SUSE

SUSE Best Practices

Security

SUSE Solution Security Risk Report 2022

All SUSE Products

Stoyan Manolov, Head of Solution Security (SUSE)

SUSE

SUSE Solution Security is committed to delivering best in class software security to its customers and to the open source community. The primary objectives are to treat software security as an ongoing and continual process.

The goal of this report is to provide a summary of all security vulnerabilities which affected SUSE products in calendar year 2022.

Contents

1 Motivation 4

2 Background 4

3 Incident rating and tracking 5

4 When to prefer version upgrades over backports 8

5 Major security vulnerabilities in 2022 8

6 Vulnerability Management in 2022 18

7 Secure Software Supply Chain 19

8 Adopting Sigstore for our Supply Chain Security 22

9 Securing Our Product Portfolio 23

10 About SUSE 25

11 Forward-looking statements 26

12 Legal notice 27

13 GNU Free Documentation License 28

1 Motivation

SUSE Solution Security is committed to delivering best in class software security to its customers and to the open source community. The primary objectives are to treat software security as an ongoing and continual process that never ends. This implies to:

- promptly react to security incidents and deliver premium quality security updates.
- continuously improve the security related functionality in SUSE products.
- continuously contribute to the rapidly growing maturity of open source software.
- respect the open source software security principles of openness, transparency and traceability.

The SUSE Security Team addresses all aspects of software security on an ongoing basis. Any software can contain errors (both deliberate and accidental) which could affect the system's security, including design flaws, programming errors, and backdoors. In addition, software security cannot be thought of as a state you can achieve at a specific point in time. Instead, it is a process that must be executed with professional expertise and continuous development. This persistent focus is what has given open source software, Linux and SUSE an excellent reputation for security.

The objective of this report is to provide a summary of all security vulnerabilities which affected SUSE products in calendar year 2022. We will go into details on the high impact vulnerabilities which affected our products in 2022 and elaborate on how we responded to these incidents.

2 Background

A modern Linux operating system, such as SUSE Linux Enterprise Server for enterprise use or the openSUSE community distribution for home use, features a rich set of security programs and functions. Those range from access control, intrusion prevention and detection, flexible and trustworthy authentication mechanisms, encryption for files and network connections, file integrity checking utilities, to network analysis tools and monitoring/logging utilities for your system. To complement this, there are advanced tools that help you to securely configure and administer your system, and to securely download and install update packages. These utilities are standard in SUSE products. The update packages fix security bugs that have been found after

your product has been released. The security features of your Linux system are waiting for you to explore them. SUSE encourages our customers to take advantage of them to further improve the level of privacy and security that is built into every system by default.

Programs are usually written by humans, and humans make mistakes. By consequence, all software can contain errors. Some of these errors appear as instabilities (the software or the entire system crashes), while others may not have any apparent, visible effect. However, some software errors may introduce a security risk. A local or a remote attacker may be able to feed specially drafted data to the software which takes advantage of the programming error. In the case of a remotely exploitable bug, the data comes from an attached network device, such as a cable or DSL modem, or a wireless network interface card. The application then either crashes, resulting in a Denial of Service (DoS) attack, or it executes code that originates from the attacker, transferring control over the execution context from what the programmer intended to what the attacker has in mind for the exploitation of the error. Depending on the software's function, the resulting security breach can impose little or high security risks for your data and your system, potentially giving an attacker the opportunity to delete, alter or even steal your data, or use the system for his own purposes.

The SUSE Solution Security team is responsible for handling all SUSE product-related security incidents. In that team, clear and well-defined roles are assigned for tracking new incidents and coordinating needed updates. The team works with all SUSE engineering software specialists.

We use multiple sources to understand security incidents. These sources include the Mitre and NVD Common Vulnerabilities and Exposures (CVE) databases, various security mailing lists (OSS security, Linux distros, distros, bugtraq, and full-disclosure), direct reports, and other Linux vendors databases. We are also part of various pre-notification mailing lists for software components, like Xen, Samba, X.ORG. Confidential pre-notifications about vulnerabilities will be treated according to established responsible disclosure procedures.

3 Incident rating and tracking

We rate the severity of incidents with two different systems, a simplified rating system and the Common Vulnerability Scoring System (CVSS) v3.1 scoring system. The CVSS is an open framework for communicating the characteristics and severity of software vulnerabilities. It is being developed by the US-based non-profit organization FIRST.org: Its main goal is to assign the right score to a vulnerability to help security administrators prioritize responses and resources to specific threats. CVSS v3.1 scoring consists of three metric groups: Base, Temporal, and Environmental. The Base group represents the intrinsic qualities of a vulnerability that are constant

over time and across user environments. The Temporal group reflects the characteristics of a vulnerability that change over time. The Environmental group represents the characteristics of a vulnerability that are unique to a user's environment. The Base metrics produce a score ranging from 0 to 10, which can then be modified by scoring the Temporal and Environmental metrics. A CVSS score is also represented as a vector string, a compressed textual representation of the values used to derive the score. Today, SUSE uses the Base score methodology to evaluate vulnerabilities throughout the support life cycle of our products. SUSE keeps the right to adjust the final score of the vulnerability as more details become known and available throughout the analysis. The most current CVSS resources can be found at https://www.first.org/cvss/ . The CVSS v3.1 calculator used by SUSE could be found at https://www.first.org/cvss/calculator/3.1. The framework is measuring the severity of a given vulnerability, not the associated risk alone. The scoring of any vulnerability may vary with different analysts hence the final score could be slightly different between vendors impacted by that vulnerability. For a more accurate assessment of the impact, vendors and application owners must always consider factors outside of CVSS such as exposure or threat.

The security incidents are tracked in our own workflow system. Technical details are tracked in the SUSE bug-tracking system, and the updated software package is built, processed, and published by our internal “Open Build System”. Internal Service Level Agreements (SLAs) corresponding to the severity rating are monitored and reviewed regularly. Our packagers backport the required security fixes to our version of the software. To protect the stability of our customer setups, we only rarely do minor version upgrades. After receiving fixes for the affected software, four eye reviews cross-check the source patches. Several automated checks verify source and binary compatibility and the completeness of patch meta information. They also check whether patches can be installed without problems. Dedicated QA teams provide integration, bugfix, and regression testing for all updates before they are released to our customers. After the release of an update, automated processes publish the updates, update notices, and cross reference information on our CVE index pages and machine-readable OVAL and CVRF XML information.

The objective of this report is to provide a summary of all security vulnerabilities which affected SUSE products in calendar year 2022. We will go into details on the high impact vulnerabilities which affected our prour products in 2022 and elaborate on how we responded to these incidents. For a better understanding of our classification mechanisms, we have described our rating system along with the equivalency of each rating to the CVSS v3.1 scoring calculator:

TABLE 1: INCIDENT RATING AND CVSS SCORE

Rating	CVSS Score	Definition
Critical	9.0 and above	This rating is given to flaws that could be easily exploited by a remote unauthenticated attacker and lead to system compromise (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms. Flaws that require an authenticated remote user, a local user, or an unlikely configuration are not classed as critical impact.
Important	7.0 to 8.9	This rating is given to flaws that can easily compromise the confidentiality, integrity, or availability of resources. These are the types of vulnerabilities that allow local users to gain privileges, allow unauthenticated remote users to view resources that should otherwise be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow remote unauthenticated users to cause a denial of service without user interaction.
Moderate	4.0 to 6.9	This rating is given to flaws that may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances. These are the types of vulnerabilities that could have had a critical impact or important impact but are less easily exploited based on a technical evaluation of the flaw, or affect unlikely configurations. Local, persistent (service needs to be restarted) denial of service conditions for basic system services (kernel, systemd, polkit, dbus, etc.) with and without user interaction should also be rated “moderate”.

Rating	CVSS Score	Definition
Low	up to 3.9	This rating is given to all other issues that have a security impact. These are the types of vulnerabilities that are believed to require unlikely circumstances to be able to be exploited, or where a successful exploit would give minimal consequences.

4 When to prefer version upgrades over backports

It is a general policy rule that no new upstream versions of a package are introduced into our enterprise products. This rule is not an absolute rule however. For certain types of packages, in particular antivirus software, security concerns weigh heavier than the conservative approach that is preferable from the perspective of quality assurance. For packages in that class, occasionally newer versions are introduced to a released version of an enterprise product line.

Sometimes also for other types of packages the choice is made to introduce a new version rather than a backport. This is done when producing a backport is not economically feasible or when there is a very relevant technical reason to introduce the newer version.

5 Major security vulnerabilities in 2022

5.1 pwnkit

Overview

In the beginning of January, Qualys securcal root exploit in "pkexec" component of polkit. The pkexec application is a setuid tool designed to allow unprivileged users to run commands as privileged users according predefined policies. Local attackers could use the setuid root /usr/bin/pkexec binary to reliably escalate privileges to root. The previous version of pkexec did not handle the calling parameters count correctly and ends trying to execute environment variables as commands. An attacker could leverage this by crafting environment variables in such a way it would induce pkexec to execute arbitrary code. When successfully executed the attack could cause a local privilege escalation from unprivileged users able to execute pkexec to root.

This vulnerability affected all SUSE Linux Enterprise Server 12 and SUSE Linux Enterprise Server 15 service packs. It did not affect SUSE Linux Enterprise Server 11, as it used a previous generation called “PolicyKit”.

Solution

Installing the updated packages provided by SUSE is sufficient to fix the problem. Use

```
zypper lp -a --cve=CVE-2021-4034
```

to search for the specific patch information. A restart of the service is not required.

Note that for any SPx (Service Pack level) which is no longer in general support, an LTSS or ESPOS subscription may be needed to obtain the update. See the SUSE “CVE Page” link in the “References” paragraph below for more details about each SPx. In future releases, SUSE has split out `pkexec` from the default installed `polkit-1` packages, to allow reducing the attack surface if `pkexec` is not required on the system.

Workaround

It is also possible to remove the setuid bit from `/usr/bin/pkexec` with

```
chmod 755 /usr/bin/pkexec
```

or even by deleting `/usr/bin/pkexec` until fixed packages can be installed.

SUSE does not recommend removing the **`setuid`** bit as it will cause breakage on the system. Removing the **`setuid`** permission from the `pkexec` binary will prevent it from working properly for legitimate use cases. This means that any application which relies on `pkexec` execution will stop working, possibly causing unexpected system errors and behavior. The workaround prevents exploitation and might be the right thing to do given how easy the exploit it, but customers must be aware that this will break functionality until the update is installed.

References

- SUSE CVE Web page for CVE-2021-4034: https://www.suse.com/security/cve/CVE-2021-4034.html
- SUSE Technical Information Document (TID) 000020564: https://www.suse.com/support/kb/doc/?id=000020564

5.2 Samba vfs_fruit remote code execution

Overview

Researcher Orange Tsai from DEVCORE reported a remote buffer overflow in the “fruit” vfs module of Samba, tracked under CVE-2021-44142. The fruit module, which is used by Samba for Apple-related extended attribute storage, can be exploited by remote attackers with access to the Samba server to execute code as the Samba server (basically as root). The Samba `vfs_fruit` module uses extended file attributes (EA, xattr) to provide “[...] enhanced compatibility with Apple SMB clients and interoperability with a Netatalk 3 AFP file server.” Samba versions prior to 4.13.17, 4.14.12 and 4.15.5 with `vfs_fruit` configured allow out-of-bounds heap read and write via specially crafted extended file attributes. A remote attacker with write access to extended file attributes can execute arbitrary code with the privileges of smbd, typically root.

Solution

The “fruit” vfs module is not configured by default. To determine if it is used, check for a line starting with `"vfs objects="` with “fruit” listed in `/etc/samba/smb.conf`.

Packages containing a fix for this security issue were made available quickly. They should be installed using

```
zypper patch --cve=CVE-2021-44142
```

for applying the fixed packages.

References

- SUSE TID 000020564: https://www.suse.com/support/kb/doc/?id=000020564
- SUSE CVE-2021-4034 Web page: https://www.suse.com/security/cve/CVE-2021-4034.html

5.3 Remote Stack Overflow in the `tipc` networking module of the Linux kernel

Overview

The Transparent Inter Process Communication (TIPC) is an IPC mechanism designed for intra-cluster communication. It represents the cluster topology using the concept of nodes and the links between these nodes. A stack overflow flaw was found in the Linux kernel's TIPC protocol functionality in the way a user sends a packet with malicious content where the number of domain member nodes is higher than the 64 allowed. This flaw allows a remote user to crash the system or possibly escalate their privileges if they have access to the TIPC network.

As part of the monitoring framework the "Overlapping Ring Supervision Algorithm" was introduced to monitor neighbouring nodes in the cluster. This also introduced a security bug that could be exploited to execute a remote Denial of Service (DoS) attack on systems using TIPC and FORTIFY_SOURCE enabled.

Solution

Installing the updated packages provided by SUSE is sufficient to fix the problem. Use the following command to search for the specific patch:

```
zypper lp -a --cve=CVE-2022-0435
```

or

```
zypper patch --cve=CVE-2022-0435
```

to apply the fixes.

After a kernel update a restart is required. Unless you are using the SUSE Linux Enterprise Live Patching extension, for which SUSE provides the live patch to fix this vulnerability allowing you to patch the kernel without shutting down your system and reducing the need for planned downtime and increasing service availability.

References

- SUSE TID 000020580: https://www.suse.com/support/kb/doc/?id=000020580
- SUSE CVE-2022-0435 Web page: https://www.suse.com/security/cve/CVE-2022-0435.html

5.4 Dirty Pipe

Overview

On Monday, March 7th, security researcher Max Kellermann published a new software vulnerability that affect users of the Linux kernel. The vulnerability, called Dirty Pipe (CVE-2022-0847), impacts Linux kernels 5.8 and later, and allows local attackers to overwrite files even if they had only read permissions, allowing for easy privilege escalation. The issue is triggered by a combination of two bugs, one bug in Linux kernels 4.9 and newer and made exploitable by the second bug introduced in Linux kernel 5.8. Our currently maintained SUSE Linux Enterprise products are not affected as they ship older Linux kernels than 5.8. The upcoming SUSE Linux Enterprise 15 SP4 with Linux kernel 5.14 will be already fixed before shipment.

Solution

We have released fixes for the first bug for SUSE Linux Enterprise 12 SP4 and newer and SUSE Linux Enterprise 15 and newer, even though they are not directly affected. To install the respective patch, use

```
zypper patch --cve=CVE-2022-0847
```

References

- SUSE TID 000020603: https://www.suse.com/support/kb/doc/?id=000020603
- SUSE CVE-2022-0847 Web page: https://suse.com/security/cve/CVE-2022-0847.html
- SUSE blog article: https://www.suse.com/c/suse-statement-on-dirty-pipe-attack/
- Wep page of dirty pipe: https://dirtypipe.cm4all.com/

5.5 Boothole 3

Overview

GRUB developers and security researchers have identified more security relevant bugs in the GRUB2 and `shim` boot loaders, which could be used by local attackers to circumvent the secure boot chain. This vulnerability has similar effects and considerations as the original Boothole and Boothole2 issues. For regular users with their machine under full control this is less of an issue as in scenarios relying on secure boot, like public systems.

Solution

Boothole 3 consists of the following security vulnerabilities:

- CVE-2021-3695: A crafted PNG grayscale image may have led to out-of-bounds write in heap.
- CVE-2021-3696: A crafted PNG image may have led to out-of-bound write during huffman table handling.
- CVE-2021-3697: A crafted JPEG image could have led to buffer underflow write in the heap.

These security issues require attackers to supply crafted images to GRUB2, which is unlikely in common local scenarios, but can allow bypassing the secure boot chain.

- CVE-2022-28733: Fixed net/ip to do ip fragment maths safely. If GRUB2 is loading artefacts from the network, a Man-In-The-Middle attack could be used to execute code. This is an uncommon scenario.
- CVE-2022-28737: Fixed a buffer overflow in `shim`.
- CVE-2022-28734: Fixed net/http OOB write for split HTTP headers.
- CVE-2022-28735: GRUB2 verifier framework changes to avoid potential bypasses.
- CVE-2022-28736: Fixed a use-after-free in chainloader command.

SUSE is switching to a new secure boot signing key for secure boot signed artefacts. We have released GRUB2 updates, with incremented SBAT revision on AMD64/Intel 64 and signed with the new secure boot key to allow disabling it on IBM Z and IBM Power. SUSE also released

Linux kernel updates signed with the new signing key in June 2022 and following days on our regular “second Tuesday of the month” kernel release time. In 2022 we released a new `shim` version that disallows use of the previous secure boot keys and fixes a `shim` security issue, with incremented SBAT version after all the previous updates.

References

- SUSE TID 000020668: https://www.suse.com/support/kb/doc/?id=000020668

`grub2` security issues

- SUSE CVE-2022-28736 Web page: https://www.suse.com/security/cve/CVE-2022-28736
- SUSE CVE-2022-28735 Web page: https://www.suse.com/security/cve/CVE-2022-28735
- SUSE CVE-2022-28734 Web page: https://www.suse.com/security/cve/CVE-2022-28734
- SUSE CVE-2022-28733 Web page: https://www.suse.com/security/cve/CVE-2022-28733
- SUSE CVE-2021-3697 Web page: https://www.suse.com/security/cve/CVE-2021-3697
- SUSE CVE-2021-3696 Web page: https://www.suse.com/security/cve/CVE-2021-3696
- SUSE CVE-2021-3695 Web page: https://www.suse.com/security/cve/CVE-2021-3695

`shim` security issue

- SUSE CVE-2022-28737 Web page: https://www.suse.com/security/cve/CVE-2022-28737

5.6 Side-channel information leaks / denial of service attack against MMIO registers

Overview

Security researchers and Intel engineers have identified several transient execution side-channel information leak attacks and one denial of service attack when accessing MMIO registers.

Multiple flavors of these issues have been identified:

- CVE-2022-21166: Device Register Partial Write (DRPW) Some endpoint MMIO registers incorrectly handle writes that are smaller than the register size. Instead of aborting the write or only copying the correct subset of bytes (for example, 2 bytes for a 2-byte write), more bytes than specified by the write transaction may be written to the register. On some processors, this may expose stale data from the fill buffers of the core that created the write transaction. This issue is mitigated using CPU Microcode and Operating System (kernel) code changes.
- CVE-2022-21127: Update to Special Register Buffer Data Sampling The RDSAND, RDSEED, SGX EGET KEY instructions use the low bandwidth MMIO interface, and their content could be sampled using side-channel information leak methods. This issue is being mitigated with CPU Microcode updates.
- CVE-2022-21125: Shared Buffers Data Sampling (SBDS) After propagators may have moved data around the uncore and copied stale data into client core fill buffers, processors affected by MFBDS can leak data from the fill buffers. This issue is mitigated using CPU Microcode and Operating System (kernel) code changes.
- CVE-2022-21123: Shared Buffers Data Read (SBDR) It is similar to Shared Buffer Data Sampling (SBDS) except that the data is directly read into the architectural software-visible state. This issue is mitigated using CPU Microcode and Operating System (kernel) code changes.
- CVE-2022-21180: Undefined MMIO Hang While not directly related to side channel information leaks, overly long MMIO reads to short MMIO registers could lead to machine hangs, causing a denial of service. This will be fixed by filtering out too long MMIO reads in kernel / hypervisor software.

Solution

An updated Intel CPU Microcode was published in the Intel IPU 2022.1 release, released by SUSE in `ucode-intel` version 20220510 packages. SUSE released kernel updates to mitigate the leaks. A new kernel boot command line option has been introduced, called **`mmio_stale_data`**.

Configuration:

```
- mmio_stale_data=off
```

Mitigation is disabled.

```
- mmio_stale_data=full
```

Mitigation is enabled, but SMT is still enabled so information might leak on the same CPU core.

```
- mmio_stale_data=full,nosmt
```

Mitigation is enabled, and SMT is disabled so the mitigation is complete. Note that this is option is also covered by using the generic “mitigations” option.

References

- SUSE Technical Information Document (TID) 000020669:https://www.suse.com/support/kb/doc/?id=000020669
- SUSE CVE-2022-21166 Web page: https://www.suse.com/security/cve/CVE-2022-21166
- SUSE CVE-2022-21127 Web page: https://www.suse.com/security/cve/CVE-2022-21127
- SUSE CVE-2022-21123 Web page: https://www.suse.com/security/cve/CVE-2022-21123
- SUSE CVE-2022-21125 Web page: https://www.suse.com/security/cve/CVE-2022-21125
- SUSE CVE-2022-21180 Web page: https://www.suse.com/security/cve/CVE-2022-21180
- Article: https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/undefined-mmio-hang.html
- Article: https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/technical-documentation/processor-mmio-stale-data-vulnerabilities.html

5.7 RETBLEED transient execution information leak side-channel attack

Overview

Security researchers Johannes Wikner and Kaveh Razavi from ETH Zuerich have identified a new transient execution information leak caused by “return” CPU instructions on Intel and AMD x86 systems. In certain scenarios the return instructions turn to use the indirect branch predictor, which can be influenced by Branch Target Injection attacks aka Spectre-BTI. This allows local attackers to execute code to leak sensitive data out of the kernel, same as other Spectre like vulnerabilities.

Affected CPUs:

- Intel Skylake and newer Intel x86 CPU generations
- AMD Bulldozer (family 0x15) up to Zen 3
- Arm CPUs that are also vulnerable to Spectre variant 2

Solution

Various mitigation methods have been implemented and can be selected manually or are selected automatically. IBPB (Indirect Branch Prediction Barrier) can be used to flush the indirect branch predictor. This has performance impact, but is the safest option. On Intel Skylake the IBRS feature is used (Indirect Branch Restricted Speculation), which clears the branch history buffer from lower privileged entries. Go to the "References" paragraph for instructions on each mitigation type.

References

- SUSE Technical Information Document (TID) 000020693:https://www.suse.com/support/kb/doc/?id=000020693
- SUSE CVE-2022-29900 Web page: https://www.suse.com/security/cve/CVE-2022-29900
- SUSE CVE-2022-29901 Web page: https://www.suse.com/security/cve/CVE-2022-29901
- SUSE CVE-2022-28693 Web page: https://www.suse.com/security/cve/CVE-2022-28693
- SUSE CVE-2022-23825 Web page: https://www.suse.com/security/cve/CVE-2022-23825

5.8 `openssl` 3 certificate parsing buffer overflow

Overview

Security researcher "Polar Bear" has reported a buffer overflow in `openssl` when parsing X.509 certificates. This problem could be used by remote attackers to cause overflows in `openssl`-based servers parsing, for example client certificates, or client overflows when parsing server certificates.

This problem only affects `openssl` 3.x and newer, older `openssl` versions are not affected. `openssl` 3 is only shipped as a secondary library starting with SUSE Linux Enterprise 15 SP4. No package currently uses `openssl` 3.

Solution

Patches are available for download and install for SUSE Linux Enterprise 15 SP4.

References

- SUSE CVE-2022-3602 Web page: https://www.suse.com/security/cve/CVE-2022-3602.html
- SUSE CVE-2022-3786 Web page: https://www.suse.com/security/cve/CVE-2022-3786.html

6 Vulnerability Management in 2022

The SUSE Solution Security team constantly monitors all the software components used in our products for security issues. More and improved tools are now available for finding out zero-day vulnerabilities and scanning for existing vulnerabilities. Such instruments can validate and report back if the application code written is following the standard security best practices or there are major gaps in potential attack vectors such as buffer overflow, denial of service or unwanted elevated access. It is clear that developers are becoming more and more security-aware and the quality of the code being developed has greatly improved in both quantity and quality. While we notice an increasing number of important vulnerabilities, the number of critical vulnerabilities is going down year over year.

TABLE 2: VULNERABILITIES WITH A UNIQUE CVE IDENTIFIED, IMPACTING SUSE PRODUCTS IN 2022

Low	Moderate	Important	Critical
100	637	436	15

TABLE 3: SECURITY UPDATES AND PATCHES RELEASED TO FIX THESE VULNERABILITIES IN 2022

Low	Moderate	Important	Critical
58	584	2364	101

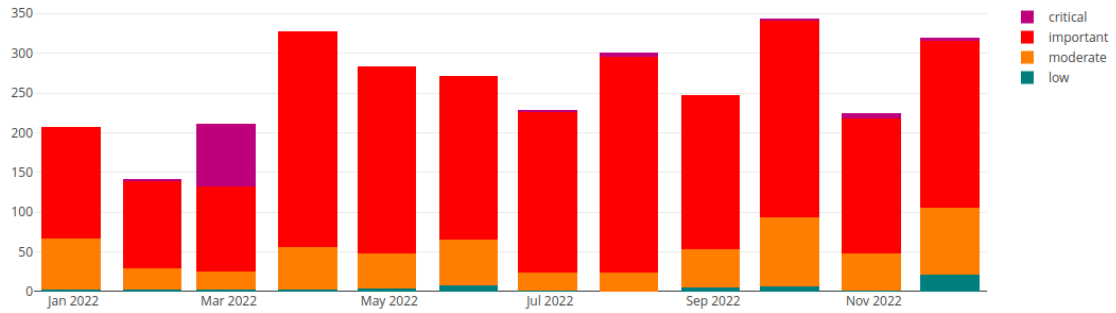

FIGURE 1: SECURITY UPDATES AND PATCHES RELEASED PER MONTH IN 2022

7 Secure Software Supply Chain

Securing our software supply chain is a top priority for SUSE to protect our customers security risks, known and zero-day vulnerabilities. Ensuring that no threat actor can inject malicious code into our build service systems is certified by industry leading security certifications. Our teams continually work to certify all SUSE products, and develop security solutions to offer our customers the highest level of trust and reliability.

A new industry standardization effort named Supply chain Levels for Software Artifacts (SLSA), started by Google and driven by several industry stakeholders, aims to protect the integrity of the software supply chain. SLSA defines four levels of assurance, going from basic requirements at level 1 to strict rules and documentation requirements at level 4. While the SLSA standard is still in development, SUSE already considers it as a great representation of needs for a secure product build environment. Thus we are adjusting our processes and tooling to meet the requirements of the highest assurance level 4. The main controls which are covering the highest assurance level 4 for our SUSE Linux Enterprise product offering include:

SOURCE CODE MANAGEMENT

Keeping source code integrity is the key aspect of supply chain integrity. Source code integrity needs to be defended against all threats originating from insider or outsider attacks.

- **Source threats:** Typical source code threats include bad code that introduces vulnerabilities or a compromised source control system. To address bad code injection, SLSA mandates two-person reviews. To prevent source code from getting compromised, SLSA mandates strong measures to secure the source control systems.
- **Build threats** Build threats include code commits to the build that were not tracked by the source control system, a compromised build platform, bypassing the CI/CD system, a compromised package repository, and injecting bad packages. Most build threats are mitigated by maintaining a controlled build environment, where each build is also fully encapsulated on its own, not influenceable from outside, or even reproducible. To prove this to the outside, detailed provenance data can be generated, which allows for the external inspection of the builds. Strong security controls ensure that the build platform is not easily compromised.
- **Dependency threats** Dependency threats come into play where risky dependencies are used. SLSA addresses this kind of threat by mandating provenance for all artifacts (files, Git commits, directories of files, container images, ...). That way one would have an indication that this dependency was not built from the proper builder or out of the designated GitHub repository.
- **Version-controlled source** Our Open Build Service complies with this requirement as it assigns numeric identifiers to commits. The commit stores information about the author, the commit time, a comment describing the commit, and other information. The commit also contains an identification of the source content, like the tree object in git.
- **Verified revision history:** Each commit stores information about the author and commit time. It also contains a comment describing the commit and other information. Identities of all actors are verifiable and use two-factor authentication.
- **Revision and change history are retained indefinitely:** We retain all sources, but also any shipped binary indefinitely. A retraction of a shipped update from the customer's channels does not lead to its removal from the build system. Furthermore, all source references and binary shipments including bug tracker references (CVE, bugzilla) are tracked by the build system for each shipment channel.
- **Two-person reviewed:** The workflows for creating a new product and delivering maintenance updates involve multiple parties, like the core code review team and the maintenance or product release managers as a minimum. Furthermore, there are reviews by subject matter experts and additional checks for quality assurance and legal aspects. These reviews are enforced by the OBS, and a single decline rejects the entire release process.

BUILDING AND BUILD SYSTEM

The next part of the integrity chain is the actual build process that turns sources to binaries. The entire build process must be secured against any kind of unknown or outside influence to avoid possible tampering with the builds. Builds must be reproducible to allow verification and checking of build results.

- **Scripted build:** This is achieved by the SUSE build script also used by the Open Build Service (OBS). Even the decision to invoke a build is made by OBS based on the submitted code changes or other builds.
- **Build service:** We are running our build service in a dedicated build cluster within the SUSE data center
- **Build as code:** The recipe files defining the build process are part of the sources of the individual packages, for example the RPM spec file, its sources and patches, or image and container description files. The build environment configuration (project config) is also under source control in OBS.
- **Isolated and Ephemeral environment** We are using an isolated KVM instance for each build. Access to the outside is not possible (no network), only the sources and binaries prepared by OBS can be used. OBS also decides which pieces of the build artifacts are used. This includes running Linux kernel in the VM.
- **Parameterless:** The build happens completely decoupled from any user interaction. Any parameter must be part of any source submission. No input is possible during the build, which is ensured by the KVM setup. The only output during the build is the build log.
- **Hermetic:** Our software builds in a KVM guest without network, and everything required for the build is injected before the build instance is brought up.
- **Reproducible:** Our OBS system tracks all used binaries for each build and can reproduce the build environment of any released binary. The binaries used are also referenced in in-toto provenance files and made available together with the sources starting from SUSE Linux Enterprise 15 SP4 builds. Older builds may have missing binaries because of the nature of the bootstrapping process. It is notable that the SUSE Linux Enterprise 15 code base is not enforcing binary identical reproducibility yet. Instead, builds are compared and known good differences are accepted (for example time stamps or build host name). This validation is done by the code in the build-compare package.

PROVENANCE

A key aspect of supply chain security is the ability to prove that a build has been completed / a package built according to all SLSA4-mandated requirements. This provenance is established by means of providing metadata that proves compliance to SLSA. The SLSA requirements for provenance include process requirements on provenance generation and consumption, and requirements on the contents of the provenance.

- **Source threats** Available
- **Source threats** Authenticated
- **Source threats** Service-generated
- **Source threats** Non-falsifiable
- **Source threats** Dependencies complete

SUSE is up to a great start with its effort to attain SLSA L4 compliance, as SLSA requirements partly overlap with the requirements of Common Criteria EAL4 + . This means that several SLSA criteria were met by SUSE's supply chain processes right from the start. The core part of this is our certified and proven build and integration process which uses the Open Build Service technology. Over the past few months, SUSE has been working on improving and tightening processes and technologies to be able to claim full SLSA L4 compliance.

For more information on our compliance with the Google SLSA Level 4 requirements, visit: https://documentation.suse.com/sbp/server-linux/html/SBP-SLSA4/index.html

8 Adopting Sigstore for our Supply Chain Security

Sigstore is a recent initiative to enhance signing and cryptographic verification of open source deliveries. SUSE has adopted additional "cosign" style signing for its published container images including Base Container Images (BCI) containers. SUSE also started uploading cryptographic signatures to the global "rekor" transparency log for its containers and product repositories in February 2022. SUSE Linux Enterprise Base Container Images (SLE BCI) offer a platform for creating SUSE Linux Enterprise Server-based custom container images and containerized applications that can be distributed freely. SLE BCIs feature the same predictable enterprise lifecycle as SUSE Linux Enterprise Server. The SLE_BCI 15 SP3 and SP4 repository (which is a subset of the SUSE Linux Enterprise repository) gives SLE BCIs access to 4000 packages available for the

AMD64/Intel 64, AArch64, PowerPC, and IBM Z architectures. The packages in the repository have undergone quality assurance and security audits by SUSE. The container images are FIPS-compliant when running on a host in FIPS mode. In addition to that, SUSE can provide official support for SLE BCIs through SUSE subscription plans.

Security

Each package in the *SLE_BCI* repository undergoes security audits, and SLE BCIs benefit from the same mechanism of dealing with CVEs as SUSE Linux Enterprise Server. All discovered and fixed vulnerabilities are announced via e-mail, the dedicated CVE pages, and as OVAL and CVRF data. To ensure a secure supply chain, all container images are signed with Notary v1, Podman's GPG signatures, and Sigstore Cosign. We have also implemented support for cosign signatures on a rekor server, so you can also check BCI container images against rekor, the immutable tamper resistant ledger. All cosign signatures of containers and generated update repositories (GPG) are uploaded to the Linux Foundation's rekor transparency log.

Stability

Since SLE BCIs are based on SUSE Linux Enterprise Server, they feature the same level of stability and quality assurance. Similar to SUSE Linux Enterprise Server, SLE BCIs receive maintenance updates that provide bug fixes, improvements, and security patches. Tooling and integration SLE BCIs are designed to provide drop-in replacements for popular container images available on hub.docker.com. You can use the general-purpose SLE BCIs and the tools they put at your disposal to create custom container images, while the language stack SLE BCIs provide a foundation and the required tooling for building containerized applications.

Redistribution

SUSE Linux Enterprise Base Container Images are covered by a permissive EULA that allows you to redistribute custom container images based on such a SLE BCI.

9 Securing Our Product Portfolio

January 25th 2022

The Defense Information Systems Agency (DISA) has released the Security Content Automation Protocol (SCAP) for the SUSE Linux Enterprise Server 15 Security Technical Implementation Guide (STIG). The SCAP is available for download at https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_SLES_15_V1R1_STIG_SCAP_1-2_Benchmark.zip. DISA also released the automated benchmark for the SUSE Linux Enterprise Server 15 Security Technical Implementation Guide (STIG). The benchmark is also available on the Cyber Exchange public site at https://public.cyber.mil/stigs/downloads.

April 19th 2022

DISA released the SUSE Rancher Manager Security Technical Implementation Guide (STIG). The STIG is also available on the Cyber Exchange public site at https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_RGS_MCM_V1R1_STIG.zip

June 13th 2022

SUSE has translated its SUSE Security Situation Advisory Guide (SUSE S2 UAG) v. 1.1 into English. This guide provides an overview of possible immediate actions and approaches that can be taken with SUSE customer products. It can be accessed via https://links.imagerelay.com/cdn/3404/ql/0fb22d6c1aa740bf829f863d5841981a/SUSE_Security_Situation_Advisory_Guide__SUSE_S2UAG.pdf

June 29th 2022

SUSE became a collaborator with the National Institute of Standards and Technology (NIST) on the Automation of the Cryptographic Module Validation Program (ACMVP). "The National Cybersecurity Center of Excellence (NCCoE), a part of NIST, is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity challenges. Through this collaboration, the NCCoE develops modular, easily adaptable example cybersecurity solutions using standards, best practices, and commercially available technology." SUSE looks forward to our joint activities with NIST and the other vendors in the program. You can read more at the NIST ACMVP Web site.

August 1st 2022

SUSE was awarded the Certificate of Software Quality Level 1 (Certificate #22-0353), also known as the Good Software (GS) Certification, by the Telecommunications Technology Association (TTA) of the Republic of Korea, for its SUSE Linux Enterprise Server 15. "The GS (Good Software) Certification certifies good quality software based on international standards, ISO/IEC 25023, 25051 and 25041 to improve the quality of software products and promote the spread of high quality products." Certificate can be viewed at https://sw.tta.or.kr/product/prod_gsce_view.jsp?num=7255&pa=2d56c77a3d1739509363eafd002e37bb

October 31st 2022

The Defense Information Systems Agency (DISA) released SUSE Rancher Kubernetes Engine 2 Security Technical Implementation Guide (STIG). This content is published as a resource to assist in the application of security guidance to systems. The STIG may be accessed at https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_RGS_RKE2_V1R1_STIG.zip

December 6th 2022

SUSE successfully implemented ISO 27001 and ISO 27701 in full scope and with all of the clauses and achieved certification of our Information Security Management System (ISMS) and the Privacy Information Management System (PIMS) to the respective standards, attesting to our commitment of secure innovation, with focus on privacy, rights and freedoms of individuals. In doing so, SUSE has obtained two certifications from NQA, the leading independent provider of environmental simulation testing, inspection and certification services. They span everything within SUSE and our entities, including all countries we operate in, subsidiaries and all processes.

The ISMS & PIMS of SUSE (as PII Controller and Processor) applies to all client facing services, internal services, processed information including personal data related to employees, clients, openSUSE Community and other interested parties, related IT and non-IT supporting infrastructure as detailed in the latest Statement of Applicability version 1.0.

For additional information, you can download the certifications accessing the below links: ISO 27001: https://www.suse.com/assets/GB235856-HO-ISMS.pdf ISO 27701: https://www.suse.com/assets/GB235856-HO-PIMS.pdf

10 About SUSE

SUSE is a global leader in innovative, reliable and enterprise-grade open source solutions, relied upon by more than 60% of the Fortune 500 to power their mission-critical workloads. We specialize in Enterprise Linux, Kubernetes Management, and Edge solutions, and collaborate with partners and communities to empower our customers to innovate everywhere – from the data center, to the cloud, to the edge and beyond. SUSE puts the “open” back in open source, giving customers the agility to tackle innovation challenges today and the freedom to evolve their strategy and solutions tomorrow. The company employs nearly 2000 people globally and is listed in the regulated market (Prime Standard) of the Frankfurt Stock Exchange. For more information, visit https://www.suse.com.

11 Forward-looking statements

Any statements in this document about future expectations, plans and prospects for the company, including statements containing the words “aims”, “targets”, “will”, “believes”, “anticipates”, “plans”“, expects”, and similar expressions, may constitute forward-looking statements and should be read with caution.

Actual results may differ materially from those indicated by such forward-looking statements as a result of various important factors, including competitive landscape, development of customer deals, reliance upon customer relationships, management of growth and acquisitions, the possibility of undetected software issues, the risks of impacts of the COVID-19 pandemic and economic downturns, pricing pressures and the viability of the Internet. In addition, any forward-looking statements included herein represent views as of the date of this document and these views could change. The Company does not have any obligation to update its forward-looking statements. These forward-looking statements are subject to change and should not be relied upon as representing the Company’s views as of any date other than the publication date of this document.

12 Legal notice

GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
 Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts". line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.