BENOCS

- On Ubuntu's Security -

Contents

OVERVIEW

Since its inception in 2004, Ubuntu has been built on a foundation of enterprise-grade, industry leading security practices. From the toolchain to the suite of packages used and from the update process to the industry standard certifications, Canonical never stops working to keep Ubuntu at the forefront of safety and reliability. The security team at Canonical is constantly working to review threats, fix vulnerabilities and upgrade security capabilities for releases to protect your systems and workloads in production.

AUTOMATIC SECURITY UPDATES

Starting with Ubuntu 16.04 LTS, `unattended-upgrades` is configured to automatically apply security updates daily.

In this configuration, unattended-upgrades updates the package list, downloads and installs available security upgrades once every 24 hours. Those actions are triggered by timer units at a set time but with a random delay: `apt-daily.timer` and `apt-daily-upgrade.timer`. These timers activate the correspondent services that run `/usr/lib/apt/apt.systemd.daily` script. However, it may happen that if the system is off at the time the timer unit elapses, the timer will be triggered immediately at the next startup.

As a result, this guarantees that any and all available security updates are always applied to the system within 24 hours.

UBUNTU’S PATCH STRATEGY

reference: https://ubuntu.com/blog/securing-open-source-through-cve-prioritisation

According to a recent study, 96% of applications in the enterprise market use open-source software. As the open-source landscape becomes more and more fragmented, the task to assess the impact of potential security vulnerabilities for an organisation can become overwhelming. Ubuntu is known as one of the most secure operating systems, but why? Ubuntu is a leader in security because, every day, the Ubuntu Security team is fixing and releasing updated software packages for known vulnerabilities. It is a continuous 24/7 effort. In fact, on average, the team is providing more than 3 updates each day and the most vital updates are prepared, tested and released within 24 hours. To achieve that result, Canonical designed a robust process to review, prioritize and fix the most crucial software vulnerabilities first. Software vulnerabilities are tracked as part of the Common Vulnerabilities and Exposures (CVE) system and almost all security updates published by the Ubuntu Security team (via Ubuntu Security Notices – USNs) are in response to a given public CVE.

Key take aways:

- The Ubuntu Security team is constantly fixing and releasing updated software packages for known vulnerabilities.
- The most vital updates are prepared, tested and released within 24 hours.
- Ubuntu uses a robust process to review, prioritize and fix the most crucial software vulnerabilities first.

THE ROBUST TRIAGE PROCESS

The Ubuntu Security team manages their own CVE database to track various CVEs against the software packages within the Ubuntu archive. As part of this process, each day the team triages the latest public vulnerabilities from various sources, including MITRE, NIST NVD and others. This triage process involves assessing every single new publicly announced CVE and determining which (if any) software packages in Ubuntu may be affected, collecting any information required for patching the package (including upstream patches) and noting any potential mitigations for the vulnerability. Once CVEs are triaged against the applicable software packages, they are assigned a priority, from the range of negligible, low, medium, high and critical. This priority is then used by the Ubuntu Security team to indicate which vulnerabilities should be addressed first.

EXTENDED CVE REVIEW

A common method for assessing the severity of CVEs is the Common Vulnerability Scoring System (CVSS). This is designed to provide a numerical value for the severity of the particular vulnerability and to allow these to be compared between vulnerabilities. The CVSS score for a given CVE is calculated using a number of inputs and whilst this allows various aspects of the vulnerability to be considered, it has a number of shortcomings. In particular, whilst CVSS was designed to assess the technical severity of a vulnerability, it is often misused instead as means of vulnerability prioritization or risk assessment. In particular, there are many aspects that are important to consider for a given vulnerability which is not captured by CVSS, including the likelihood that the given software package is installed or in use, whether the default configuration of a package may mitigate the vulnerability and whether a known exploit against the vulnerability exists.

CVE PRIORITIZATION

In contrast, the priority value assigned by the Ubuntu Security team is designed to capture these and other elements so that it can be used as an effective measure to prioritize security software updates taking into account every Ubuntu instance – including server, desktop, cloud and IoT. Vulnerabilities that affect the largest number of Ubuntu installations and which present the largest risk (by say being remotely exploitable without any user input, etc.) are prioritized critical or high. Those which affect only a small number of users and might require user-input or might only cause smaller effects such as a denial-of-service may be prioritized as a medium, low or negligible. This CVE prioritization is done on a case-by-case basis for each vulnerability and since a given vulnerability might apply to more than one package in the Ubuntu archive, this can be assigned further on a vulnerability-per-package basis as well. This ensures that those vulnerabilities which have the highest risk and impact and which are likely to affect the largest number of Ubuntu installations are fixed first, regardless of the given CVSS score, to ensure that the risk of exploitation by known software vulnerabilities is limited as much as possible.

OVERVIEW OF CVE PRIORITIES

reference: https://people.canonical.com/~ubuntu-security/priority.html

The priorities assigned to vulnerabilities in Ubuntu are for prioritizing the work of when CVEs will be fixed as opposed to just an assessment of severity, importance or risk. The priority is based on many factors including severity, importance, risk, install base, software configuration, active exploitation and other factors which may adjust the impact of certain vulnerabilities such as Ubuntu's proactive security features. Importantly, these priority levels are distinct from other published severity levels such as CVSS as used in the National Vulnerability Database).

Priority	Description
Not Vulnerable	Packages which do not exist in the archive, are not affected by the vulnerability or have a fix applied in the archive.
Pending	A fix has been applied and updated packages are awaiting arrival into the archive. For example, this might be used when wider testing is requested for the updated package.
Unkown	Open vulnerability where the priority is currently unknown and needs to be triaged.
Negligible	Open vulnerability that may be a problem but otherwise does not impose a security risk due to various factors. Examples include when the vulnerability is only theoretical in nature, requires a very special situation, has almost no install base or does no real damage. These typically will not receive security updates unless there is an easy fix and some other issue causes an update.
Low	Open vulnerability that is a problem but does very little damage or is otherwise hard to exploit due to small user base or other factors such as requiring specific environment, uncommon configuration, user assistance, etc. These tend to be included in security updates only when higher priority issues require an update or if many low priority issues have built up.
Medium	Open vulnerability that is a real problem and is exploitable for many users of the affected software. Examples include network daemon denial of service, cross-site scripting and gaining user privileges.
High	Open vulnerability that is a real problem and is exploitable for many users in the default configuration of the affected software. Examples include serious remote denial of service of the system, local root privilege escalations or local data theft.
Critical	Open vulnerability that is a world-burning problem and is exploitable for most Ubuntu users. Examples include remote root privilege escalations or remote data theft.

UBUNTU SECURITY NOTICES

Developers issue an Ubuntu Security Notice when a security issue is fixed in an official Ubuntu package. These notices are published on a searchable web-page (https://ubuntu.com/security/notices), as an Atom feed (https://ubuntu.com/security/notices/atom.xml), as a RSS feed (https://ubuntu.com/security/notices/rss.xml) as well as through a mailing list (https://lists.ubuntu.com/mailman/listinfo/ubuntu-security-announce)

UBUNTU OVAL DATA

reference: https://ubuntu.com/security/oval

Canonical's Security Team also produces Ubuntu OVAL files. These are an industry-standard, structured, machine-readable dataset for all supported Ubuntu releases that contain details of all known security vulnerabilities and fixes relevant to the Ubuntu release. OVAL files can be used to evaluate and manage security risks related to any existing Ubuntu components. They can also be used to audit a system to check whether the latest security fixes have been applied. It is based on the Open Vulnerability and Assessment Language (OVAL).

HOW UBUNTU USES OVAL

Ubuntu OVAL uses the OVAL vulnerability and patch definitions to enable auditing for Common Vulnerabilities and Exposures (CVEs) and to determine whether a particular patch, via an Ubuntu Security Notice (USN), is appropriate for the local system.

Ubuntu OVAL also allows for any third-party Security Content Automation Protocol (SCAP) compliant tools to accurately scan an Ubuntu system for vulnerabilities.

USING UBUNTU'S OVAL DATA

USING OPENSCAP

Download the compressed XML:

```
wget https://security-metadata.canonical.com/oval/com.ubuntu.$(lsb_release
-cs).usn.oval.xml.bz2
```

Uncompress the data:

```
bunzip2 com.ubuntu.$(lsb_release -cs).usn.oval.xml.bz2
```

Use OpenSCAP to evaluate the OVAL and generate an html report:

```
oscap oval eval --report report.html com.ubuntu.$(lsb_release -
cs).usn.oval.xml
```

The output is generated in the file report.html. The report contains a table with all known security vulnerabilities and fixes and reports on whether they have been patched.

The report can be opened using your browser:

```
xdg-open report.html
```

UBUNTU OVAL DATA PARAMETERS

Parameter	Description
CVE_ID	CVE number as reported by MITRE
USN	Corresponding Ubuntu Security Notice
Description	A short description of the security risk addressed
Severity	CVE or USN severity as defined by the Ubuntu Security team
Affected Platform	Affected Ubuntu release(s), incl ESM
Title	CVE number, affected Ubuntu release(s) and Severity
Public date	The date on which a CVE was publicly announced
Public date of USN	The date on which a USN was published
Reference	Links to more information about the issue
BugReport	Link to bugreport about the issue

Note: The above parameters are included in the OVAL xml file, but not all are shown in the resulting generated OpenSCAP report.

HOW UBUNTU OVAL DATA WORKS

As software vulnerabilities are discovered, they are assigned CVE identifiers by MITRE and other organizations. Canonical triages these CVEs to determine whether the vulnerabilities affect software distributed within Ubuntu. The results of this triage are then used to generate the CVE OVAL. The CVE OVAL can be used to assess the local system for vulnerabilities.

When the Ubuntu Security Team patches software to address one or more CVEs, an Ubuntu Security Notice (USN) is published announcing the update. The USN OVAL data is generated from information encapsulated within the USN and can be used to assess the system for missing patches.

HOW TO VALIDATE IF A GIVEN CVE HAS BEEN PATCHED

To audit the current patch level of a system, an OpenSCAP report can be produced using the latest OVAL file provided by Ubuntu. That report contains a table with all known security vulnerabilities and fixes and reports on whether they have been patched. See Section **Using OpenSCAP** for a detailed walk-through on how to do that.

ADVANTAGES OF USING UBUNTU PROVIDED PACKAGES OVER THIRD-PARTY ONES

There are several advantages when using Ubuntu provided packages as opposed to using third-party versions.

Before making a security update available, the update needs to be tested to see if it fixes the flaw and also doesn't introduce any regressions. The Ubuntu Security Team uses the QA Regression Testing suite when performing testing. QA Regression Testing has information on performing tests, checklists, scripts and various other information to help with testing. This ensures that the updated package continues to work in the same way as before: There are no changes in the behavior, all of the configuration files and binaries stay in their expected places and it is ensured that the existing configuration can still be used with the new version. In contrast, when using third-party packages, all of these tests, checks, audits and evaluations have to be done manually each time a new update becomes available.

Ubuntu has a well-defined process for evaluating, prioritizing, documenting and patching newly discovered vulnerabilities. If a similar process exists for third-party repositories, it needs to be evaluated on a case-by-case basis. Each additionally added repository opens a new attack vector. Instead of relying on the security of one repository, each and every used repository needs to be fully trusted. Compromising either of the repositories will put the system at risk. Essentially, the point here is that, the more third-party maintained software is installed, the more complex and volatile the system becomes. And it becomes harder to maintain.

The official Ubuntu packages can contain additional things that integrate into existing frameworks to improve the overall system security. One example of this is AppArmor profiles which limit the reach a given software has and thus the impact on the system security, should that software include vulnerabilities. The official Ubuntu packages are built on a trusted infrastructure from open source.

While they cannot be guaranteed to be secure, the provenance of the code and the build system ensure that they can be audited and known to work in a particular way. (reference: https://ubuntu.com/about/packages)

Many security features are available through the default compiler flags used by Canonical to build packages for Ubuntu. These flags provide stack and heap protectors, pointer obfuscation, address space layout randomization (ASLR), several compile-time and run-time protections in glibc and protection against several memory corruption attacks (reference: https://wiki.ubuntu.com/Security/Features#Userspace_Hardening). Whether these security features are also available when using third-party repositories, needs to evaluated for each and every one of them.

Ubuntu provided packages are separated into two different pockets (categories): "updates" and "security". "updates" includes things that have gone through Ubuntu's "StableReleaseUpdates" process (https://wiki.ubuntu.com/StableReleaseUpdates) and contain various important bug fixes. Anything built for "updates" is built on top of whichever version of a package is newest between "updates" and "security", so that nothing in "updates" will introduce security regressions. "security" includes only updated packages that contain security-related fixes and are built to not require anything from "updates". Anything built for "security" is built on top of which ever version of a package is newest between "updates" and "security", so that nothing in "security" will introduce bug regressions. (reference: https://wiki.ubuntu.com/SecurityTeam/FAQ)

Given these guarantees by the Ubuntu package maintainers that updates from the "security" contain fixes for all known vulnerabilities and neither introduce regressions nor change the behavior, `unattended-upgrades` can safely be enabled on a system to automatically apply all available security updates daily. This guarantees that a fix for each newly known vulnerability is applied to the system within at most 24 hours.

FURTHER READING: CANONICAL'S RESPONSE ON NCSC REQUIREMENTS

GCHQ's National Cyber Security Centre (NCSC) is the UK's lead technical authority on cyber security. Every service provider in the UK needs to follow the guidance from them and use only software that is compliant with these strict regulations. In the following whitepaper you can see how open source software supported by Canonical, including Ubuntu, OpenStack, Kubernetes, MAAS and others, complies with the detailed requirements and keeps your critical infrastructure safe.

The whitepaper can be downloaded from the following URL:

https://ubuntu.com/engage/canonical-response-on-NCSC-req