Longitudinal Study of Internet-Facing OpenSSH Update Patterns

Jonathan Codi West and Tyler Moore

School of Cyber Studies, The University of Tulsa, Tulsa OK, USA
{codiwest,tyler-moore}@utulsa.edu

Abstract. Keeping server software patched and up-to-date is a never-ending struggle for system administrators that is crucial for security. Nevertheless, we know little about how well or how consistently software updates are applied over time across the Internet. We shed light on software update behavior on publicly addressable networks by utilizing Internet-wide scans of OpenSSH banners. We primarily focus on OpenSSH banners which contain patch-level information in order to map accurate release dates. We augment this view by tracking which software security backports fix vulnerabilities in older OpenSSH versions. We find that the availability of backports, not CVE announcements or upstream software updates, trigger rapid updates. Unfortunately, we also determine that the lag in publishing backports (if they are published at all) combined with the steady cadence of new vulnerability reports ensures that most of the time, the vast majority of machines are vulnerable to at least one CVE. Additionally, we observe that major cloud hosting providers are consistently faster to apply patches.

1 Introduction

One of the pillars of cybersecurity hygiene is updating software regularly. When vulnerabilities are identified, developers issue security patches to seal the hole. While much progress has been made in improving the update process for end users, patching server-side systems can still be difficult. Unpatched systems in turn offer an opportunity for attackers to exploit vulnerabilities that lead to compromise.

Despite the importance of patching, we do not know very much about the patching practices at Internet scale. In this paper, we measure software outdatedness on the publicly-facing IPv4 address space. We demonstrate that naïve approaches to measure outdatedness through publicly-announced version information paints too negative a picture of patching in the enterprise.

We hone in on "backports", patches applied by operating system distributors to fix older software versions, to get a more accurate picture. One positive conclusion is that backports are rapidly applied by many, and therefore software is often more up-to-date than what can be inferred by looking at the version information alone. Nonetheless, we uncover significant concerns. For the case of OpenSSH backports on Ubuntu, we determined that around 25% of CVEs had

no backport issued. During most of the 2015–2019 period under investigation, all servers were vulnerable to at least one CVE, often more. Moreover, at any given time between 40-80% of machines that could apply a backport to fix a CVE have not yet done so.

We review our approach to data collection in Section 2. We then set out to iteratively refine our definition of software outdatedness and patching levels in Section 3. In Section 4 we explicitly connect software outdatedness to software vulnerabilities for the case of OpenSSH software running on Ubuntu. We discuss limitations in Section 5, followed by Related Work before concluding in Section 7.

2 Data Collection Methodology

We utilize Censys [2] to acquire open ports and service banner data across the entire IPv4 address space. Censys keeps historical data, and thus we download several snapshots between 2017 and 2020, which contain banners for the services listed in Table 1. After narrowing the scope to just OpenSSH banners, we downloaded weekly snapshots of the entire IPv4 address space that have SSH banners. These weekly snapshots range from October 2015, which was the earliest we found SSH banners on Censys, through December 2020, although more recent data is planned for future work.

We gather software version release dates of several popular Internet-facing software packages from Github [5] and their respective websites and changelogs where available. As will be explained in Section 3.2, security patch release and superseded dates for the OpenSSH software package running on the Ubuntu or Debian Linux distributions are gathered from Launchpad [6]. We acquired OpenSSH patch data dating back to OpenSSH 1.3.8 on Ubuntu Warty (4.10) in 2005.

We gather announced IPv4 address space for several cloud service providers, namely Amazon AWS, Azure Cloud, and Google Cloud. The announced address spaces for these providers are mapped to the IPv4 addresses gathered from Censys. Additionally, we use MaxMind's GeoIP2 dataset [4] and Bureau van Dijk's Orbis [9] resource to identify company ownership for IPv4 CIDR blocks.

2.1 Ethical Considerations

We did not perform active or passive scanning of Internet hosts in our data collection. We chose to use pre-existing data and not perform unnecessary scans.

3 Measuring Software Outdatedness

We iteratively build a more sophisticated and accurate measurement of software outdatedness and apply it to the data gathered.

Table 1: Software versions inferred from banner, with example banner text.

Port	Software	Example Banner
80, 443	Apache2	`Apache/2.4.16 (Unix) OpenSSL/1.0.1e-fips`
80, 443	NGINX	`nginx/1.10.3 (Ubuntu)`
21	Bftpd	`bftpd 2.2`
21	FileZilla Server	`FileZilla 0.9.47`
21	Proftpd	`ProFTPD 1.3.4a`
21	Vsftpd	`vsftpd 3.0.2`
22	OpenSSH	`SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u4`

3.1 First cut: base software version

We initially explored banner data for over a dozen ports from Censys based on data availability of those ports over time and their perceived popularity. Some of these ports run software that present version information in the banner. This version information may be parsed out and then mapped to the release dates gathered from GitHub and their respective websites. Table 1 displays which software had release dates gathered.

The software release dates are mapped to each IP address with version info. To give an initial impression of the age of Internet software, we subtract a given software version's publish date from the Censys snapshot date to compute the *days since release* for that software version. But this does not really measure outdatedness, since software only becomes out of date once a newer version is released. Hence, to track software freshness, we calculate the difference between the snapshot date and when a given software version was *superseded* by a newer version. The days superseded metric more accurately conveys how long the server owner waited to upgrade and is therefore responsible for running outdated software. If a software version is at the latest version at the date of the snapshot, then days superseded is set to 0.

For the services in Table 1, we compare the distribution of days superseded among ports. Figure 1 shows CDFs for each port using the 2020-08-08 snapshot. We see that the software on these ports tends to be rather old. It is reasonable that port 80 and port 443 have very similar curves given that they run the same software, although port 443 is running on roughly half of the number of IPs that port 80 runs on. Port 443 also has a slight edge in running more recently released software versions. OpenSSH on port 22 lags behind port 80 and 443 for the first three years and then follows a similar curve. The biggest difference is that Port 21 (FTP) is running much older software than the other three ports.

3.2 Second cut: integrating security patches

Using the superseded date of the base software version, gives a rather incomplete view of the age of Internet software. Basing software patch levels entirely on the

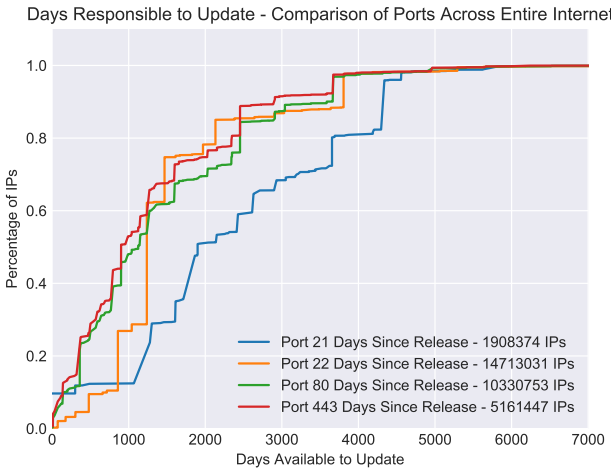


Fig. 1: Comparison of days superseded for the various services in Table 1.

software version information alone may be misleading as it ignores common security practice. Some operating systems will "backport" security patches into older versions of a given software without changing the base version number (referred to hereon as the *upstream* version number or upstream patch). In these cases, software may appear to be quite old when looking at the superseded date of the upstream version number even though the security patches are more recent and may fix vulnerabilities which were present in that upstream version.

It is often the case that security patch level information is hidden to all but those with access to the system, which is unfortunate in the case of external measurement via the Internet. Fortunately, we have identified one case where we can reliably observe the presence of backports. In OpenSSH, the security patch version is shown in some banners depending on the configuration of the host operating system, including the popular Ubuntu and Debian Linux distributions. See the example of the following Ubuntu OpenSSH banner string:

```
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
```

In this banner, the base version of OpenSSH is 7.6p1, which was released on 2017-10-03 according to [5]. Comparatively, the patch level is 7.6p1-4ubuntu0.3, which was released on 2019-03-04 according to Launchpad [7]. Coincidentally, we can infer the operating system version of this machine because this security patch version is only found on Ubuntu Bionic (18.04). All banners on Launchpad refer to a specific backport. While usually unique, the same banner is occasionally used for an LTS release and concurrent development release of Ubuntu. We considered patches in the release, updates, and security channels/pockets on Launchpad. Out of 286 entries examined from 2005 to 2020, 30 were duplicated across two releases. In those cases, the only discernible difference is the date when the patch

was published on its respective release. Consequently, we use the patch release date of the LTS version of Ubuntu if it differs. While spoofing these banners is possible, we expect it to be rare since doing so requires editing and compiling the OpenSSH source code. Any edited banners that do not exactly match an Ubuntu backport banner are excluded from that portion of the analysis.

Clearly, these patches can be mapped to a much later release date than initially inferred from looking at the upstream OpenSSH version alone. Figure 3a compares the "days superseded" of the Ubuntu security patch level (green line) to the upstream OpenSSH version level (orange line). Now the picture is not only more accurate, it is also a much better outlook from a security perspective. Around 80% of the Ubuntu OpenSSH servers immediately apply patches. If one simply judged software freshness based on the OpenSSH version, 80% of servers would be considered more than three years outdated.

From this analysis, we conclude that the picture of server software updates is not as bad as it is often portrayed. We are not out of the woods, though, because 20% of OpenSSH servers are slow to patch. That is a non-trivial number of servers. Moreover, more work needs to be done to connect the application of OpenSSH patches to the presence of software vulnerabilities, which we undertake in the next section.

Focusing on OpenSSH banners with security patch information does limit the number of IP addresses that can be used for measurement. In the case of Figure 3a, which is based on the 2020-08-08 snapshot of Censys, we are using roughly one-third of the total OpenSSH IPs for the Ubuntu security patch measurement. For reference, this same snapshot has a total of 132 million IP addresses, and about 17.6 million of those have port 22 open, 14.7 million of which run OpenSSH. Of these 14.7 million, 4.8 million run Ubuntu, and therefore have accurate backport information. Figure 2 demonstrates the coverage of OpenSSH that the Ubuntu distribution provides. Debian and Raspbian also provide patch information in the banner and could be analyzed in future work. While RedHat Enterprise Linux and its relatives make extensive use of backports [10], they regrettably do not provide patch information in the banner and thus fall into the unknown category with other distributions. For the remainder of this work, we narrow our focus where OpenSSH backported security patch information is visible, namely servers running Ubuntu Linux.

3.3 Do cloud-hosted servers update faster?

We next consider whether large cloud providers exhibit different updating behavior compared to the rest of the Internet. We consider three major hosting providers, Amazon Web Services (specifically EC2), Azure Cloud, and Google Cloud. We map their IP addresses based on the publicly announced netblocks each share.

Figure 3b uses OpenSSH patch superseded dates on Ubuntu to compare cloud providers and the remainder of the Internet. Overall, we see that all three cloud providers have relatively similar patch level distributions and that the remainder of the Internet lags behind. These cloud providers collectively account for around

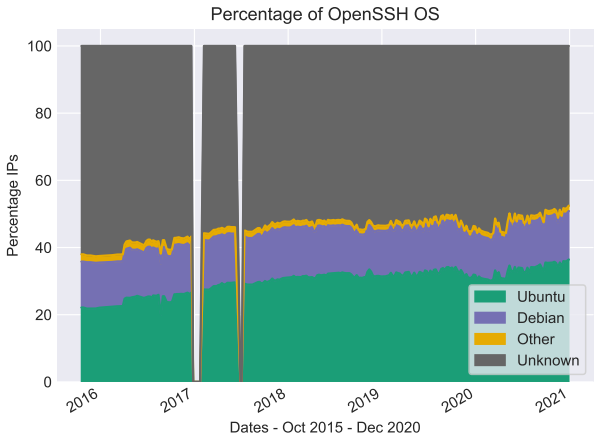


Fig. 2: Coverage of OpenSSH by operating system over time.

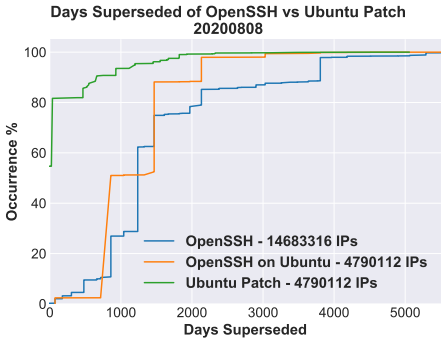
one third of all hosted Ubuntu OpenSSH servers, so their improved performance does make a difference overall.

4 How vulnerable is outdated software?

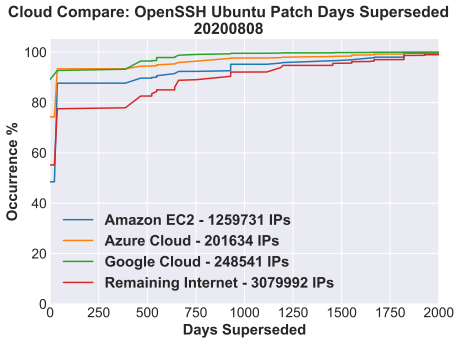
Now that the software patch level of OpenSSH can be more accurately deduced, we look at the distribution of CVEs for OpenSSH over time to test if CVEs influence patch speed. We have collected additional snapshots from Censys which range from late 2015 through 2020 on a nearly weekly basis and contain SSH banners for the entire IPv4 address space collected by Censys. We parse every SSH banner and map to a backported patch version where applicable.

We examined CVEs announced between August 2015 and the end of 2019. For each of these 27 CVEs, we create a mapping of which backported patches are affected by which CVEs. For this mapping, we start by checking which upstream versions are affected by each CVE from the National Vulnerability Database [8]. If the upstream OpenSSH version of a given security patch is not affected by a CVE, then that patch version is not considered to be affected either. For security patches where the upstream version is affected by a given CVE, we inspect the changelog text for that patch, available on Launchpad [7], to see whether either that patch or a previous patch in its tree claim to fix that CVE.

Table 2 reports the CVEs with initial publish dates, the earliest upstream patch date, and earliest Ubuntu backport patch date (if one is available). We also compute the lag between when a given CVE was announced and when it was patched in the upstream or backported fix. In the cases where the lag for the OpenSSH fix is negative, the CVE affected older versions but not the most



(a) Comparison of days superseded for OpenSSH servers overall, on Ubuntu (upstream version) and Ubuntu (backport version).



(b) Days superseded for OpenSSH Ubuntu servers running on major cloud service providers using backport information.

Fig. 3

recent one released at that time. In the few cases where the Ubuntu backport lag is negative, the CVE appeared as fixed in the changelog before the official publish date on the NVD website.

We note that 7 of the 27 CVEs are not fixed in any backports, so the only way to eliminate these vulnerabilities is to manually update the OpenSSH software to a newer upstream version. One CVE (2016-8858) was disputed in the community whether it was even a vulnerability. No backport was issued.

We utilize the mapping of CVEs to vulnerable Ubuntu backports in order to compute the number OpenSSH servers that are vulnerable and not vulnerable to each CVE for each Censys snapshot within 2015 and 2019. We then combine these over time to show the total number of IPs which are and are not vulnerable to a given CVE over time. An example of this for CVE-2016-10009 is shown in Figure 4. For this figure, the blue line represents the number of IP addresses which are vulnerable to CVE-2016-10009 over time, while the orange line is the number of IPs which are not vulnerable. The sum of the orange and blue lines at a given point on the x-axis is equal to the number of servers on that Censys snapshot which are running OpenSSH on Ubuntu with a security patch available on Launchpad. The noise in the table is related to the number of IP addresses Censys scanned at each point in time. The number of Ubuntu servers that Censys scans generally increases over time, aside from the sharp drop in early 2016 which might be attributed to large providers opting out out of Censys scanning [3]. The gaps in the lines show where no scan data was available from Censys. Several snapshots between 2016 and 2018 contain significantly fewer IP addresses than adjacent snapshots, but we chose not to omit these scans from the plots because they accurately reflect variations in how much scanning Censys completed at those points in time. These dips should be ignored when drawing conclusions.

Table 2: OpenSSH CVEs released during period of study

	CVE		**OpenSSH Fixed**		**Ubuntu Backport**	
ID	CVSS	Date	Date	Lag	Date	Lag
2015-5352	4.3	2015-08-02	2017-07-01	699	2015-08-14	12
2015-5600	8.5	2015-08-02	2015-08-11	9	2015-08-14	12
2015-6563	1.9	2015-08-24	2015-08-11	-13		
2015-6564	6.9	2015-08-24	2015-08-11	-13		
2015-6565	7.2	2015-08-24	2015-08-11	-13		
2015-8325	7.2	2016-04-30	2016-08-01	93	2016-04-15	-15
2016-0777	4	2016-01-14	2016-01-14	0	2016-01-14	0
2016-0778	4.6	2016-01-14	2016-01-14	0	2016-01-14	0
2016-1907	5	2016-01-19	2016-01-14	-5	2016-05-09	111
2016-1908	7.5	2017-04-11	2016-02-29	-407	2016-05-09	-337
2016-3115	5.5	2016-03-22	2016-03-10	-12	2016-05-09	48
2016-6210	4.3	2017-02-13	2016-08-01	-196	2016-08-15	-182
2016-6515	7.8	2016-08-07	2016-08-01	-6	2016-08-15	8
2016-8858	7.8	2016-12-09	2016-12-19	10	(disputed)	
2016-10009	7.5	2017-01-04	2016-12-19	-16	2018-01-22	383
2016-10010	6.9	2017-01-04	2016-12-19	-16	2018-01-22	383
2016-10011	2.1	2017-01-04	2016-12-19	-16	2018-01-22	383
2016-10012	7.2	2017-01-04	2016-12-19	-16	2018-01-22	383
2016-10708	5	2018-01-21	2016-12-19	-398	2018-11-06	289
2017-15906	5	2017-10-25	2017-10-03	-22	2018-01-22	89
2018-15473	5	2018-08-17	2018-08-24	7	2018-11-06	81
2018-15919	5	2018-08-28	2018-10-19	52		
2018-20685	2.6	2019-01-10	2019-04-17	97	2019-01-22	12
2019-6109	4	2019-01-31	2019-04-17	76	2019-02-07	7
2019-6110	4	2019-01-31	2019-04-17	76		
2019-6111	5.8	2019-01-31	2019-04-17	76	2019-02-07	7
2019-16905	7.8	2019-10-09	2019-10-09	0		
			median	-5		12

We observe that the near simultaneous publication of the CVE and upstream patch has very little impact on the deployment of vulnerable servers. The steady increase in vulnerable OpenSSH servers continues for slightly more than one year until the Ubuntu backport is published, at which point the patch is rapidly applied to more than one million machines, followed by a steady linear increase in the subsequent months and years. From this one example, at least, it appears that the availability of backports is by far the dominant factor in applying updates to eliminate software vulnerabilities.

We analyzed the plots of all 27 CVEs (available in Appendix A) to see if the same trend held true. We see that many IP addresses patch very quickly as soon as a backported patch for a CVE is released, although some never patch. For the CVEs that do not have a security backport (with the darker backgrounds), the results are less predictable. Often, the number of vulnerable servers naturally goes down over time as users update their software for other reasons (e.g., CVE-2015-6563). In other cases, the number of vulnerable servers continues to increase well after the CVE was published (e.g., CVE-2018-15919).

Another consistent finding from inspecting the graphs is that the publication of the CVE and upstream patch is *not* the catalyst for updates. Rather, it is consistently the backport that sparks an uptick in patches to plug vulnerabilities.

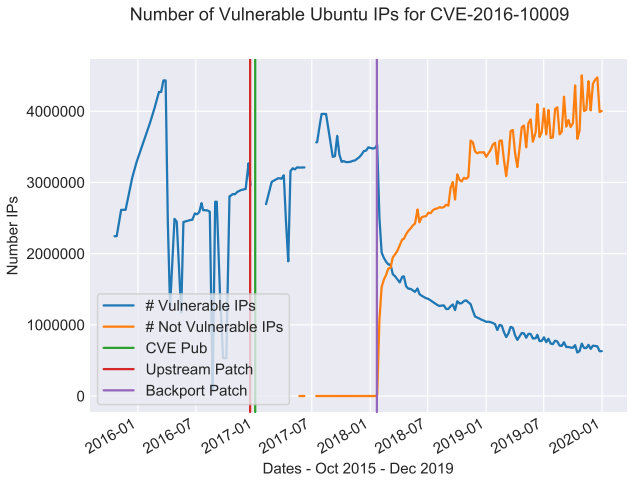
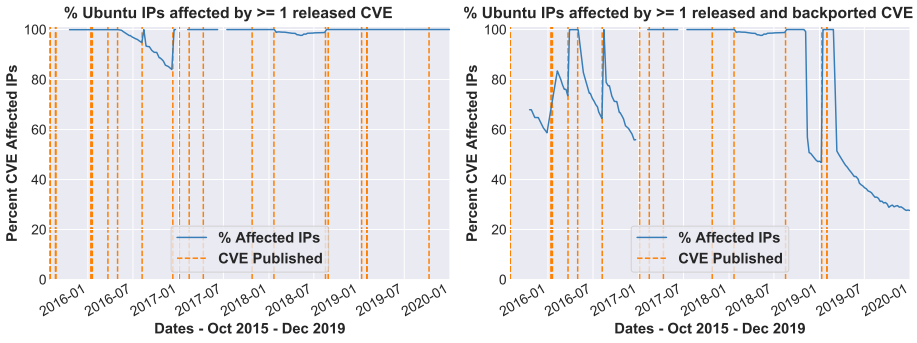


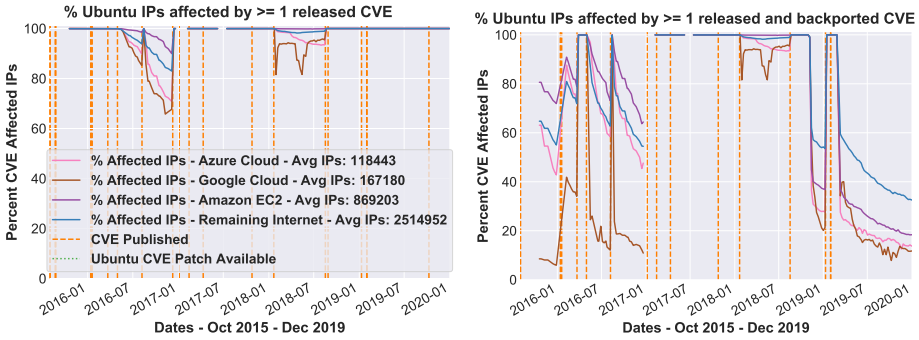
Fig. 4: Number of IP addresses that are affected by CVE-2016-10009 over time



(a) % affected by ≥ 1 CVE

(b) % affected by ≥ 1 CVE (Ubuntu backports only)

Fig. 5: Fraction of Ubuntu OpenSSH servers with vulnerabilities over time.



(a) % affected by ≥ 1 CVE

(b) % affected by ≥ 1 CVE (Ubuntu backports only)

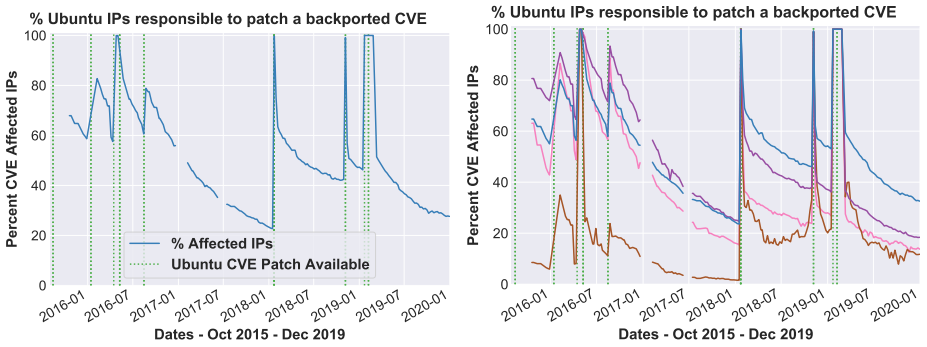
Fig. 6: Fraction of Ubuntu OpenSSH servers with vulnerabilities over time split by cloud provider.

We now collapse back down from the 27 individual plots in Appendix A that capture exposure to individual CVEs to a single, aggregated view of the presence of vulnerabilities in OpenSSH over time. Ultimately, what matters from a security perspective is whether systems have any unpatched vulnerabilities present.

We start by calculating the fraction of Ubuntu OpenSSH servers throughout the Internet which are affected by at least one published CVE. We see in Figure 5a that almost all OpenSSH servers are affected by at least one CVE throughout the time of our measurement, saving a small percentage that quickly updated in late 2016. Given that not all CVEs received a backported patch in Ubuntu, this result is perhaps inevitable, but it is alarming nonetheless.

To account for this, we construct additional measures that focus only on the 20 CVEs which have an associated Ubuntu backport. Figure 5b again plots the percentage of Ubuntu OpenSSH servers that affected by one or more vulnerabilities with an available backport patch. The percentages here are a bit better, falling to around 60% of hosts vulnerable to at least one CVE before new vulnerabilities are published, rendering all hosts vulnerable until a backport can be issued. Note that for two years in 2016 to 2018 new CVEs were consistently published before backports to the older CVEs were disseminated. The largest reason for this is that several CVEs were released in 2016 which did not receive a patch until 2018, so in the meantime, every LTS version of Ubuntu that was stuck on the upstream OpenSSH version without a security backport was vulnerable. At that point, the only way to not be vulnerable to those CVEs would be to install a fixed upstream version of OpenSSH directly. From the above plots, we can see that no matter how quickly one applies security backports, there is still a chance that the server is vulnerable to at least one CVE.

One could argue that the measurements in the previous graphs are "unfair" to system administrators because their servers will remain vulnerable even if they



(a) all Ubuntu OpenSSH servers

(b) split by cloud providers - Legend shared with Figure 6a

Fig. 7: Fraction of Ubuntu OpenSSH servers which are fully patched with available patches that fix vulnerabilities over time.

follow best practice and apply security patches as soon as the backports are published. To distinguish between those making a best effort and those simply not applying available security patches, Figure 7a instead plots the percentage of machines which have not applied all available CVE-related backport patches. This can be interpreted as the fraction of machines responsible to patch a vulnerability that have not done so.

Here, the findings are better, but still quite discouraging. The green dashed vertical lines here indicate when the backport patch is published. Hence, when a new patch is published, the percentage who can apply the patch but have not jumps to 100, before rapidly diminishing. We can see that during periods when relatively few backports are issued, the population can catch up, with the unpatched share falling to nearly 20% at one point in early 2018. However, we can also see that in other cases, when there are multiple backports issued in a row, the servers cannot keep up with applying all of the patches. Hence, an increasing frequency of distributing backports can in fact make it harder for systems to maintain security.

We utilize the previous method for distinguishing cloud providers from the remainder of the Internet for Figures 6a, 6b, and 7b. Once again, machines running on one of the three main cloud providers are generally patched faster than the remainder of the Internet. While all perform better, a greater percentage of machines on Google Cloud is consistently patched and not vulnerable to CVEs. We also note that where the cloud providers do best is in rapidly applying backports when they become available. They do not appear to upgrade systems by applying upstream patches, which is shown in the similarly poor performance in Figure 6a. Even when hosted in the cloud, most machines are vulnerable to at least one vulnerability most of the time.

5 Limitations

One limitation of our work is that we narrow our focus to OpenSSH security backports on Ubuntu in order to obtain a reliable views of update level. This leaves a large percentage of IP addresses with unknown software update levels due to either using a different operating system or Linux distribution.

An additional limitation is the reliability of using of cloud IP address mappings gathered in 2021 on historical data dating to late 2015. As more servers move to cloud providers, those providers may have needed to acquire more IPv4 addresses over the years, causing the IP mappings to not be constant over time. We attempt to account for this by comparing the cloud IP mappings to 10 historical snapshots of MaxMind's GeoIP dataset [4] spanning from July 2015 to January 2020 at roughly six month intervals. For each Censys snapshot date, the nearest GeoIP mapping is compared to the given cloud provider's announced mapping from 2021. For example, with Amazon EC2, we verify that "amazon" is in the GeoIP mapping. A similar pattern is followed for Google and Microsoft. IP addresses that conflict with the organizations listed in the nearest GeoIP mapping are not counted in our figures as either a cloud provider or with the remaining Internet. For Azure and Google Cloud, no IP addresses are omitted due to the lack of conflicting mappings, but an average of 3.6% (ranging from 1% to 5%) of Amazon EC2 IP addresses are filtered at each snapshot.

6 Related Work

In 2015, Durumeric et al. [12] released Censys, which builds upon their work on fast Internet scanning with Zmap in [14]. Censys scans the Internet using Zmap and Zgrab and stores the information in a database. The Internet-wide scan data can be queried by researchers through either their web frontend or through Google BigQuery [1]. Historical Censys data can be queried as well from Google BigQuery. We utilize Censys extensively for this work through researcher access on Google BigQuery.

The empirical measurement of security patches and vulnerabilities is highly relevant to our work. Durumeric et al. [13] follow the release and subsequent patching of the Heartbleed vulnerability. Li and Paxson [15] analyzed thousands of security patches in the National Vulnerability Database [8]. An interesting finding is the difference in the time from when a CVE ID is publicly disclosed and when it appears in the NVD database. This disparity helps explain how a Ubuntu backport can fix a vulnerability that was not "published" until later (see Figure 2).

O'Hare [17] utilizes Censys and Shodan to identify vulnerabilities on Internet-wide scans. The possibility of backporting security patches to fix CVEs is mentioned, but it does not seem that steps were taken to account for this. Demir et al. [11] analyze 5.6M websites and discussed the update behavior of many types of HTTP(S)-related software (port 80 and 443) and libraries. It also discusses the implications of CVSS scores to updates. It does not appear to consider security backports of server software, which is a focus of our study.

Several researchers have used surveys and interviews to better understand the update process of system administrators, which is relevant to our work in that it can provide explanations for why software updates are applied or not. Li et al. [16] conducted over 100 surveys and 17 qualitative interviews with system administrators and outline what they found to be the 5 stages of the system administrator update process. Similarly, Tiefenau et al. [18] conducted 7 qualitative interviews and 67 online surveys. Both Li et al. and Tiefenau et al. discuss the obstacles that delay or prevent system administrators from applying updates. They also demonstrate the variance in how system administrators approach updates. While most agree on the necessity of timely updates for the sake of security, a minority did not.

7 Concluding Remarks

Despite its importance for cybersecurity, measuring the extent to which software is up-to-date at Internet scale has not often been attempted. One reason why is that it is often hard to construct an accurate picture with external measurements. In this paper, we have demonstrated that simple approaches to measuring outdatedness based on version information appearing in publicly observable banners often fall short. Instead, we have shown that by focusing on the special cases where we can observe the presence of backports, we can construct a more accurate global measurement for the case of the 4 million-plus servers running OpenSSH on Ubuntu Linux.

We find that these backports do in fact trigger the application of security patches for a significant fraction of the population, much more than vulnerability announcements or updates directly from the software developer. We also observe that when backports are not created, these vulnerabilities tend to remain unfixed for most of the population. Moreover, the frequency of introducing new vulnerabilities has ensured that most servers remain vulnerable most of the time. While we have also presented evidence that cloud providers do a better job, it is not enough to keep hosts running on those platforms from being consistently laden with unpatched vulnerabilities.

Acknowledgments

This research was supported by the Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-0152. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

References

1. BigQuery: Cloud Data Warehouse, https://cloud.google.com/bigquery
2. Censys, https://censys.io/
3. Censys Opt Out, https://support.censys.io/hc/en-us/articles/360043177092-Opt-Out-of-Scanning
4. GeoIP® Databases & Services: Industry Leading IP Intelligence | MaxMind, https://www.maxmind.com/en/geoip2-services-and-databases
5. GitHub, https://github.com/
6. Launchpad, https://launchpad.net/index.html
7. Launchpad: Publishing history: Openssh package : Ubuntu, https://launchpad.net/ubuntu/+source/openssh/+publishinghistory
8. NVD, https://nvd.nist.gov/
9. Orbis | Compare Private Company Data | Bureau van Dijk, https://www.bvdinfo.com/en-us/our-products/data/international/orbis
10. What is backporting, and how does it apply to RHEL and other Red Hat products? https://www.redhat.com/en/blog/what-backporting-and-how-does-it-apply-rhel-and-other-red-hat-products
11. Demir, N., Urban, T., Wittek, K., Pohlmann, N.: Our (in)Secure Web: Understanding Update Behavior of Websites and Its Impact on Security. In: Hohlfeld, O., Lutu, A., Levin, D. (eds.) Passive and Active Measurement. pp. 76–92. Lecture Notes in Computer Science, Springer International Publishing, Cham (2021)
12. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A Search Engine Backed by Internet-Wide Scanning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15. pp. 542–553. ACM Press, Denver, Colorado, USA (2015), http://dl.acm.org/citation.cfm?doid=2810103.2813703
13. Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., Halderman, J.A.: The Matter of Heartbleed. In: Proceedings of the 2014 Conference on Internet Measurement Conference. pp. 475–488. IMC '14, Association for Computing Machinery, New York, NY, USA (Nov 2014). https://doi.org/10.1145/2663716.2663755, https://doi.org/10.1145/2663716.2663755
14. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: Fast Internet-wide Scanning and Its Security Applications ZMap: Fast Internet-Wide Scanning and its Security Applications. In: 22nd USENIX Security Symposium (USENIX Security 13). USENIX Association, Washington, D.C. (Aug 2013), https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric
15. Li, F., Paxson, V.: A Large-Scale Empirical Study of Security Patches. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 2201–2215. CCS '17, Association for Computing Machinery, New York, NY, USA (Oct 2017). https://doi.org/10.1145/3133956.3134072, https://doi.org/10.1145/3133956.3134072
16. Li, F., Rogers, L., Mathur, A., Malkin, N., Chetty, M.: Keepers of the Machines: Examining How System Administrators Manage Software Updates p. 16 (2019)
17. O'Hare, J., Macfarlane, R., Lo, O.: Identifying Vulnerabilities Using Internet-Wide Scanning Data. In: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3). pp. 1–10. IEEE, London, United Kingdom (Jan 2019). https://doi.org/10.1109/ICGS3.2019.8688018, https://ieeexplore.ieee.org/document/8688018/

18. Tiefenau, C., Häring, M., Krombholz, K., von Zezschwitz, E.: Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators. In: Sixteenth Symposium on Usable Privacy and Security ({SOUPS} 2020). pp. 239–258 (2020), https://www.usenix.org/conference/soups2020/presentation/tiefenau

Appendix A Plots of Ubuntu IPs affected by CVEs

Below are the plots (similar to Figure 4 of vulnerable Ubuntu IPs per CVE within the October 2015 through December 2019 measurement period ordered sequentially by vulnerability publication date. Plots with a darker background do not have a backport on Ubuntu.


Number of Vulnerable Ubuntu IPs for Given CVE Over Time
CVE-2015-5352
CVE-2015-5600
CVE-2015-6563
CVE-2015-6564
CVE-2015-6565
CVE-2015-8325
CVE-2016-0777
CVE-2016-0778
CVE-2016-1907
CVE-2016-1908
CVE-2016-3115
CVE-2016-6210
CVE-2016-6515
CVE-2016-8858
CVE-2016-10009
CVE-2016-10010
CVE-2016-10011
CVE-2016-10012
CVE-2016-10708
CVE-2017-15906
CVE-2018-15473
CVE-2018-15919
CVE-2018-20685
CVE-2019-6109
CVE-2019-6110
CVE-2019-6111
CVE-2019-16905
Number IPs in millions
Vulnerable IPs
Not Vulnerable IPs
CVE Pub
Upstream Patch
Backport Patch