which explicity consider the security implications of the cloud, many papers consider an overview of challenges in cloud security [17] [18], look more at the regulatory and liability issues surrounding moving data into the cloud [19] [16], or focus on theoretical implementations of security infrastructure [20]. Work such as [20] has looked at how firewalls may be adapted for cloud environments, but did not examine any current implementations in comparison.

Cloud Identification: There are numerous papers investigating cloud behavior in the wild. While considering the cloud as a black box, they aim to reveal internals on functionality and frequently deal with security issues. [21] investigate the security of the Amazon Image ecosystem. [22], [23], [24], [25] investigated the impact of resource sharing among multiple virtual instances, especially targeting a shared network interface card (NIC). [26], [27] consider file duplication for fault-tolerance and local distribution of file storing in clouds. [28] revealed internal behavior of the popular cloud storage solution *Dropbox*.

However, none of them has looked at the firewall implementations of major cloud providers and examines the current state of the art as we did. As far as our research has shown, this is one of the first papers examining the current, in the wild performance of cloud firewalls.

VIII. Conclusion

In this paper we have examined the current state of firewalls in cloud computing, focusing on EC2 system offered by Amazon, Google's Compute Engine and Microsoft's competing Azure cloud. We conclude that firewalls remain a critical part of the security infrastructure even as computing moves into this new environment. We found that although basic, the functionality of the firewalls provides mostly sufficient protection, given also that they can be augmented by a client based security setup.

We find that the ease of customisation and secure default settings of these firewalls has contributed to keeping cloud computing comparatively secure. We believe that the lack of certain firewall features is one of the main areas for enhancement in this field. Finally, we examined the implications of this research and how it applies to modern cloud security.

Acknowledgment

This research was funded by COMET K1, FFG - Austrian Research Promotion Agency.

References

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, p. 13, 2009.

[2] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[3] "Cloud computing - the business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176 – 189, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923610002393

[4] J. Bort, "Amazon is crushing ibm, microsoft and google..." www.businessinsider.com/amazon-cloud-beats-ibm-microsoft-google-2013-11, Accessed: 2014-09-04.

[5] S. Bellovin and W. Cheswick, "Network firewalls," *Communications Magazine, IEEE*, vol. 32, no. 9, pp. 50–57, Sept 1994.

[6] M. Gouda and A. Liu, "A model of stateful firewalls and its properties," in *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, June 2005, pp. 128–137.

[7] K. Ingham and S. Forrest, "A history and survey of network firewalls," *University of New Mexico, Tech. Rep*, 2002.

[8] A. Atlasis, "Attacking ipv6 implementation using fragmentation," *Black-Hat Europe*, 2012.

[9] K. K. Frederick, "Abnormal ip packets," http://www.symantec.com/connect/articles/abnormal-ip-packets, Accessed: 2014-08-09.

[10] M. Gouda and X.-Y. Liu, "Firewall design: consistency, completeness, and compactness," in *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, 2004, pp. 320–327.

[11] A. Liu and M. Gouda, "Diverse firewall design," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 9, pp. 1237–1251, Sept 2008.

[12] A. Wool, "A quantitative study of firewall configuration errors," *Computer*, vol. 37, no. 6, pp. 62–67, June 2004.

[13] ——, "Trends in firewall configuration errors: Measuring the holes in swiss cheese," *Internet Computing, IEEE*, vol. 14, no. 4, pp. 58–65, July 2010.

[14] D. Newman, "Bencharking Terminology for Firewall Performance," RFC 2647, August 1999.

[15] N. Freed, "Behaviour of and Requirements for Internet Firewalls," RFC 2979, October 2000.

[16] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[17] K. Curran and S. Carlin, "Cloud computing security," *Int. J. Ambient Comput. Intell.*, vol. 3, no. 1, pp. 14–19, Jan. 2011. [Online]. Available: http://dx.doi.org/10.4018/jaci.2011010102

[18] A. Tripathi and A. Mishra, "Cloud computing security considerations," in *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on*, Sept 2011, pp. 1–5.

[19] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, Sept 2008, pp. 5–13.

[20] S. Yu, R. Doss, W. Zhou, and S. Guo, "A general cloud firewall framework with dynamic resource allocation," in *Communications (ICC), 2013 IEEE International Conference on*, June 2013, pp. 1941–1945.

[21] S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider, "Amazonia: When elasticity snaps back," in *18th ACM Conference on Computer and Communications Security*, 2011, pp. 389–400.

[22] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, "Detecting co-residency with active traffic analysis techniques," in *ACM Cloud Computing Security Workshop*, 2012, pp. 1–12.

[23] V. Varadarajan, T. Kooburat, B. Farley, T. Ristenpart, and M. M. Swift, "Resource-freeing attacks: Improve your cloud performance (at your neighbor's expense)," in *ACM Conference on Computer and Communications Security*, 2012, pp. 281–292.

[24] A. Herzberg, H. Shulman, J. Ullrich, and E. Weippl, "Cloudoscopy: Services discovery and topology mapping," in *ACM Cloud Computing Security Workshop*, 2013, pp. 113–122.

[25] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, "On detecting co-resident cloud instances using network flow watermarking techniques," *International Journal of Information Security*, vol. 13, no. 2, pp. 171–189, 2014.

[26] K. D. Bowers, M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "How to tell if your cloud files are vulnerable to drive crashes," in *18th ACM Conference on Computer and Communications Security*, 2011, pp. 501–514.

[27] K. Benson, R. Dowsley, and H. Shacham, "Do you know where your cloud files are?" in *3rd ACM Cloud Computing Security Workshop*, 2011, pp. 73–82.

[28] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," in *USENIX Security*, 8 2011.