Nephology: Characterizing the Security of Base Images across Cloud Providers

Narong Chaiwut[0009−0003−3374−6255] and Nick Nikiforakis[0000−0002−9366−357X]

[1] Stony Brook University, Stony Brook NY 11794, USA
[2] {nchaiwut, nick}@cs.stonybrook.edu

Abstract. Cloud computing has become a foundational component of modern software development, with organizations increasingly relying on cloud platforms to deploy and manage applications. While prior studies have examined cloud providers from performance, cost, and networking perspectives, the security posture of *default configurations* (i.e. the settings users inherit before making any modifications) remains largely unexplored. These configurations define the initial attack surface and can introduce security weaknesses that persist into production deployments. In this work, we present the first large-scale, systematic analysis of default security weaknesses across cloud providers. We evaluate 45 public cloud platforms and analyze four major Linux distributions (Ubuntu, Fedora, CentOS Stream, and Debian) resulting in configuration data from 307 instantiated virtual machines. Our measurement pipeline collects data such as kernel versions, end-of-support operating system availability, firewall behavior, SSH authentication policies, system hardening results, and privilege-related configurations. Using the Common Weakness Scoring System (CWSS), we convert these observations into comparable security weakness scores across providers. Our findings reveal substantial variation in default security postures. Major providers such as AWS, Google Cloud Platform, and Azure consistently exhibit lower security weakness scores, while the majority of providers fall into a moderate-weakness category, often due to outdated OS images, permissive firewall defaults, or insecure authentication configurations. These results highlight the need for increased transparency and standardization in default cloud security practices and provide actionable insights for both cloud consumers and platform operators.

Keywords: cloud computing, default configurations, cloud provider security, operating system security, CWSS

1 Introduction

Cloud computing has experienced rapid growth since its inception, with global market projections exceeding $1 trillion [3]. Public cloud platforms specifically, allow developers to spin up virtual infrastructure in seconds, ranging from single-core virtual machines to deliver lightweight web applications, to multi-core, multi-GPU ones to respond to AI workloads. As developers increasingly rely

on cloud infrastructure for application deployment and delivery, understanding the underlying security of these public cloud platforms is essential.

Existing work on cloud comparison has focused primarily on performance and cost efficiency. For example, CloudCmp [14] evaluates the performance and pricing characteristics of major providers. On the security side, prior research has examined side-channel attacks in multi-tenant environments [26–29], co-residency detection [24], and security weaknesses during initial deployment windows [12]. None of these studies systematically analyzed the broader security posture of cloud provider *default configurations* across operating system images.

Despite widespread reliance on cloud infrastructure, the security posture of newly provisioned instances is often assumed rather than verified. Default configurations provided by cloud platforms play a critical role in determining the initial attack surface exposed to users. Misconfigurations such as outdated kernels, permissive (or non-existent) firewall rules, and insecure authentication settings can exist *before* any user-driven customization occurs. These "default weaknesses" present an unnecessary window of vulnerability for newly provisioned VMs and may even persist into production deployments, if they go by unnoticed. As cloud adoption accelerates, understanding the inherent security properties of provider-supplied images becomes increasingly important, yet this aspect has received little systematic study.

In this paper, we conduct the first large-scale comparative analysis of default security weaknesses across cloud providers. We evaluate 45 public cloud providers across four major Linux distributions: Ubuntu, Fedora, CentOS Stream, and Debian. At a high level, we aim to answer the following question: beyond pricing and performance differences, is a base VM image of one cloud provider, just as secure as the corresponding image of another provider?

During the first phase of our analysis, we inspect each provider's web interface to catalog the available operating system images and versions, recording the range provided by each provider and the availability of images past their official end of support. Next, we provision the most commonly offered OS versions and automatically collect system-level data from 307 instantiated virtual machines. This data includes kernel versions, firewall behavior, SSH authentication settings, system-hardening results, and other settings relevant to server security.

To quantify and compare the security posture of these cloud providers across their base images, we use the Common Weakness Scoring System (CWSS) to derive individual weights for each measured weakness which we then combine into a total security score, for each distribution-provider combination.
Our main contributions are summarized as follows:

- We conduct the first large-scale, systematic analysis of default security configurations across 45 public cloud providers.
- We cover four of the most popular Linux distributions, collecting base configurations and kernel data from 307 Linux instances.
- We apply the CWSS framework to compute comparative security weakness scores, enabling cross-provider evaluation based on measurable configuration properties.

All components of our data-collection framework and associated data are available at `https://github.com/pragseclab/nephology`

2 Background

Cloud Security and Configurations Cloud environments allow developers to deploy virtual operating systems within minutes, either through automation tools [8, 9] or via cloud provider web interfaces. While software vulnerabilities remain a major attack vector, insecure default configurations can also significantly expand the attack surface [18]. Provider-side misconfigurations, such as permissive firewall defaults or outdated OS images, can create opportunities for attackers to compromise cloud systems.

Security weaknesses often originate from configuration decisions made during the provisioning phase. For example, insecure SSH settings, such as enabling root login or password authentication, can provide direct entry points for attackers. Even when software is free of vulnerabilities, these configuration weaknesses can significantly weaken the overall security posture of a cloud instance. Although experienced users may eventually mitigate such issues (e.g., updating packages, deploying firewalls, or hardening SSH settings), attackers can exploit the first minutes/hours/days of a VM's lifetime before it is secured [12]. Configuration weaknesses can also easily propagate because many providers offer ready-to-use OS templates that users repeatedly deploy without modification, potentially inheriting insecure defaults. This highlights the need for a systematic approach to measure and compare the security weaknesses of cloud providers.

Security Scoring System One of the most widely adopted scoring systems in the security community, used by the National Vulnerability Database (NVD) [20], is the Common Vulnerability Scoring System (CVSS) [10], which assigns severity scores from 0 to 10 to reported vulnerabilities. Instead of CVSS, we employ the Common Weakness Scoring System (CWSS) [6], which is designed to evaluate weaknesses rather than individual vulnerabilities. While CVSS effectively quantifies the severity of specific vulnerabilities, CWSS provides greater flexibility for prioritizing and comparing structural weaknesses across systems [23] by yielding scores on a scale from 0 to 100 based on weakness severity. This property aligns more closely with our goal of assessing cloud provider configurations and their relative security postures.

3 Methodology

In this section, we present our pipeline for analyzing the security of cloud providers' base OS images. We first describe the cloud provider selection process, followed by the procedures used to collect configuration data. Finally, we explain how the collected data are processed, scored, and compared to derive security weakness levels.

3.1 Cloud Selection

To build our dataset, we first compiled a comprehensive list of candidate providers across multiple regions (e.g., the United States, Europe, and Asia). We then applied an initial filter to select providers offering elastic, pay-as-you-go compute services that allow users to create and destroy VMs on demand. Most providers bill by instance-seconds or instance-hours (e.g., AWS and Linode), while a few charge flat monthly fees (e.g., Hivelocity and Hostinger).

Next, we manually registered and completed the verification process for each provider. Some required additional steps, such as Know Your Customer (KYC) verification or credit card authorization. A few providers required phone interviews (e.g., IBM), which we were unable to complete.

As a result, we obtained a final set of 45 cloud providers (listed in the Appendix). This set represents a diverse sample of commonly available cloud platforms across different geographic regions, including both globally popular providers and less prominent regional ones.

3.2 Data Collection

To collect the data required for our analysis, we employed two complementary approaches: operating system image data collection and dynamic data collection.

OS Images Data Collection We manually visited each cloud provider's web interface to gather information about the operating system images they offer, focusing on four major Linux distributions: Ubuntu, Fedora, CentOS Stream, and Debian. For each provider, we recorded the available distributions, version numbers, and the corresponding end-of-support (EOS) status as defined by the upstream maintainers. When multiple minor versions were available for a given distribution, we selected the latest minor release within each major version. These OS image listings form the foundation of our dynamic data collection stage, in which we provision instances using the selected images to obtain additional system-level and configuration-level information.

Dynamic Data Collection Dynamic data consists of information collected from provisioned instances after deployment. For each instance, we developed a probing tool to automatically collect kernel information, system configuration details [21], kernel hardening results [1], and firewall responsiveness through

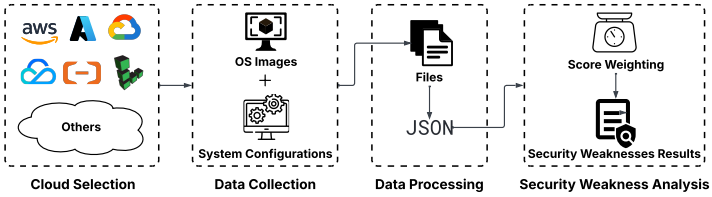


Fig. 1: *Overview of our measurement pipeline, consisting of four main stages: cloud selection, data collection, data processing, and security weakness analysis.*

internal port enumeration and external scanning tools. Given the large number of providers and OS variants (45 providers × multiple versions), collecting this data manually would be time-consuming and difficult to repeat. Therefore, we automated the pipeline using three workflows: *I.Provisioning Tools:* We used Terraform [8] to provision instances and inject data-collection scripts via `cloud-init` [5]. When `cloud-init` was unsupported, we connected using Terraform's SSH provisioner and uploaded the scripts for data collection. *II.Web API:* For providers lacking Terraform support, we used their available REST APIs to create instances. Once deployed, we developed and used an Ansible playbook [9] to automatically connect to the instance and execute our probing tool. *III.Fallback:* When both Terraform- and API-based provisioning were unsupported or failed, we manually created instances and used the aforementioned Ansible playbook to probe the newly-created VM.

All collected logs were stored on instances and later uploaded to our server for data processing.

3.3 Data Processing

Given the diversity of the collected data, further analysis required transforming the raw logs into a structured format. To achieve this, we developed several parsers to extract and convert the information into a consistent representation. The main categories of processed data are as follows:

System-Related Data: This category follows the structure provided in [21]. For our analysis, we focused on the metrics used to identify security weaknesses: kernel information, number of exploits, number of packages, number of SUID files, and number of SGID files.

System Hardening: We extracted the number of system hardening failures from the kernel hardening assessment [1], capturing only the failed checks for use in our weakness scoring.

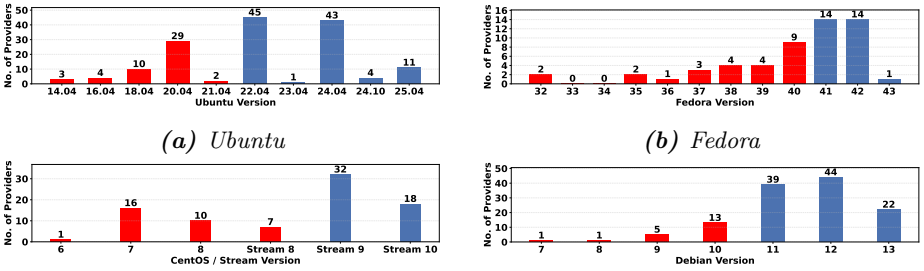
Connection Information: Our parsers processed SSH configuration files to extract crucial authentication parameters, such as, `permitrootlogin` and `passwordlogin`, which indicate whether root login and password-based authentication are enabled (both of which are considered insecure since they allow credential-based, brute-force attacks).

Firewall: We probed the presence of a default firewall both internally (from within the VM itself) as well as externally (from a remote machine port scanning the newly-created VM). The resulting open ports were then transformed into categories of firewall "strength" (i.e. the more ports that are blocked by default, the stronger the firewall).

3.4 Security Weaknesses Analysis

To compare the security weaknesses of all cloud providers, our methodology consists of two main steps.

First, we normalize all collected metrics. These metrics may take the form of binary values, ordinal values, or grouped categorical values, each of which is normalized according to its respective scale. Second, we apply the CWSS framework



(a) Ubuntu *(b) Fedora*

(c) CentOS / Stream *(d) Debian*

Fig. 2: *Version distribution across cloud providers for Ubuntu (a), Fedora (b), CentOS Stream (c), and Debian (d). Bars highlighted in red indicate end-of-support (EOS) releases.*

to assign security weightings and compute weighted scores. The weighting process proceeded as follows: the first author produced an initial weighting proposal based on CWSS guidelines; two additional graduate students with cybersecurity expertise independently reviewed the metrics and generated their own weighting assignments; finally, all three evaluators discussed discrepancies and reached consensus on the final weighting values.

The final security weakness score for each instance is computed by summing the normalized metric values multiplied by their corresponding weights, and normalizing the result once more. This produces comparable weakness scores and enables classification of providers into low-, medium-, and high-weakness categories.

3.5 Ethical Considerations

All data collection, probing, and configuration analysis activities were performed exclusively within virtual machines (VMs) that we deployed in our own cloud accounts, on resources that we controlled. At no point did we perform exploits or attack attempts against our own VMs, nor did we attempt to probe, test, or compromise the infrastructure or management interfaces of any cloud provider. Our methods were limited to non-intrusive inspection, information gathering, and configuration analysis, ensuring no disruption or risk to the cloud platforms, their other tenants, or the broader Internet.

4 Cloud Provider Analysis

4.1 OS Availability

Figure 2 summarizes the availability of four major Linux distributions (Ubuntu, Fedora, CentOS Stream, and Debian) each of which includes multiple version variants. The red bars represent End-of-support (EOS) versions as defined by the upstream vendors. EOS Linux distributions are OS versions that no longer receive security updates or maintenance from their upstream vendors.

Ubuntu is the most widely-available distribution. Versions 22.04 and 24.04 are the most commonly offered, available on all 45 providers and 43 providers, respectively. The majority of cloud providers (64%) continue to offer the EOS Ubuntu 20.04, which reached end of standard support from Canonical in May 2025. The oldest Ubuntu version observed in our dataset is 14.04, still available on DreamHost, Brightbox, and HostVDS, despite having reached end of support almost seven years ago (i.e. in April 2019).

Debian is the second most widely-available distribution. Debian 12 is offered by nearly all cloud providers (44/45) with the exception of Amazon AWS. Debian 11 is the second most common release, available on 39 of the 44 providers that support Debian. The oldest Debian release observed is Debian 7 (EOS since May 2018), which is still available on Tencent Cloud.

CentOS Stream is the third most commonly offered distribution. The major versions available are Stream 9 (32/45 providers) and Stream 10 (18/45). Stream 8 (EOS since May 2024) remains available on a smaller set of providers (7/45), including Xenyth, Alibaba Cloud, Tencent Cloud, and Huawei Cloud.

Fedora has the fewest available version variants compared to the other distributions. Only 31% of providers offer Fedora 41 or Fedora 42. Notably, among the Big Three cloud providers (AWS, Azure, and GCP), only Google Cloud Platform (GCP) offers Fedora, specifically version 43. However, nine providers continue to offer Fedora 40, which has been unsupported since May 2025. The oldest Fedora release in our dataset is Fedora 32, originally released in April 2020 and no longer supported; it remains available on Katapult and Hostwinds.

One additional feature we observed during our analysis is the presence of End-of-Support tags displayed alongside the available operating system options in the provider's web interface. Such notifications serve as an important warning, potentially steering users away from deploying outdated OS versions. However, only 9 out of 45 providers (20%) offer this feature in their web GUI. These providers are GCP, Linode, PhoenixNAP, Fuga, Alibaba Cloud, Tencent Cloud, Huawei Cloud, Zenlayer, and RamNode.

4.2 Default Configurations

Next to collecting OS availability, we also gather configuration information for the common versions of each operating system. This data is later used to compare the base security posture and security weaknesses of the same purported image across cloud providers. In this way, we aim to understand whether the choice of cloud providers matters beyond the surface-level availability of different versions of OS and distributions.

The common versions examined for each distribution are as follows: Ubuntu: 20.04, 22.04, and 24.04; Fedora: 40, 41, and 42; CentOS Stream: 9 and 10; Debian: 11, 12, and 13. In total, we collected configuration data from 307 Linux instances across all distributions. The list of operating systems and versions included in our dataset is presented in Table 3 (in the appendix). We explain each type of collected data as follows:

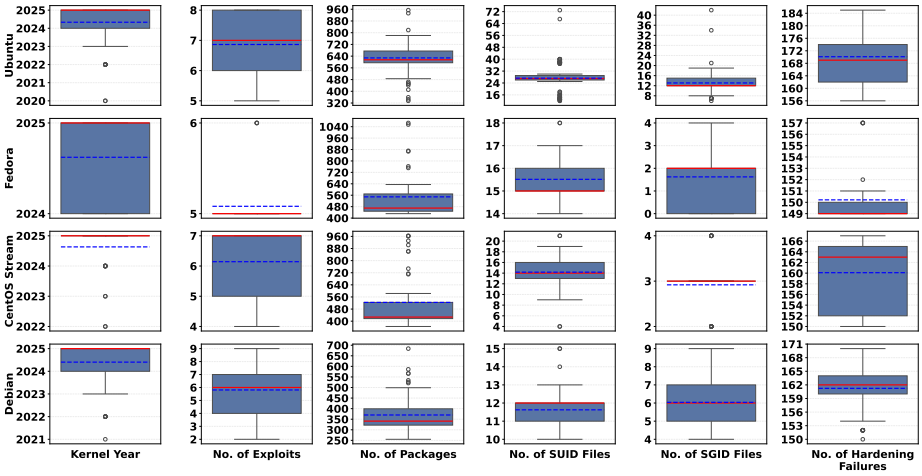


Fig. 3: *Box plot illustrating the distributions of the collected metrics. The red line represents the median, while the blue line indicates the mean.*

Kernel This metric represents the kernel versions used by each Linux distribution. The majority of kernel versions follow the format `Major.Minor.Patch-ABI`, The Application Binary Interface (ABI) number can be used to distinguish newer builds even when the kernel version itself remains unchanged. The leftmost column in Fig. 3 presents the box-plot distribution of kernel release years across all cloud providers in our dataset.

For the Ubuntu distribution, we first examine Ubuntu 20.04. All providers deploy kernel version 5.4.0; however, the ABI varies significantly, ranging from 26 to 216. ABI 26 represents the oldest kernel in our dataset (released in April 2020), while ABI 216 corresponds to the most recent (April 2025). Despite this variance, all kernels in the 5.4.0 series have already reached end of support.

For Ubuntu 22.04, most providers use kernel version 5.15.0, with 26 out of 45 (58%) offering a relatively up-to-date build. Notably, only the Big Three cloud providers (AWS, Azure, and Google Cloud Platform) upgrade their instances to kernel version 6.8.0. These providers consistently update and rebuild their kernels, resulting in regularly advancing ABI versions. At the time of measurement, the ABIs for AWS, Azure, and GCP were 1029, 1031, and 1041, respectively.

For Ubuntu 24.04, the dominant kernel version is 6.8.0, used by more than 74% of providers (32 out of 43). Similar to Ubuntu 22.04, only the Big Three offer a more recent kernel version, 6.14.0.

Fedora exhibits comparatively newer kernel versions than Ubuntu, primarily due to its twice-yearly release schedule. For example, Fedora 40 and Fedora 41 were both released in early and late 2024, respectively. Each Fedora release includes a different kernel range. For instance, Fedora 40 contains kernel versions ranging from 6.8.x to 6.14.x. The oldest kernel observed is 6.8.5-301, used by most cloud providers offering Fedora 40 (4 out of 9). In contrast, the most recent kernel version, 6.14.5-100, appears on Scaleway and Tencent Cloud.

A similar pattern is observed for Fedora 41 and Fedora 42, which include kernel versions ranging from 6.11.x to 6.16.x. Only eight cloud providers continue to offer kernel versions released in 2024 for Fedora 41.

CentOS Stream offers kernel versions ahead of Red Hat Enterprise Linux (RHEL), as it serves as a rolling preview of upcoming RHEL releases. CentOS Stream 9 was released in 2021, and CentOS Stream 10 became available in 2024. Both versions continue to receive upstream support. The primary kernel version for Stream 9 is 5.14.0 across all cloud providers, with variation occurring only in the ABI revisions. The oldest ABI identified is 105, released in June 2022 on LightNode, whereas the newest is 624, released in October 2025 on Clouding. In our dataset, all observed Stream 10 instances use kernel version 6.12.0, with all releases originating in 2025.

Debian 11 includes kernel variants released between 2021 and 2025. The majority—92% (35 out of 38)—offer kernel version 5.10.0, which is supported as an LTS kernel. Only three cloud providers (VPSserver, Hostwinds, and Kamatera) offer a newer kernel version (6.1.0) through backports, providing improved hardware support and additional security features.

A similar trend appears in Debian 12. The majority of providers deploy kernel version 6.1.0, while only PhoenixNAP offers kernel version 6.9.10. For Debian 13, all cloud providers use newer kernel versions in the 6.12.x series (6.12.38–6.12.48), released in late 2025.

Number of Exploits This metric represents the total number of publicly known kernel vulnerabilities associated with each instance. We include all exploit counts obtained from the sources referenced in [21]. The minimum number of exploits observed is two, reported for Debian 12 and Debian 13 on Vdsian, as well as Debian 12 on Katapult. The vulnerabilities that may be exploitable include `CVE-2022-2586` [19], and `CVE-2021-22555` [17]. Both vulnerabilities have a CVSS 3.x score of 7.8 (High).

The largest number of known kernel vulnerabilities observed is nine, which occurs on Debian 11 on HostVDS. Upon investigation, the variation in vulnerability counts across Debian versions correlates with differences in kernel versions. Debian 11 uses the 5.10.x kernel series, whereas Debian 12 and Debian 13 rely on kernel versions 6.1.0 and 6.12.41, respectively.

For other OS distributions, including Ubuntu, the modal number of vulnerabilities is 6, 8, and 7 for versions 20.04, 22.04, and 24.04, respectively. For Fedora, the mode is 5 across all observed versions. CentOS Stream exhibits modal values of 7 for Stream 9 and 5 for Stream 10. Across all instances in the dataset, the mean number of known kernel vulnerabilities is 6.

Number of Packages This metric represents the total number of installed packages collected from each instance. Although not all packages contain vulnerabilities, a large number of pre-installed packages generally increases the attack surface by introducing additional components that may contain exploitable flaws. Prior work has shown that reducing unused functionality can significantly reduce a system's attack surface [4]. The third column in Fig. 3 shows the distribution of package counts across providers.

For Ubuntu, the majority (50%) of instances have 600–680 pre-installed packages. The largest number of packages observed is 954 on Tencent Cloud. During our analysis, we found that one of these additional packages reports suspicious or potentially malicious files to users, which Tencent Cloud has chosen to have pre-installed. The lowest package counts were observed on HostVDS, with 359, 345, and 335 packages on Ubuntu 20.04, 22.04, and 24.04, respectively.

Other distributions generally offer fewer packages than Ubuntu, with typical ranges of 320–400 for Debian, 420–520 for CentOS Stream, and 450–580 for Fedora. An outlier is Alpha3Cloud whose Fedora distributions include more than twice as many pre-installed packages as the average Fedora distribution.

Number of Hardening Failures To evaluate the security posture of each instance, we rely on an automated tool that assesses Linux security hardening configurations [1]. This tool examines kernel configuration options and reports whether each hardening measure is enabled (OK) or disabled (Fail).

For our analysis, we consider only the hardening failures. That is, the security mechanisms that are not enabled by default. The total number of hardening failures serves as an indicator of the potential security weaknesses present in each instance. As shown in the rightmost column of Figure 3, 50% of Ubuntu images exhibit higher hardening failure counts compared to other distributions, with values typically ranging between 162 and 175 failures.

Number of SUID Files Privilege escalation is another attack vector in which a malicious actor attempts to obtain higher privileges. Set User ID (SUID) programs are binaries that execute with the effective user ID of the file owner—often `root`. For this metric, we enumerate all programs configured with the SUID bit set to `root`. We do not evaluate whether these SUID programs are practically exploitable; instead, we focus solely on their presence as a potential risk factor.

As shown in Figure 3 (fourth column), Ubuntu exhibits a relatively narrow but higher distribution of SUID binaries, typically ranging between 25 and 30. Upon investigation, we found that many of these SUID entries originate from Snap packages installed within the Snap environment (e.g., `/snap/...`). On Ubuntu 20.04 provided by Alpha3Cloud, we observed more than 70 SUID-configured programs. In contrast, approximately 50% of Debian, Fedora, and CentOS Stream instances offer significantly fewer SUID binaries, typically around 11 to 16, compared to Ubuntu.

Number of SGID Files Similar to SUID, Set Group ID (SGID) programs execute with the effective group ID of the file owner. Attackers may leverage SGID binaries to escalate privileges within the group privilege domain. For this metric, we consider SGID binaries that are configured with `root` group owner.

As with SUID files, Ubuntu instances exhibit a higher total number of SGID binaries compared to other distributions. As with SUID, a primary reason for this increase is the presence of Snap packages, which assign the root group to several of their installed files. In contrast, Fedora and CentOS Stream show significantly lower SGID counts, typically between 2 and 4.

Table 1: *Default Security Configuration Across Distributions*

Metric	Sec	Mod	Ins	Metric	Sec	Mod	Ins
Ubuntu				CentOS Stream			
Firewall	33.3%	1.7%	65.0%	Firewall	55.1%	4.1%	40.8%
Root Login	32.5%	N/A	67.5%	Root Login	14.3%	N/A	85.7%
Password Login	43.6%	N/A	56.4%	Password Login	42.9%	N/A	57.1%
Fedora				Debian			
Firewall	51.4%	N/A	48.6%	Firewall	32.7%	1.9%	65.4%
Root Login	27.0%	N/A	73.0%	Root Login	28.8%	N/A	71.2%
Password Login	59.5%	N/A	40.5%	Password Login	43.3%	N/A	56.7%

Note: Sec(Secure) = root/password login disabled or strong firewall; Ins(Insecure) = root/password login enabled or weak firewall; Mod(Moderate) = mid-range firewall responsiveness.

Firewall Our firewall feature aims to capture the presence of a default firewall on a new VM (running either in the VM itself or on the underlying cloud platform) and the extent of its restrictiveness. To evaluate firewall availability across cloud providers and Linux distributions, we adopt a detection technique in which the top 50 most common ports are opened on each instance, by running TCP listening software on each port. We then automatically perform an external port scan from a separate server, to understand how each VM appears from the point of view of an attacker.

Rather than treating the firewall as a binary feature (enabled or disabled), we classify the observed behavior into three levels: *Strong* (instances exposing 0–5 reachable ports), *Medium* (instances that do not fall into the strong or weak ranges, typically around 11 ports in our dataset), and *Weak* (instances exposing a large number of reachable ports, generally between 40 and 50). This classification captures the practical exposure of each instance and accounts for discrepancies and measurement noise introduced by network conditions between our vantage point and cloud providers [25].

We observed that several providers, such as AWS, Azure, Google Cloud Platform, and Exoscale, enable a provider-level firewall by default, which blocks all inbound traffic unless explicit rules are added. Other providers, such as Linode, offer firewall functionality but do not enable it as part of the default configuration. For providers in the former group, we added a minimal inbound rule permitting SSH (port 22) solely to ensure access to the instances in case manual inspection was required during data collection.

As shown in Table 1, Ubuntu and Debian have a large proportion of instances that expose numerous insecure ports to the public (weak firewall), with 65.0% of Ubuntu instances and 65.4% of Debian instances classified as weak. The number of strong-firewall instances is noticeably lower—33.3% for Ubuntu and 32.7% for Debian—representing fewer than half the number of weak-firewall cases.

In contrast, Fedora and CentOS Stream exhibit a different trend, showing a slightly higher number of strong-firewall instances compared to weak ones.

We also observed that even when a cloud provider does not enforce a provider-level firewall, the operating system's internal firewall may still be enabled by default, thereby reducing the number of externally reachable ports. The variation in open-port counts follows naturally from these configurations: strong-firewall instances typically expose only essential or mandatory ports such as 22 (SSH), 80 (HTTP), and 443 (HTTPS), whereas weak-firewall instances tend to expose most or all of the ports that were opened during our measurement process.

From our analysis, we identified only one cloud provider, Clouding, that consistently applies a default firewall configuration across all tested distributions (CentOS Stream 9 and 10, Debian 12 and 13, and Ubuntu 22.04 and 24.04). Clouding allows eleven open ports by default, all of which correspond to well-known services such as 21 (FTP), 22 (SSH), 25 (SMTP), 80 (HTTP), 443 (HTTPS), and 3389 (RDP).

Root Login Root login refers to the ability for a user to authenticate directly to an instance using the `root` account, a practice that is generally considered risky [2]. If permitted, attackers can attempt to brute-force password attacks against the root account [22]. To determine whether root login is allowed or blocked, we examined both the SSH configuration and the forced-command settings in the root user's shell configuration. Forced commands are commonly used by cloud providers to automatically redirect login attempts from `root` to a non-privileged user.

Across all distributions, the majority of instances (71.1%) allow direct root login, while only about a quarter explicitly disable it. As previously noted, some providers implement forced-command redirection when a user attempts to connect as `root`, mapping the session to a designated non-root user. Examples include `ec2-user` on AWS, `azureuser` on Azure, and `cloudsigma` on Alpha3Cloud.

Password Login Password-based SSH authentication is widely considered a less-than-ideal security practice [7], as attackers may exploit weak or reused passwords through brute-force attempts. In our analysis, the majority of cloud providers (62.2%) using Ubuntu, CentOS Stream, and Debian enable SSH password login by default. We also observed that most cloud providers allow users to choose between public-key authentication and password-based login during instance setup. Only a few providers (e.g.AWS) mandate the use of public-key authentication.

5 Cloud Provider Weakness Scoring

5.1 Scoring Weaknesses

Metrics Weighting Using the calculation process described in Section 3.4, Table 2 presents the final weighting values used in our analysis. The three highest-weighted metrics are Root Login (78.0), End-of-Support (EOS) OS (76.2), and Password Login (55.6), respectively. Root login and password-based authentication are considered low-effort attack vectors that can immediately increase the

Table 2: *Weighting assigned to security weakness metrics (higher is worse).*

Security Weakness Metric	Weight	Security Weakness Metric	Weight
Root Login	78.0	Outdated Kernel	47.5
End-of-Support OS	76.2	No. of SUID	45.4
Password Login	55.6	No. of SGID	33.2
No. of Exploits	48.8	No. of Packages	21.5
No. Hardening Failures	48.5	Firewall	9.6

exposure of an instance. Similarly, EOS operating systems are no longer maintained by upstream vendors and are likely to contain known kernel and package vulnerabilities, thereby significantly affecting a base-image's security.

Scoring Process To calculate the final security weakness score, we first normalize all collected data, then aggregate the normalized values, and finally convert the results into a 0–100 scale. Lower scores represent a stronger security posture (fewer weaknesses), whereas higher scores indicate weaker security (more weaknesses).

We apply different normalization strategies depending on the type of metric. For continuous numerical metrics, we use min–max normalization. These metrics include: the number of EOS OS versions offered by a provider, the number of exploits, the number of installed packages, kernel hardening failures, the number of SUID files, and the number of SGID files.

For binary configuration metrics, we normalize by assigning values of 0 or 1, where 0 represents a secure state (disabled) and 1 represents an insecure state (enabled). This applies to Root Login and Password Login. For ordinal or grouped metrics, we map values into the range 0–1. For example, kernel release years are mapped as: years $< 2023 \rightarrow 0$, year $2024 \rightarrow 0.5$, and year $2025 \rightarrow 1$. Firewall strength is mapped as: strong $= 0$, medium $= 0.5$, and weak $= 1$.

After normalization, each metric is multiplied by its corresponding weight and summed to obtain the overall weakness score. We then classify the final scores into three levels of security weakness: [0, 33] represents low weakness, [34, 66] represents moderate weakness, and [67, 100] represents high security weakness.

5.2 Scoring Results

To compare security weaknesses across all datasets, rather than evaluating each OS version individually for every cloud provider, we aggregate the weakness scores of all versions of a given distribution offered by a provider and compute their average. This average reflects the provider's overall security weakness level, for that specific distribution (e.g. Ubuntu vs. Debian). Figure 4 (see the appendix, leftmost column) shows the distribution of Ubuntu scores across cloud providers. The majority fall into the medium-weakness category, with 25 out of 45 providers (55%). Both the low- and high-weakness categories contain 10 providers each.

To illustrate the distinction between low and high weakness scores, consider the following example. Azure is classified as having the lowest security weakness (i.e. highest security posture), offering Ubuntu 22.04 and 24.04 with an average weakness score of 0.85. In contrast, Hostwinds—which offers all three Ubuntu versions (20.04, 22.04, and 24.04)—has an average score of 93.89. Several configuration factors contribute to this disparity. Azure deploys kernels released in 2025, disables root login, and provides strong default firewall protections. Hostwinds, on the other hand, uses 2024 kernels across all variants, allows root login, and lacks default firewall rules.

CentOS Stream and Debian exhibit patterns similar to Ubuntu, with most providers falling within the medium-weakness range. Fedora, however, displays a more polarized distribution: an even split between low- and high-weakness classifications (7 providers in the low category and 8 in the high). This divergence is primarily influenced by differences in whether root login is disabled and whether a firewall is enabled by default. Low-weakness Fedora providers tend to enforce stronger authentication and firewall policies, whereas high-weakness providers generally do not.

Overall, we observe few instances of providers changing score bands between different OS distributions (Vultr is such an instance where their Ubuntu and Debian distributions score significantly better than their Fedora and CentOS ones). At the same time, there is measurable in-band variance where a given provider is never "always the best" or "always the worst."

5.3 Security Weakness Weighting Sensitivity Analysis

To assess potential bias in the security weakness weighting scheme, we examine whether cloud provider rankings change under different metric weights. We measure ranking similarity using Kendall's τ [11], where -1 indicates perfect disagreement and 1 indicates perfect agreement.

We evaluate three scenarios: (i) $\pm 20\%$ weight perturbation, (ii) equal weighting, and (iii) random weighting. For the perturbation and random-weighting scenarios, we report the mean Kendall's τ over 5,000 iterations. Results show that provider rankings remain highly consistent across all scenarios. Across all OS distributions, τ exceeds 0.94 under $\pm 20\%$ perturbation, remains above 0.82 under equal weighting, and stays above 0.70 even under random weighting. In all cases, the correlations are statistically significant ($p < 10^{-5}$).

These results indicate that our derived cloud-provider rankings are robust to reasonable variations in metric weighting.

5.4 Web ranking vs. Security Stance

In addition to evaluating security weaknesses, we examine the reputation of each cloud provider using the Tranco List [13] to determine whether higher-ranked providers exhibit fewer security weaknesses. Across all distributions, however, we observe no clear correlation between provider reputation and security weakness scores. For example, within the Debian dataset, Fuga (which ranks 662,407 on the Tranco list) has a low security weakness score of 8.2, whereas Hostinger,

ranked 1,216, has a much higher weakness score of 69.0. Similar patterns appear across other distributions, where provider reputations are scattered across all weakness levels with no significant relationship.

6 Discussion and Limitations

We observed that some default configurations vary depending on the connection method used during initialization, particularly for SSH settings. For example, Zenlayer offers two authentication options: public-key and password-based login. The resulting SSH configuration differs between these methods: when connecting with a public key, password authentication is disabled, whereas it remains enabled when using the password-based setup. This discrepancy can influence our scoring. In our experiments, when multiple connection methods were available, we did not enforce a strict selection criterion and used one of the available options, which may introduce slight variations in scoring. Notably, such inconsistencies were not observed across most other providers. In several cases, even when authenticating with a public key, the default SSH configuration still enabled password-based authentication regardless of the connection method used.

Considering the totality of our experiments, we observe the following key takeaways:

- Most cloud providers (66.7%) keep offering Linux distributions past their official end of support, potentially exposing users to security vulnerabilities.
- The majority of cloud providers (82.2%) use kernels maintained by upstream sources; only a few (17.8%) deploy newer kernels for older distribution versions.
- Most cloud providers (82.2%) do not by default block all non-SSH ports.
- Root login (71.1%) and password-based authentication (62.2%) remain enabled on many cloud platforms, even though both are considered insecure configurations.
- Most providers occupy a specific scoring band for all of their distributions; no provider is *always the best* or *always the worst*.
- There is no correlation between web reputation and security weaknesses; in some cases, lesser-known providers demonstrate a stronger security posture.

Limitations. All security weakness evaluations presented in this work are based solely on our defined metrics and methodology. Therefore, a low weakness score does not imply that a cloud provider is fully secure from a holistic security perspective, nor does a high weakness score indicate poor security practices overall. Instead, the scores reflect comparative differences within the constraints of our dataset and do not account for factors outside our measurement scope.

Our analysis is snapshot-based. That is, our collected scores and statistics were valid at the time of their collection (October 2025). It is entirely possible that these scores evolve over time, changing the weakness scores of individual cloud providers and how they rank compared to others. We automated as much

as possible of provisioning and probing of different VMs so that these analyses can be repeated in future work, in the scope of a longitudinal study. Future work can further expand upon our work by incorporating additional security metrics and by evaluating multiple instance types, regions, and Windows-based operating systems.

7 Related Work

Cloud Security A major area of cloud security research has focused on side-channel attacks [26–29], where attackers exploit resource sharing on multi-tenant cloud infrastructures. These attacks demonstrate that co-resident virtual machines can leak sensitive information such as private keys. To mitigate such threats, cloud providers introduced the concept of Virtual Private Clouds (VPCs) to isolate tenants within a provider's infrastructure, although challenges related to co-residency remain. For example, Varadarajan et al. [24] proposed techniques for detecting co-residency by actively probing cloud placement behavior.

More recently, Kondracki et al. [12] investigated the security implications of deploying TLS-enabled web servers in the cloud. Their work showed that, once a TLS certificate is published in Certificate Transparency (CT) logs, automated scanners and malicious actors quickly discover the corresponding servers, which exhibit weakened security postures at the time they first appeared in CT logs. Their findings highlight how default or improperly-configured cloud instances can unintentionally broaden the attack surface immediately upon deployment. While not directly related to cloud security, Van Goethem et al. conducted a large scale study of the adoption of web-security mechanisms across popular sites of the web [23]. Among others, the authors devised a security scoring system to be able to compare sites against each other, which inspired the scoring system used in this paper.

These prior works emphasize the risks introduced by information leakage, co-residency, and unintended exposure of newly deployed cloud services. However, most existing research focuses on specific attack vectors or post-deployment behaviors. In contrast, our work investigates the security posture of cloud instances *at the moment of provisioning* by systematically measuring the default configurations supplied by cloud providers.

Cloud Comparison The pay-as-you-go model allows users to optimize cloud usage based on pricing, performance, and feature requirements. Prior work in cloud comparison has primarily focused on performance and cost evaluation rather than security analysis. For example, Li et al. [14] analyzed and compared the performance and pricing of four major cloud providers: Amazon AWS, Microsoft Azure, Google AppEngine, and Rackspace. Nadeem et al. [16] proposed a framework for evaluating and ranking cloud services using functional and non-functional Key Performance Indicators (KPIs). Similarly, Mok et al. [15] conducted a latency analysis focused solely on Google Cloud Platform.

While prior work has compared cloud providers based on price, performance, and limited operational characteristics, no study to date has systematically ex-

amined default security weaknesses across a broad set of cloud providers. Our work addresses this gap by evaluating the security posture of cloud environments from the perspective of default configurations.

8 Conclusion

In this paper, we examined the security of public cloud platforms, focusing on the default configurations of popular VM images. Our findings show that the most commonly offered Linux distributions across providers are modern versions of Ubuntu and Debian. However, we also identified cases where outdated or unsupported distributions, such as Ubuntu 14.04, are still offered with (66.7%) of cloud providers still offering distributions past their end-of-service. We conducted a security weakness assessment using ten collected metrics and classified providers into low, moderate, and high weakness categories. Our analysis shows that the most influential factors affecting the security weakness score are enabled root logins, the availability of end-of-support (EOS) operating systems, and the presence of password-based SSH authentication. Our results *categorically* show that where an image is deployed matters, from a security perspective. We hope that our findings provide practical insights for users when selecting cloud providers and encourage providers to ensure that the VM images they offer are as secure as possible.

Acknowledgment

We thank our shepherd and the reviewers for their valuable feedback. This work was supported by the Office of Naval Research (ONR) under grant N00014-24-1-2193, as well as by the National Science Foundation (NSF) under grants CNS-1941617 and CNS-2211575.

References

1. a13xp0p0v: Kernel hardening checker (2025), `https://github.com/a13xp0p0v/kernel-hardening-checker`
2. Academy, S.: Root user account and how to root phones (2025), `https://www.ssh.com/academy/pam/root-user-account`
3. Alexandre: Cloud market share 2026: Top cloud providers and trends (2025), `https://holori.com/cloud-market-share-2026-top-cloud-vendors-in-2026`
4. Azad, B.A., Laperdrix, P., Nikiforakis, N.: Less is more: Quantifying the security benefits of debloating web applications. In: USENIX Security Symposium (2019)
5. Canonical: Cloud-init (2025), `https://cloud-init.io/`
6. Corporation, T.M.: Common weakness scoring system (cwss™) (2014), `https://cwe.mitre.org/cwss/cwss_v1.0.1.html`
7. Garn, D.: Eight ways to protect ssh access on your system (2020), `https://www.redhat.com/en/blog/eight-ways-secure-ssh`
8. Hashicorp: Automate infrastructure on any cloud (2025), `https://developer.hashicorp.com/terraform`

9. Hat, R.: Ansible (2025), https://www.redhat.com/en/ansible-collaborative
10. of Incident Response, F., Teams, S.: Common vulnerability scoring system sig (2023), https://www.first.org/cvss/
11. Kendall, M.G.: A new measure of rank correlation. Biometrika **30**(1-2), 81–93
12. Kondracki, B., Ferdman, M., Nikiforakis, N.: Ready or not, here i come: Characterizing the security of prematurely-public web applications. In: Annual Computer Security Applications Conference (ACSAC). pp. 175–189. IEEE (2024)
13. Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., Joosen, W.: Tranco: A research-oriented top sites ranking hardened against manipulation. In: Network and Distributed Systems Security (NDSS) Symposium (2019)
14. Li, A., Yang, X., Kandula, S., Zhang, M.: Cloudcmp: comparing public cloud providers. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. pp. 1–14. IMC '10 (2010)
15. Mok, R.K., Zou, H., Yang, R., Koch, T., Katz-Bassett, E., Claffy, K.C.: Measuring the network performance of google cloud platform. In: Proceedings of the 21st ACM internet measurement conference. pp. 54–61. IMC '21 (2021)
16. Nadeem, F.: Evaluating and ranking cloud iaas, paas and saas models based on functional and non-functional key performance indicators. IEEE Access **10**, 63245–63257 (2022)
17. NIST: CVE-2021-22555 (2021), https://nvd.nist.gov/vuln/detail/cve-2021-22555
18. NIST: Nsa and cisa red and blue teams share top ten cybersecurity misconfigurations (2023), https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a
19. NIST: CVE-2022-2586 (2024), https://nvd.nist.gov/vuln/detail/cve-2022-2586
20. NIST: National vulnerability database (2024), https://nvd.nist.gov/
21. PEASS-ng: Privilege escalation awesome scripts suite new generation (2025), https://github.com/peass-ng/PEASS-ng
22. Singh, S.K., Gautam, S., Cartier, C., Patil, S., Ricci, R.: Where the wild things are: brute-force ssh attacks in the wild and how to stop them. In: 21st USENIX Symposium on Networked Systems Design and Implementation. pp. 1731–1750. NSDI '24 (2024)
23. Van Goethem, T., Chen, P., Nikiforakis, N., Desmet, L., Joosen, W.: Large-scale security analysis of the web: Challenges and findings. In: International Conference on Trust and Trustworthy Computing. pp. 110–126. Springer (2014)
24. Varadarajan, V., Zhang, Y., Ristenpart, T., Swift, M.: A placement vulnerability study in multi-tenant public clouds. In: Proceedings of the 24th USENIX Conference on Security Symposium. pp. 913–928. SEC '15 (2015)
25. Wan, G., Izhikevich, L., Adrian, D., Yoshioka, K., Holz, R., Rossow, C., Durumeric, Z.: On the origin of scanning: The impact of location on internet-wide scans. In: Proceedings of the ACM Internet Measurement Conference (2020)
26. Yarom, Y., Falkner, K.: FLUSH+RELOAD: A high resolution, low noise, l3 cache Side-Channel attack. In: 23rd USENIX Security Symposium. pp. 719–732. USENIX Security 14 (2014)
27. Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-vm side channels and their use to extract private keys. In: Proceedings of the 2012 ACM conference on Computer and Communications Security. pp. 305–316. CCS '12 (2012)
28. Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-tenant side-channel attacks in paas clouds. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 990–1003. CCS '14 (2014)

29. Zhao, Z.N., Morrison, A., Fletcher, C.W., Torrellas, J.: Last-level cache side-channel attacks are feasible in the modern public cloud. In: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. pp. 582–600. ASPLOS '24 (2024)

A Evaluated Cloud Providers

Fig. 4: *Security weakness score distributions across cloud providers, grouped into three categories: low weakness (green), medium weakness (yellow), and high weakness (red).*

Table 3: *Operating systems used in our data collection across major cloud providers. Columns shaded in red indicate end-of-support (EOS) distributions.*

Cloud providers	**Ubuntu**			**Fedora**			**CentOS Stream**		**Debian**		
	20.04	22.04	24.04	40	41	42	9	10	11	12	13
Amazon AWS	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓
Amazon Lightsail	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
Microsoft Azure	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗
Google Cloud Platform (GCP)	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓
Linode	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
DigitalOcean	✗	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓
Kamatera	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓
PhoenixNAP	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
Vultr	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
OVHcloud US	✓	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓
Cloudscale	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓
Latitude	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
AraCloud	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
Serverspace	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
Gcore	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗
Cloudcone	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
DreamHost	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
Hetzner	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
Scaleway	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
UpCloud	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓
Exoscale	✗	✓	✓	✗	✗	✗	✓	✓	✗	✓	✓
Aruba Cloud	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓
Katapult	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
Brightbox	✓	✓	✓	✗	✓	✓	✗	✗	✓*	✓	✓
Cherry Servers	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
VDSina	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓
Zomro	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
Fuga	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
VPSserver	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓
Clouding	✗	✓	✓	✗	✗	✗	✓	✓	✗	✓	✓
Alibaba Cloud	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗
Tencent Cloud	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✗
Huawei Cloud	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
Zenlayer	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✗
Hostwinds	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
Hostinger	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
LightNode	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
Cloudzy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Gandi	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓
HostVDS	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗
RamNode	✓	✓	✓	✗	✓	✗	✓	✓*	✗	✓	✗
Hivelocity	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
Hostkey	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
Alpha3Cloud	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗
Xenyth	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗
Total	29	45	43	9	14	14	32	18	39	44	22

Note: Data were collected on October 6, 2025.
* Indicates distributions where issues were encountered during dynamic analysis.