Sharing is Caring: Towards Analyzing Attack Surfaces on Shared Hosting Providers

Jan Hörnemann,[1] Norbert Pohlmann,[2] Tobias Urban,[3] Matteo Große-Kampmann[4]

Abstract: In this paper, we shed light on shared hosting services' security and trust implications and measure their attack surfaces. To do so, we analyzed 30 shared hosters and found that all of them might leak relevant information, which could be abused unnoticed. An adversary could use this attack surface to covertly extract data from various third parties registered with a shared hoster. Furthermore, we found that most hosters suffer from vulnerabilities that can be used by an internal attacker (i.e., someone using the service) to compromise other hosted services or the entire system.

Keywords: shared hosting; data leaks; cloud computing

1 Introduction

Shared hosting providers offer web services like storage, hosting, or data warehousing at affordable and competitive prices. Shared hosting vendors often advertise that they are suitable for small and medium-sized enterprises (SMEs) as an easy and affordable way to offer web services. Shared hosting is affordable because one shares the hosting hardware with other users.

In contrast to *dedicated hosting*, shared hosting provides the same computing and storage resources to different parties. This circumstance makes shared hosters a rich target for malicious actors because they might be able to not intrude into one but multiple entities at the same time. Shared hosting providers need to be aware of these risks and need to account for them accordingly (e.g., by implementing suiting security measures).

In this work, we focus on the technical security of *shared hosting* services. Based on 30 randomly selected hosters, we analyze if and to what extent an adversary could get access to the private data of other users or even overtake (some) services on the server. To do so, we evaluate common security threats used for privilege escalation. When analyzing the shared hosting providers, we processed over 3.5 million log files, analyzed 219 SUID binaries, and found 4,319 usernames. Finally, we assess the deployed security mechanisms of three randomly sampled hosters in a case study.

[1] AWARE7 GmbH, Institut für Internet-Sicherheit jan@aware7.de

[2] Insitut für Internet-Sicherheitpohlmann@internet-sicherheit.de

[3] Institut für Internet-Sicherheit urban@internet-sicherheit.de

[4] Hochschule Rhein-Waal, AWARE7 GmbH, Germany matteo.grosse-kampmann@hochschule-rhein-waal.de

In summary, we make the following contributions:

- We analyze real-world shared hosters and identify tactics adversaries could use to escalate their privileges or exfiltrate data from shared hosters.
- For 30 randomly sampled hosters, we check for potential vulnerabilities an adversary could abuse (e.g., based on the installed kernel version) or sensitive data the hosters might leak (e.g., usernames and passwords).
- In three case studies, we assess specific implemented security mechanisms, and find that these hosters expose valuable information adversaries can use to attack the system (e.g., endpoints of other users or installed software).

2 Background on Shared Hosting

Web hosting services allow users and companies to make their websites easily accessible to everyone. Several models exist to implement such hosting services (e.g., cloud hosting, dedicated hosting, or shared hosting). Hosters choose the shared hosting model because it is easy and affordable to set up and maintain; therefore, they can offer budget-friendly service. However, from a security perspective, such an approach has several downsides. An obvious one is that once a hosted service is compromised or registered by a malicious actor, the security of all hosted services on the same machine is in jeopardy [Ta17b]. From a privacy perspective, shared hosting also comes with several challenges that are probably not considered by some users [Hu20]. One of these challenges is that all services that use the same shared machine get access to shared resources of the operating system that might leak sensitive information (e.g., log files or password files) [Mi12]. The information gained from these resources might provide meaningful insights about the hosted services, their users, or used technologies. Adversaries might collect and sell this data (e.g., data breaches) or use it to compromise the hosted services (e.g., overtake the service).

3 Methodology

In this section, we provide an overview of our underlying attack model, describe the selection of the analyzed hosting providers, and discuss our measurement approach.

Attack Model. The attacker model for shared hosters potential information leakage or privilege escalation is quite simple. We assume that the adversary actively registers an account at a given shared hoster and that the adversary gets, from her point of view, random resources to work with. The main objective of the actor in our threat model is to exfiltrate as much data as possible extracted from other hosted systems or their users or to overtake services from different users. Hence, the adversary has no direct control over which machines of the shared hoster they can access and which application might run on it. Furthermore, we assume that the adversary can connect to the server (e.g., via `ssh`) and interact with it within the boundaries defined by the hoster (e.g., user permissions). For ethical, reasons we do not actively abuse mechanisms to increase our privileges (i.e., “privilege escalation”), but we use the available resources for data reconnaissance of potential victims. Our attacker focuses on collecting “Common Vulnerabilities and Exposure” (CVE) information, which references publicly known security vulnerabilities [Mi05]. However, we report on any instances where this might be possible. Furthermore, this will strictly decrease the possible detection of any malicious actions by the provider, if any are in place [CBF13]. Our research uses the CVE database “CVE Details” [Se23], and we restricted the analyzed CVEs to locally exploitable vulnerabilities because we already have local access to the web server.

Web Hoster Selection. The selection of shared web hosters to analyze is not a straightforward task because the computation of market shares is not feasible without the internal knowledge of all participants (e.g., in terms of hosted applications). Furthermore, we cannot use techniques used by previous work (e.g., [Ta16; Ta17a], as those used an extensive database of passive DNS resolutions in combination with `WHOIS` requests. Due to legal restrictions, particularly the GDPR, this approach does not work anymore because `WHOIS` data does not always contain private information anymore. Hence, we rely on a compiled list of web hoster market shares provided by *Datanzye*, which is publicly available [Da20]. While the computation of this list and used metrics are unknown to us, using this list at least increases the comparability and reproducibility of this work.

From a provided list, we chose all hosters that offer shared hosting and allow users to connect to the server via `SSH`, and ended up with 30 hosters. Henceforth, we use pseudonyms to preserve the hosters anonymity and avoid putting any live systems in danger.

Measurement Approach. This subsection describes our approach to exfiltrate data from the shared hosters and what we did to analyze the data we obtained.

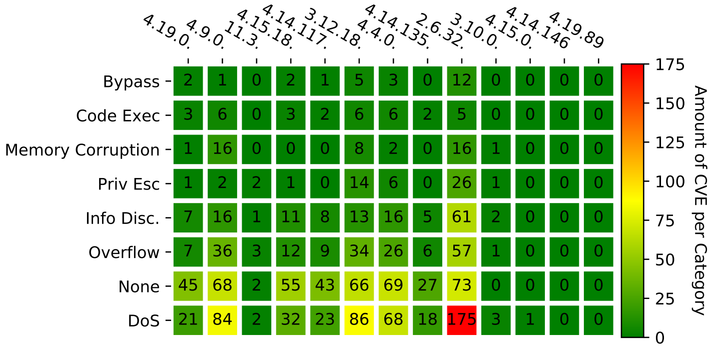
To conduct our analysis, we registered a (paid) account with each of the 30 service providers so that we could utilize the services. After we obtained an account, we connected to the shared resources via `SSH` and uploaded our analysis script. We run the script with as many privileges as possible, using `chmod`.

Our analysis script is designed to measure the attack surface of each provider. It analyzes if security measures are deployed while being as non-intrusive as possible. We focused on thirteen potential vulnerabilities or leakage indicators, from sniffing TCP connections on port 80/21 over port forwarding to passwords in logs to checking which files get root privileges during execution. A complete catalog of analyzed properties can be found in the supplementary material to our work (cf. Section A).

4 Results

This section presents the results of our shared hoster analysis.

Fig. 1: Distribution of vulnerability types per kernel version



4.1 Analyzing Operating System Security Level

An essential part of each application's security concept is the consideration of the used platform to host the application itself. Therefore, in the *Unix* world, one has to consider the installed kernel version and if one can use it securely in the field. In our measurement of the 30 hosters, we identified 13 different kernel versions ranging from *3.10.0* to *4.19.89* with *3.10.0* being the most common one used by 13 (43%) hosters. The second most common one, *2.6.32*, is used by 4 (13%) shared hosting providers. This finding is troubling because at the time we made the measurement, all kernel versions below *4.4* are not supported anymore and are marked as "End Of Life" (EOL). For example, according to the maintainer of kernel version *3.10.0* it should have been updated in 2017 [Ta17c]. According to the *National Vulnerability Database* (NVD) [Na23], 83% (25) of the observed kernels have at least one listed vulnerability (see Figure 1).

For all observed kernel versions, the known vulnerabilities were identified in the form of CVEs. For three of the observed kernel versions, we found CVEs that allow privilege escalation. For all kernel versions for which there are known CVEs categorized as "Privilege Escalation", there is at least one CVE vulnerability that leads to complete loss of integrity. Thus, for the majority of all systems examined (28[93.3%]), there are known CVEs and thus vulnerabilities through which a local user can gain root privileges. Especially one of the most used kernel versions (`2.6.32`) is vulnerable to *26* CVEs categorized as "Privilege Escalation". The often used kernel version `2.6.32` is vulnerable to the most vulnerabilities ($n = 425$). Of these vulnerabilities 175 are known to lead to a denial of service and 61 can lead to information disclosure, which expose sensitive internal information from the system's memory. A closer look at the noticeable number of denial of service vulnerabilities shows that the majority of these vulnerabilities (146, 83.4%) might lead to a *complete* loss of system availability.

A further 16% of the denial of service vulnerabilities of the kernel version lead to availability

Fig. 2: Impact CVEs categorized as Privilege Escalation on the integrity

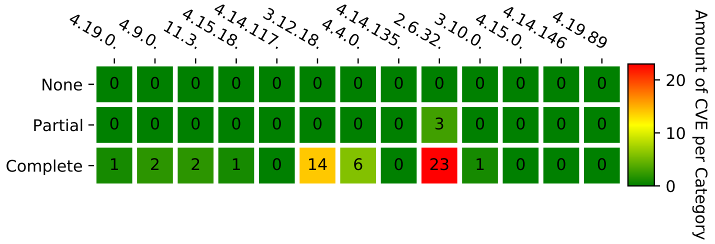
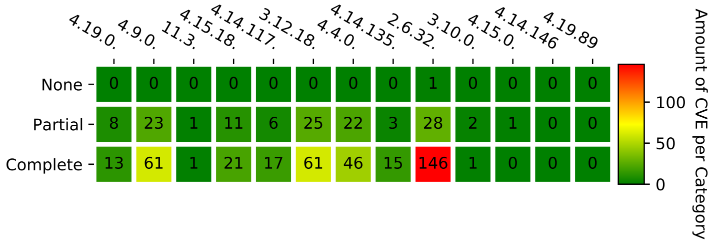


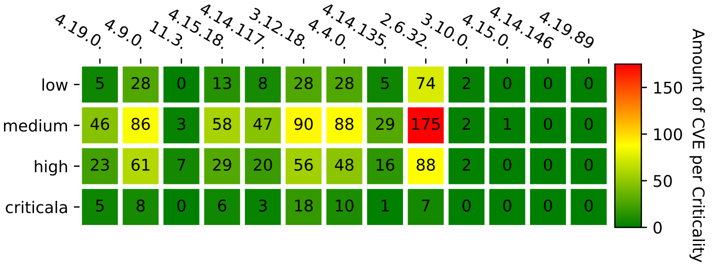
Fig. 3: Impact CVEs categorized as Denial of Service on the availability



losses of individual components of the system. In general, Figure 3 shows that the majority of all known denial of service attacks and CVEs lead to a complete loss of system availability. Another remarkable aspect of kernel version `2.6.32` is that no other tested kernel has as many privilege escalation vulnerabilities, as shown in Figure 2. These 26 privilege escalation vulnerabilities pose a threat to the entire system. 88% of these CVEs might compromise the complete integrity of the system. This means that through most known CVEs, a local adversary might gain root privileges and, thus, could access all the data of all other users. This observation poses a great threat to the security of the system, since the adversary might be able to execute code on the system and thus use these local vulnerabilities to gain higher privileges on the system. Overall, Figure 2 shows that there are only five kernel versions (version `4.15.0`, `4.14.135`, `4.14.146`, `4.19.89` and `4.14.117`), for which there is no known CVE categorized as privilege escalation at the time of analysis.

To illustrate the criticality of these identified vulnerabilities, we evaluated the score of the CVEs. We mapped them as seen in Figure 4. The majority of CVEs in all kernels are classified with the criticality "Medium." These are primarily vulnerabilities that only lead to the complete loss of one security property, such as availability in the event of denial of service attacks. It can be stated that there is at least one critical known vulnerability for all kernel versions, except for kernel `4.15.0`, `4.14.146`, and `4.19.89`. Critical CVEs always have an impact on several security properties, such as integrity and availability. In addition to these critical CVEs, Figure 4 shows other known vulnerabilities that are categorised as high. Medium CVEs can also be identified in the majority of the kernel versions examined

Fig. 4: Distribution of criticality of known vulnerabilities per kernel version



and lead to a complete loss of all information security properties. Especially noticeable is kernel `2.6.32`, which contains 175 medium vulnerabilities. This number corresponds to the number of known denial of service vulnerabilities of this kernel version.

Due to the partly very high number of vulnerabilities, we decided to examine five randomly picked of the thirty tested hosters in detail. We extracted the following data: (1) Unix distribution, (2) distribution Version, (3) kernel version, and (4) chipset architecture.

We have used this information to specifically search for local exploits. For this research we used open source tools like *linux-exploit-suggester*, *searchsploit*, and the *ExploitDB*. From this information, we were able to evaluate which local privilege escalation exploits are most likely usable on the system under examination. It could be determined that there are publicly known exploits for all hosters that can be used for a privileged escalation. The exploits to which the hosters are vulnerable are very similar. Four out of five hosters are potentially vulnerable to the two exploits `sudo pwfeedback` and `PIE_stack_corruption`. Since the vulnerable hoster is a live system and we would put other users at risk we did not exploit the mentioned exploits.

4.2 System Specific Findings

In this chapter we examine key system vulnerabilities, like file permissions with extended rights to potential gaps in logfiles.

Programs with set SUID bit. The script checks for files where the SUID bit is set. The SUID or `setuid` bit gives a file or program extended rights. If the SUID bit is set, the program is executed in the user context but also in the context of the owner. We found a lot of programs on the systems that run with root rights. The range is between 0 and 41 programs. One hoster also used a program which can be used to get a root shell according to GTFOBin (rsync) [GT20]. Getting a root shell as a user that is not in the administrator group could have diverse effects for the other users in terms of confidentiality, integrity, and availability.

Writable Files in /etc/. The folder `/etc/` contains almost all the important system configurations. If an attacker has write access on this directory, she can configure services or functions to get access to even more services. In our study, we had access to the configuration folder of 131 services and could write to the files there. We have been able to change configs for mail servers (Exim), web server (Apache's *srm.conf*), PHP and things like `/etc/resolv.conf`. An attacker could use this property to change configurations of shared services like mail servers and route mails away from the addressed recipient to his inbox or could include malicious code on websites that are hosted on the shared hosting provider.

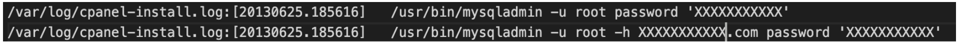
Unmounted Filesystems. We have looked in `/etc/fstab` for disks that are currently not mounted. These may theoretically contain sensitive data which an attacker can read when he mounts the disk. There was only one hoster that "attached" an unmounted disk. This is a SWAP partition. This is used when the RAM of a computer gets full. If this is the case, Linux can write parts of the RAM into the SWAP. A normal user should not have access to this data because if you mount the disk you can potentially read parts of the RAM. We successfully mounted the partition but did not perform analysis on the contents of the SWAP due to ethical considerations.

Internal Network Services. Open ports are an potential entry point for adversaries because they increase a system's attack surfaces. We focus on the services which are not offered on the Internet but the internal network. These ports towards the internal network can be used by an adversary to laterally move from the internet-facing system towards the internal network to compromise the system even further [PJL16]. Most hosters (27) do not offer services on the internal network that they do not provide on the Internet. One of the verified hosters offered an SSH service on port 8022 on a local interface that was not accessible via the Internet. We found that the SSH service provided on the internal interface is an older version than the one offered on the Internet. Most of the time, we observed DNS services on the internal network.

Passwords in Logs. In this step, we also looked at which files in `/var/logs/`. A total of *3.460.841* lines of readable log files were identified. We went through all the available lines and looked for the following keywords: (1) *password*, (2) *pwd*, (3) *username*, (4) *user*, (5) *mail*, and (6) *email*.

We were able to gather a *5.231* lines in log files which contain at least one of our keywords. Interestingly, an adversary can find partly complete login requests with usernames and additional corresponding details. The amount of log files an adversary could investigate and the amount of intelligence she gathers is critical. In nearly one-third (*27%*) of the sampled set, we retrieved readable logs which contain one of the following keywords. (1) *password*, (2) *pwd*, (3) *username*, (4) *user*, (5) *mail*, and (6) *email*. An adversary could use these lines and information to gain access or data from another third party using the same shared hoster. Since the content of the files is very sensitive the implications of what an attacker might do with them are manifold. They can include local passwords so that an attacker can access the

Fig. 5: The readable log shows an SQL login sequence; we redacted the password.



```
/var/log/cpanel-install.log:[20130625.185616]   /usr/bin/mysqladmin -u root password 'XXXXXXXXXXX'
/var/log/cpanel-install.log:[20130625.185616]   /usr/bin/mysqladmin -u root -h XXXXXXXXXXX.com password 'XXXXXXXXXXX'
```

local virtual system of the user, local databases of the user as seen in 5 or an attacker can even access remote services that were connected to the shared hoster.

Read/Write Permissions. We have searched the system for files that we are allowed to read. We found many files (41.180), so we limited the analysis to sensitive folders e.g., webroots or homefolders. We find hosters that do not set read permissions well. Three hosters allow us to view the data of other users in their webroots. Another hoster allows us to view all data in the home folders of the other users. In general, read rights are not handled as strictly as write rights. An attacker can gain insights to the data stored on other virtual machines on the same shared hoster. This might lead to a breach of confidentiality of user data.

5 Case Studies

In the following, we discus three randomly sampled hosters (`A`, `B` and `C`) in detail.

Running Software. For our case studies, we start by analyzing the `/etc`, `/usr/bin` and `/bin` folders to get a grasp of the installed software. For our analysis, we excluded all preinstalled program binaries that ship with the respective *Linux* distribution. Furthermore, we combined results of programs that lead to multiple entries in the respective folders, e.g., *MySQL* that also generates entries like `mysql_config`, *mysql_config_editor*, etc. We found that there are multiple software tools such as *Python*, *PHP*, and *Perl* available in up to 12 different versions on the hosters. However, most programs are available in only one version. On average, there were 7.75 different *Python* versions installed, followed by *PHP* with an average of 6.25 different versions.

By identifying the versions of the installed software, we can query the corresponding CVE databases to find known vulnerabilities for them. In total 22 CVEs were identified and half of them could partial compromise the confidentiality (59.09%), integrity (54.55%) and availability (59.09%) of a hoster.

Most vulnerabilities are introduced by vulnerable PHP versions, which introduce include 13 potential vulnerabilities. On average, 8.5 vulnerabilities are found on the distinct web hosts. `C` has fewer vulnerabilities (3) CVEs, than `A` (8), and `B` (11). However, we found CVEs with partial and complete threats to confidentiality, integrity, and availability for each hoster.

Configuration Files. On each shared hoster, we identified the installed Unix distribution with the installed version, the kernel version and system architecture. We used three open-source tools (*linux-exploit-suggester* [Zi21], *ExploitDB* [Se21], and *linenum.sh* [Re20]) to identify option that could lead to local privilege escalations that could possibly be used by an adversary. We focused on local privilege escalations attacks because they might allow an adversary to gain access of (personal) data other users that utilize the shared hosters. Note that there is no guarantee that the exploits identified will work (i.e., we report an upper bound). Overall, we identified 50 potential exploits and at least 2 (max: 16) vulnerabilities for each hoster. For example, one allows to exploit position independent executables (CVE-2017-1000253). CVE-2017-1000253 is a local privilege escalation that exploits Linux kernel errors when loading PIEs (Position Independent Executables) [Qu17].

Users On a System. Since the user management on all three hosters is implemented differently, we report on the findings for them separately.

A: This hoster does not assigned a login shell to each user. A closer analysis of `A`'s `passwd` file shows that for 141 users a comment indicates the owner or purpose of the account. For example, for 10 accounts an URL of the hosted service was given and for the other ($n = 57$) accounts the purpose of the account could be determined (e.g., “khaccount”).

B: This hoster does not store the system accounts at the standard location (e.g., `passwd`) but uses `/etc/cbi/passwd`. In that file we found 4,352 entries. Of those, only 14 (0.32%) entries seem to be for technical users, and the remaining users (4,338) share the same home directory and group (`/customers/homepages/xx/xx/htdocs`). The */etc/group-* and */etc/cbi/group* assign this GID to the “ftpusers” group. This observation is critical because all clients that belong to the same group can access a file once appropriate permissions have been assigned in the group.

C: For *C* we can identify 153 entries in the `passwd` file. Looking closely at these entries, one entry stands out that includes the value “root” (“uberrroot”) in its name and that this could be a root account of the hoster.

Summary: In summary, it can be stated that for the three web hosts where a meaningful *passwd* was available, a lot of entries could be identified that were other customers of the web host or root accounts.

6 Related Work

With the growing popularity of hosting services on the Web, a large variety of work focuses on different aspects of this new ecosystem. Tajalizadehkhoob et al. found that the ecosystem consists of more than 45,000 hosting providers with only little consolidation in the market [Ta16]. Several papers present different attack vectors on hosting providers [DB15; MTS19; So11], most notably the applicability of the *Rowhammer* attack in cloud environments by Yuan et al. [Xi16]. These works present a novel attack vector or defense mechanism but do not account for data leakage across services, which is the focus of this paper.

Shared hosting is vulnerable to different attacks due to its concept. For example, Canali et al. have shown that shared hoster often cannot detect a direct attack on their infrastructure [CBF13], Nikiforakis et al. highlight the security implications due to the lack of isolation [NJJ11], and different works show that shared hosters are more often used with malicious intentions, in contrast to other hosting types [Ta16; VWM16]. Similar to our study is the work of Mirheidari et al. , in which they discuss the theoretical possibility of reading/writing log files on shared hosting platforms [Mi12] and discuss potential attack vectors resulting from this practice (i.e., poising the log or snooping information from it) [Mi13]. Our work shows the extent and severity of the leakage of such data in the field by analyzing 30 shared hosting providers. Tajalizadehkhoob et al. performed a large-scale study on the failure of shared hosters to patch the software they use [Ta17b] and show that service providers significantly impact a webpage's overall security.

7 Discussion & Limitations

Our approach comes with different limitations. Most notable is the lack of ground truth for the collected and analyzed data. We do not know if the adversary profits from the analyzed vectors since we did not abuse the found vulnerabilities for ethical reasons. Furthermore, we do not know if the analyzed companies know the weaknesses. Furthermore, we rely on an intuitive approach to test for potential options to perform privilege escalation or compromise the server. Thus, it has to be seen as a lower bound since other ways might exist. Another influence on the findings' exploitability might be mandatory access control, e.g., *SE Linux* or *AppArmor*. Further research can analyze this impact.

Our results highlight that most analyzed hosting providers provide a potentially large attack surface to remote adversaries if state-of-the-art security solutions do not monitor or protect malicious behavior. One issue that the providers should resolve is that they use older kernel versions that are exploitable. If a malicious user gains kernel privileges, there is a considerable risk of data leakage for every client hosting service on their platform.

8 Conclusion

In summary, we have shown that privilege escalation and data leakage are severe and large-scale problems with shared hosting providers. One reason is that “end of life” kernel versions are used. However, operators of shared hosting services can implement many security countermeasures to mitigate the threats posed by local attackers that have access to the resources of a machine. This work showed that most of the hosters studied are vulnerable to known vulnerabilities, which could be remedied through effective patch management. Strict client separation can also eliminate access to sensitive information in log data. Software such as *SE Linux* can be used to manage access to files and logs. Our results indicate that the shared hosting providers provide a rich attack surface for adversaries, often unknown to the customers of the service.

Acknowledgement

The authors gratefully acknowledge funding from the *Federal Ministry of Education and Research* (16KIS1628K & 16KIS1629 “UbiTrans,” and 16KIS1648 “DigiFit”, 16KISR001K & 16KISR002 “HealthNet”), and the the *Federal Office for Information Security of Germany* (grants 01MO23033B “5Guide”, 01MO23033B“5Guide” and 01MO23025B “Pentest-5GSec”).

References

[CBF13] Canali, D.; Balzarotti, D.; Francillon, A.: The Role of Web Hosting Providers in Detecting Compromised Websites. In: Proceedings of the 22nd International Conference on World Wide Web. WWW, 2013.

[Da20] Datanyze Inc.: Market Share, Accessed: 2020-05-05, 2020, URL: https://www.datanyze.com/market-share.

[DB15] Delignat-Lavaud, A.; Bhargavan, K.: Network-Based Origin Confusion Attacks against HTTPS Virtual Hosting. In: Proceedings of the 24th International Conference on World Wide Web. WWW ’15, International World Wide Web Conferences Steering Committee, Florence, Italy, pp. 227–237, 2015, ISBN: 9781450334693, URL: https://doi.org/10.1145/2736277.2741089.

[GT20] GTFOBins: GTFOBins, Accessed: 2020-06-05, 2020, URL: https://gtfobins.github.io/.

[Hu20] Huynh, T. T.; Nguyen, T. D.; Nguyen, N. T.; Tan, H.: Privacy-Preserving for Web Hosting. In: International Conference on Industrial Networks and Intelligent Systems. Springer, pp. 314–323, 2020.

[Mi05] Mitre, C.: Common Vulnerabilities and Exposures, 2005.

[Mi12] Mirheidari, S. A.; Arshad, S.; Khoshkdahan, S.; Jalili, R.: Two Novel Server-Side Attacks against Log File in Shared Web Hosting Servers. In: Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions. ICITST’12, pp. 318–323, 2012, ISBN: 978-1-908320-08-7.

[Mi13] Mirheidari, S. A.; Arshad, S.; Khoshkdahan, S.; Jalili, R.: A Comprehensive Approach to Abusing Locality in Shared Web Hosting Servers. In: Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. TrustCom’13, pp. 1620–1625, 2013.

[MTS19] Matic, S.; Tyson, G.; Stringhini, G.: PYTHIA: A Framework for the Automated Analysis of Web Hosting Environments. In: The World Wide Web Conference. WWW ’19, Association for Computing Machinery, San Francisco, CA, USA, pp. 3072–3078, 2019, ISBN: 9781450366748, URL: https://doi.org/10.1145/3308558.3313664.

[Na23] National Institute of Standards and Technology: National Vulnerability Database, 2023, URL: https://nvd.nist.gov/.

[NJJ11] Nikiforakis, N.; Joosen, W.; Johns, M.: Abusing Locality in Shared Web Hosting. In: Proceedings of the 4th European Workshop on System Security. EuroSEC'11, ACM Press, Salzburg, Austria, 2011.

[PJL16] Purvine, E.; Johnson, J. R.; Lo, C.: A graph-based impact metric for mitigating lateral movement cyber attacks. In: Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense. 2016.

[Qu17] Qualys: Linux PIE/stack corruption, 2017, URL: https://www.qualys.com/2017/09/26/linux-pie-cve-2017-1000253/cve-2017-1000253.txt.

[Re20] Rebootuser: LinEnum, 2020, URL: https://github.com/rebootuser/LinEnum.

[Se21] Security, O.: The Exploit Database, 2021, URL: https://github.com/offensive-security/exploitdb.

[Se23] SecurityScorecard: CVE security vulnerability database. Security vulnerabilities, exploits, references and more, 2023, URL: https://www.cvedetails.com/.

[So11] Somorovsky, J.; Heiderich, M.; Jensen, M.; Schwenk, J.; Gruschka, N.; Lo Iacono, L.: All Your Clouds Are Belong to Us: Security Analysis of Cloud Management Interfaces. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop. CCSW '11, Association for Computing Machinery, Chicago, Illinois, USA, pp. 3–14, 2011, ISBN: 9781450310048, URL: https://doi.org/10.1145/2046660.2046664.

[Ta16] Tajalizadehkhoob, S.; Korczyński, M.; Noroozian, A.; Ganán, C.; van Eeten, M.: Apples, oranges and hosting providers: Heterogeneity and security in the hosting market. In: Proceedings of the 2016 Conference on Network Operations and Management Symposium. NOMS'16, IEEE, pp. 289–297, 2016.

[Ta17a] Tajalizadehkhoob, S.; Gañán, C.; Noroozian, A.; Eeten, M. v.: The role of hosting providers in fighting command and control infrastructure of financial malware. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. Pp. 575–586, 2017.

[Ta17b] Tajalizadehkhoob, S.; Van Goethem, T.; Korczyński, M.; Noroozian, A.; Böhme, R.; Moore, T.; Joosen, W.; van Eeten, M.: Herding Vulnerable Cats: A Statistical Approach to Disentangle Joint Responsibility for Web Security in Shared Hosting. In: Proceedings of the 2017 ACM Conference on Computer and Communications Security. CCS '17, ACM Press, Dallas, Texas, USA, pp. 553–567, 2017.

[Ta17c] Tarreau, W.: Look back to an end-of-life LTS kernel : 3.10, Accessed: 2020-06-05, 2017, URL: http://wtarreau.blogspot.com/2017/11/look-back-to-end-of-life-lts-kernel-310.html.

[VWM16] Vasek, M.; Wadleigh, J.; Moore, T.: Hacking Is Not Random: A Case-Control Study of Webserver-Compromise Risk. IEEE Transactions on Dependable and Secure Computing 13/2, pp. 206–219, 2016.

[Xi16] Xiao, Y.; Zhang, X.; Zhang, Y.; Teodorescu, R.: One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation. In: 25th USENIX Security Symposium (USENIX Security 16). USENIX Association, Austin, TX, pp. 19–35, Aug. 2016, ISBN: 978-1-931971-32-4, URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/xiao.

[Zi21] Ziulek, M.: Linux privilege escalation auditing tool, 2021, URL: https://github.com/mzet-/linux-exploit-suggester.

A Availability of Data & Code Artifacts

To foster future research, we release our code, queries for the entire data processing pipeline and evaluation, and other supplementary information online at: https://github.com/awareseven/sharing-is-caring.