Canonical

A guide to infrastructure hardening

Best practices to improve the security posture of your Linux-based infrastructure deployments.

January 2023

Executive summary

Cloud technology and virtualisation enable developers and administrators to deploy infrastructure at a pace previously unheard of, which has brought them huge gains and enabled almost anyone to create internet services. However, this facility and flexibility does not mask the underlying need for deployments to be secure. The ever-present threats of ransomware attacks and data breaches make it imperative to lock down systems and prevent attackers from gaining an easy foothold.

A fully secure system is made up of many layers, from the hardware to the operating system and the application servers running your mission-critical code. Each deployment will be different but there are a number of universal themes to securing a system and hardening the infrastructure that your application is running on.

This paper describes the steps to take in order to harden your infrastructure, from the base operating system through to the application layer. It provides guidelines based on industry best practices, such as the Center for Internet Security (CIS) benchmarks, and will be especially useful if you are using Ubuntu or considering it for your next project.

What is hardening?

Hardening a system is synonymous with reducing its attack surface: remove unnecessary software packages, lock down default values to the tightest possible settings and configure the system to only run what you explicitly require. This is something that needs to be applied with judgement based on the risks to the systems in question. It's very important to harden your production environments and critical systems, including anything that is internet-facing, but may be less relevant to development machines for instance. It's up to you to decide how far to go with hardening your infrastructure, but we will provide some guidelines and best practices.

There are of course a myriad of technologies and vendors to choose from when building IT infrastructure, and no one guide can cover them all. In this piece we will discuss Linux systems, with a particular focus on Ubuntu as this is the most widely used platform for infrastructure deployments according to a 2022 OpenLogic report. Though the general approach should apply equally to almost any Linux environment.

Why would you want to harden your infrastructure and systems to start with? Isn't Linux secure enough already? Security is all about defence in depth. When it comes to protecting your most critical assets, it makes sense to take as many precautions as possible.

The operating system forms the foundation layer of an infrastructure deployment, and whatever steps are taken to secure an application are wasted if the underlying operating system doesn't provide a secure base to build on.

A default Ubuntu system comes configured with a sensible set of default options to balance security, usability and performance. In this piece we will give some recommendations as to how to tweak the installation to make it more secure; this will necessarily come with some usability trade-offs. We will then cover approaches to hardening other applications and systems that build on the base Ubuntu operating system.

Center for Internet Security (CIS) benchmarks

Regardless of the OS you are using, the best place to start when hardening systems is to look at CIS benchmarks for best practices. You will find guidelines to safeguard systems, software, and networks against today's evolving cyber threats. These guidelines are globally recognised and consensus driven by industry veterans and security professionals. CIS publishes benchmarks for a range of operating systems, cloud platforms, orchestration frameworks such as Kubernetes, and specific applications including web servers and databases.

CIS defines several different profiles for hardening an operating system, based on its intended use. They offer a Server and a Workstation profile, each of which can have 2 levels of hardening. Level 1 is designed to be practical and not inhibit the intended use of the system where possible; level 2 goes further, for situations where security is paramount, though might negatively impact the operation of the system. Judgement is required at all stages of infrastructure hardening.

CIS publishes Ubuntu benchmarks for the Long Term Support (LTS) releases, from 14.04 up until 22.04. These benchmarks are a very good starting point for hardening an Ubuntu system.

Automated hardening

Given how comprehensive and detailed the CIS benchmarks are, a number of software vendors offer tools to automate their implementation. Canonical, for instance, have worked with CIS to provide an automated way to apply the benchmark rules to an Ubuntu system, as well as to audit and check a system to see how many of the benchmark rules are being adhered to. This is known as the Ubuntu Security Guide (USG).

USG is an easy to use tool for compliance and auditing on Ubuntu operating systems. USG encodes the hundreds of rules from the CIS benchmarks, allowing you to apply the benchmark by re-configuring and tweaking the system in one simple operation. USG can also audit a system to generate a list of the benchmark rules which are not being adhered to. This output is available in both human and machine readable formats.

USG is available for 20.04 Focal Fossa LTS (with the latest 22.04 Jammy Jellyfish LTS version in development) with an Ubuntu Pro subscription.

Operating system hardening specifics

When it comes to hardening your operating system, a combination of manual configuration steps and automatic tooling will provide a strong baseline.

Pre-installation

The following section details design choices that need to be made before installing the Operating System. These are SecureBoot, Full Disk Encryption and Partitioning, which when combined enable a secure chain of trust when booting up, from the BIOS to the bootloader to the Linux kernel. Full Disk Encryption ensures that the files on disk have not been tampered with, and cannot be read by an offline attacker. Partitioning the disk into logical areas means that the system can continue to function if any of the world-writable locations become completely filled, which would otherwise result in resource exhaustion causing the system to halt.

SecureBoot

SecureBoot is a BIOS option to ensure that the operating system being loaded has been cryptographically signed by a known vendor, using Public Key Infrastructure. This makes it difficult for an attacker to modify the Operating System by inserting rootkits and other malicious software before the OS is loaded, as these changes would be detected by SecureBoot.

On Ubuntu, all pre-built binaries intended to be loaded as part of the boot process, with the exception of the initrd image, are signed by Canonical's UEFI certificate, which is implicitly trusted by being embedded in the shim loader, itself signed by Microsoft. SecureBoot should be enabled to ensure that the files on disk have not been tampered with.

Read more about SecureBoot on Ubuntu

Full Disk Encryption

Full Disk Encryption (FDE) prevents an attacker from either reading or writing files on the system if they do not have the credentials to unlock the disk encryption. The initial stage of the Operating System's boot process is loaded before the disk encryption password can be entered, and is therefore unencrypted, though this is protected by SecureBoot. The rest of the disk is fully encrypted so that it is not possible to extract any files or data from the disk, or modify the contents, when it is not powered on unless the password is known.

On Ubuntu, Full Disk Encryption is enabled via LVM (the Logical Volume Manager) and LUKS (Linux Unified Key Setup), which can be configured during the OS installation.

Partitions

Linux systems come with several world-writable and user-writable locations, as well as system-writable locations that are used for logs and temporary files. The system requires some temporary headroom on the filesystem to operate correctly, and if the filesystem becomes completely filled then the system ceases to be able to function correctly.

In order to prevent such resource exhaustion, world- and user-writable locations should be created on separate partitions, so that even if they become completely filled the system can still operate correctly.

A remote attacker could try and take a system offline by causing running services to generate excessive log files, or a local attacker could simply fill up the world-writable areas. To prevent this, the following filesystem locations should be mounted on separate partitions along with their recommended mounting options (which are relevant for any Linux filesystems):

```
/boot (rw)
/tmp (rw,nosuid,nodev,noexec)
/var/tmp (rw,nosuid,nodev,noexec)
/home (rw,nosuid,nodev)
/var/log (rw,nosuid,nodev,noexec)
/var/log/audit (rw,nosuid,nodev,noexec)
```

Hardening best practices

In order to minimise vectors of attack against infrastructure and systems, administrators can take precautions to identify and isolate the software and services that are installed on the system, and remove any pieces of software that are not required for the routine operation of the system. Unused software increases the attack surface of the system, and were at attacker to gain unauthorised access they could utilise software such as compilers and network analysis tools to gain further privilege and access to the infrastructure.

For any service, particularly one that is exposed to the network, such as a web server or database, it is recommended to limit the system capabilities afforded by default - this means that if an attacker compromises a service, they are restricted in their ability to further extend their reach into the rest of the infrastructure. On Ubuntu, this is facilitated by AppArmor - other Linux distributions may use SELinux, which fulfils the same purpose.

AppArmor

AppArmor is a Linux kernel security module which provides Mandatory Access Control using easy-to-configure text files. AppArmor limits the capabilities of processes to a known set of defaults in order to mitigate against unknown threats and application compromise. AppArmor can run in Complain mode, where it logs the actions taken by an application but allows it to run, and Enforce mode, where it blocks any action not allowed by the application's profile.

Ubuntu comes pre-installed with a range of AppArmor profiles for common applications. However, if your critical workload application doesn't have a profile, it is straightforward to create one. This is particularly important for any network-facing processes.

In order to generate a new profile, AppArmor can be put into Complain mode while the application is run with its full range of capabilities, capturing the actions taken into a file. This file can then be used as a new profile for Enforce mode.

Visit the AppArmor Community Help Wiki for more information

Remove unused software and services

This aspect of system hardening is very dependent on the specific functionality and workload of a system, and judgement is required to determine which packages and services should be installed. It is useful to consider broad categories of software here: graphical interface, X windows, printer subsystem, bluetooth, email server, web server, compilers and development tools, databases etc.

It's particularly important to audit the services which listen on a network interface, and to determine whether each service is totally necessary, as network services have a far greater exposure than just code running locally. On most Linux systems the **lsof** command will list which services are actively listening on the network:

```
sudo lsof -i -P -n | grep -v ESTABLISHED
```

Unencrypted protocols and services

Network services which don't have encryption are insecure and should be disabled. Administrators should inspect all running services and ensure that they are configured with encryption by default. If the service uses HTTP then it is possible to install a web proxy which serves HTTPS, such as nginx or apache, and ensure that the network traffic is encrypted even if the local service is not. Nmap can be used to check for services that might be unencrypted:

```
sudo nmap -v -sV -p- <address>
```

The output of nmap lists all the TCP network services that are listening on a host interface, and the "-sV" option tells nmap to interrogate the service and try to identify what it is. This will indicate whether or not the service is using an encrypted protocol or plain text.

Logging and forensics

Now that we have covered some of the general hardening best practices, we will cover some of the preventative security measures involved in day-to-day operations.

Infrastructure rarely exists in isolation, and when troubleshooting issues or tracking down suspicious log entries, it is very important to have a full picture of the activity on all the infrastructure components. One aspect of this is to archive logs to a central location where they can be monitored and analysed. Another aspect is to ensure that the system clocks are synchronised on all the various pieces of infrastructure so the log messages can be accurately correlated.

Intrusion Detection Systems (IDS) form a category of mechanisms to determine if a system has been compromised by an attacker. An IDS will monitor a piece of infrastructure, such as the contents of the filesystem, or running processes, and generate an alert if any activity falls outside the scope of what is expected. An IDS can either be host-based, running on the system itself, or network-based, running as a dedicated network appliance monitoring traffic.

Time servers

In order to keep log files in sync across machines within the enterprise, and aid forensic investigations, it's important to keep the system time updated with a Network Time Protocol (NTP) server. Time synchronisation is particularly important on Virtual Machines and in cloud environments without physical hardware clocks.

One point to note is that time synchronisation is more accurate with lower latency, so it matters how close the NTP server is located. Check that at least one NTP server is physically close to the system, ideally in the same country.

Ubuntu supports systemd-timesyncd, chrony, or ntp for time synchronisation, and any of these packages will work well, though only one should be active to prevent them fighting each other and overcorrecting the clock.

Learn about time synchronisation in Ubuntu
Learn about Ubuntu time management

Logging servers

Having a centralised logging server is a huge benefit for security, as attackers frequently clean up the log files of machines that they have compromised in order to hide their tracks. Along with synchronised system clocks, a central log repository enables a complete and accurate record of activity across the fleet.

On Ubuntu, either **rsyslog** or **journald** (using the **systemd-journal-remote** package) can be configured to send all locally generated logs to a remote server, and should be set up with a Transport Layer Security (TLS) connection to the log server.

The Ubuntu Security Guide can ensure that the log file size, permissions and rotation are configured.

AIDE

The Advanced Intrusion Detection Environment (AIDE) is a File Integrity Monitoring framework that checks for unexpected modification of files and directories, an essential part of Intrusion Detection and an important component in a hardened and secure system. AIDE scans configured areas of the filesystem on an automated schedule, generating a database of the file signatures it encounters. If it detects any additions or modifications compared to the previous cycle then it generates an alert.

There are many options available for File Integrity Monitoring, and Intrusion Detection generally - Ubuntu provides the **aide** package.

AIDE requires some configuration to set up the correct areas of the filesystem to scan and to customise its runtime and alerting. By default, AIDE will run once per day, though it is worth considering increasing this frequency. The most important aspect to consider, however, is triaging and handling the alerts. An Intrusion Detection System is only useful if suspicious changes are acted upon promptly, and are not lost in a queue of unhandled alerts.

Learn about AIDE file integrity monitoring

Patching and vulnerability management

In a report published by Verizon 2022, only 25% of the scanned organisations were found to patch known vulnerabilities within two months of their public disclosure, but at the same time, exploiting known vulnerabilities is one of the most common causes of organisations being compromised in ransomware attacks and data breaches. Whilst nobody can prevent zero-day attacks (where hackers take advantage of previously unknown software flaws), patching known vulnerabilities should be a fundamental part of every organisation's security strategy. It is good practice to scan systems regularly to check for known vulnerabilities, and at the same time also check for weak configuration settings. Triaging the results of these scans forms another part of the strategy.

Ubuntu and unattended upgrades

Ubuntu has the ability to automatically apply security patches for vulnerabilities through the **unattended-upgrades** package, which checks for and applies security updates daily. This mechanism works for almost all packages within the Ubuntu ecosystem, except for the core system libraries and the Linux kernel.

Learn about Automatic Security Updates Wiki

Ubuntu Pro

Canonical publishes Long Term Support (LTS) releases of Ubuntu every 2 years, in April. The main packages that make up these releases are supported by Canonical for 5 years, which means that Critical, High, and selected Medium severity security vulnerabilities will be promptly patched. For additional packages outside the main operating system, which are hosted in the ***universe*** repository, only best-effort security patching is offered.

Ubuntu Pro provides 10 year security maintenance for both ***main*** and ***universe*** packages, as well as access to the Ubuntu Security Guide, FIPS-140 compliant cryptographic modules, and kernel Livepatch.

Learn more about Ubuntu Pro

Livepatch

In order to patch the Linux kernel without rebooting and causing downtime, Canonical offers Livepatch as part of an Ubuntu Pro subscription. Livepatch removes the need for unplanned maintenance windows for high and critical severity kernel vulnerabilities by patching the Linux kernel while the system runs.

Learn about Ubuntu Livepatch Service

Vulnerability scanning

A 3rd party vulnerability scanning tool that has an up-to-date threat intelligence feed should be deployed to monitor all systems within the organisation. Scanning tools check for publicly known vulnerabilities as well as weak, insecure or missing configurations in a variety of software applications and services. It is useful to perform scans both internally within the network perimeter of each network segment, and externally from an internet-facing perspective: each section has its own risk profile that will inform the patching strategy and mitigation process.

Hardened configuration for services

Building on top of the base operating system layer, setting the application and service configuration to be secure by default is a key part of infrastructure hardening. The first place to start is by locking down the firewall to ensure that incoming network traffic is blocked unless explicitly required by the workload running on the system. These workloads, such as web servers or Secure Shell (SSH) servers, should then be configured to serve only encrypted connections and present trusted digital certificates.

Firewall

Network firewalls are one of the most widely deployed security devices, and the range and complexity of their configuration is out of the scope of this document. The basic principles apply across any firewall though. For a hardened system, the firewall should be set up to block any incoming traffic apart from what is explicitly required for the workload running on the system.

Ubuntu supports several different software firewalls: **iptables**, **ufw**, and the newer **nftables**.

To test the firewall configuration, it is instructive to scan the system from a test machine on the local network using a tool such as **nmap**. TCP, or stateful, services will respond to a handshake request if they are listening on the network interface and not blocked by the firewall:

```
sudo nmap -v -p- <address>
```

UDP, or stateless, services are more difficult to detect generically as UDP does not have the same handshake mechanism, so this method is less reliable:

```
sudo nmap -v -sU -p- <address>
```

Learn about UFW firewall
Learn about the new NFTables firewall
Read about the nmap scanning tool

Web server configuration

All web servers should be configured to support only known-secure ciphers and protocols. SSL and TLS versions 1.0 and 1.1 should be disabled. TLS v1.2 and v1.3 are currently understood to be secure, using strong cipher suites. Key exchange algorithms should provide perfect forward secrecy to prevent traffic from being decrypted if long-term secrets are compromised; this means that RSA should not be used for establishing session keys.

Mozilla has produced a web server configuration generator that sets up strong TLS ciphers and options for a wide variety of common servers, which is an excellent starting point:

Generate secure TLS configuration

In addition, Mozilla publishes a comprehensive guide to web server hardening, covering all the main security features built into modern browsers:

Learn about web application security features

Web server configuration for public-facing servers can be tested using SSL Labs scanner:

Test the security posture of web servers

For internal and non-public servers, configuration can be checked using **sslscan**.

TLS certificate management

All TLS servers require certificates, and there are organisations such as Let's Encrypt that provide certificates for free. This also applies to any service that uses TLS, such as databases and email servers.

Read about Let's Encrypt

These certificates are short-lived, and so require automation to keep them refreshed. Certbot is one such tool:

Learn about Certbot

All certificates that are published by a public Certificate Authority such as Let's Encrypt are recorded in a Certificate Transparency (CT) log, and it is useful to check the CT log for certificates issued to your domains to ensure that unexpected certificates haven't been issued by a malicious actor, or subscribe to an alert service to be notified whenever certificates are generated and entered into the log.

Search the Certificate Transparency log

SSH Server

The CIS profiles provide hardened configurations for SSH servers. There are several aspects that require further manual configuration however. User accounts should be managed centrally to ensure timely account deactivation, logging and auditing. SSH keys can also be loaded onto hardware tokens in order to prevent the risk of machine compromise leading to stolen private keys.

By default, a default installation of an SSH server will generate a new host key identifier, and clients trust the key upon first use. This operational model is not effective for Infrastructure-as-Code deployments where server resources are more ephemeral. Server certificates and identifiers should be utilised, so that clients can trust any server that has been provisioned with the correct credentials regardless of its lifecycle.

Operational hardening

Hardening infrastructure doesn't stop after the deployment and configuration phase: it is necessary to manage and maintain the systems throughout their operational lifecycle. This includes user account management, asset management, configuration management and secrets management. These might sound uninteresting compared to the initial infrastructure setup phase, but attackers commonly take advantage of lapses in operational security in cases such as test accounts, leaked credentials, or shadow infrastructure that is not accounted for by the system administrators.

User account management

Administering user accounts is one of the most important aspects of operational security, as compromised credentials and accounts are the most prevalent means for threat actors to gain access to enterprise infrastructure. User accounts should be centrally managed so that they can be audited in the same manner as other enterprise assets, which can be accomplished using a Single Sign-On (SSO) solution, giving each user a unique set of credentials. Logging and tracking of account activity needs to be implemented, and unused accounts disabled.

Individuals are encouraged to use a password manager, with unique passwords for each system to sign in to. Passwords should be long, at least 14 characters, and be used in conjunction with Multi-Factor Authentication (MFA) devices.

NIST provides guidelines for digital identity and authentication which are an excellent foundation for building out a user account policy.

Asset management

Knowing what assets are part of the enterprise estate is crucial to being able to defend that estate from attack. This is especially important in hybrid cloud and Infrastructure-as-Code deployments. An asset management system, such as Landscape, can track assets, audit systems for configuration and security updates, and monitor for health and performance.

Learn about Landscape

Config management

Configuration files for all assets within the estate should be stored in a version control system, such as git. Enterprises need to develop change control processes that allow for configuration changes to be applied in a timely manner according to the class of asset. This ensures that any changes made by a threat actor can be detected, and won't disrupt the operation of critical systems.

Secrets management

Storing secrets, credentials and private key material for configuration management is a challenge. Secrets stored on disk, or in environment variables, are susceptible to being exfiltrated by an attacker. Secrets should be stored in an encrypted vault storage system, with read access limited to the deployment framework.

High Availability setup

When it comes to patching software, having services and systems deployed in a High Availability (HA) means that components can be updated and refreshed without interrupting the service operation. HA also mitigates against the risk of hardware failure, and allows components of the system to be isolated in the case of an attacker compromise. Going into further details of HA setups is beyond the scope of this guide, but modern orchestration frameworks such as kubernetes allow for flexible scalability of your workloads.

Hardening containerised infrastructure

Modern software deployments that take advantage of HA typically run on virtualisation platforms using containers, and while these setups offer a plethora of security features and options, the basic principles of hardening still apply.

Containers

Container images provide security, stability and ease of use. However, container images are made of software from many different projects and sources, and an important aspect of selecting and maintaining containers is to choose those with a strong security maintenance commitment and history of security updates; software running in containers needs to be patched at the same time as non-containerised systems.

Container developers should remove all unneeded software components from the containers, and ensure that network-facing services use encrypted connections with trusted TLS certificates.

Learn more about the security of LTS docker images

Kubernetes

Kubernetes is a hugely complex framework that requires careful configuration to ensure a secure and hardened deployment. Whilst the technical details are specific to kubernetes, the general principles outlined in previous sections still apply, particularly around user management and web server configuration. Users interacting with the kubernetes API should be authorised and accounted for, and the API servers configured for secure TLS connections with trusted certificates.

The specific details for hardening kubernetes are outside the scope of this document, but there are resources available for in-depth hardening procedures, such as the guidance published by the NSA.

For a easy-to-use hardening, there are also tools such as kube-bench from Aqua Security which automate the security checks published in the CIS Kubernetes Benchmark.

Independent validation

The final piece of the puzzle, once the infrastructure is completed but ideally before it has gone live in production, is to have your work reviewed by an independent third party. The industry standard assessment methodology is to commission a penetration test, which will give you peace of mind that the infrastructure is built and configured securely as well as giving confidence to your customers that they can trust your services with their data.

Independent penetration test

A good penetration tester will be aware of the latest industry tools, techniques and best practices, and can provide a comprehensive and impartial assessment of the state of the deployment. The independence of the test is important, as it will be unbiased and free from any concerns that might have influenced the design and implementation of the system setup.

Choose a penetration testing company that is registered with an accredited industry standards body, such as CREST.

Conclusions

This hardening guide outlines the steps to harden Linux and Ubuntu-based infrastructure, and provides links to further resources for more specific and in-depth procedures. Using an automated hardening tool, such as the Ubuntu Security Guide, along with a set of rules provided by the Center for Internet Security, produces a good starting point for a secure operating system deployment.

Any applications or workloads deployed on top of the base operating system also need hardening. Following the principles outlined above, including secure configuration, removing unneeded components, minimising accounts and privileges, maintaining sound operational practices, and getting a third party to validate the deployment, can help you deliver a secure and hardened system.

Learn more

If you would like to know more about the Canonical approach to hardening, contact us. We offer a security proposition that encompasses the complete modern infrastructure stack, from bare metal provisioning through to kubernetes in public and private clouds.

Ubuntu