

Guidance

End User Devices Guidance: Ubuntu 14.04 LTS

Published

Contents

1. Usage scenario
2. Summary of platform security
3. Significant risks
4. How the platform can best satisfy the security recommendations
5. Network architecture
6. Deployment process
7. Provisioning steps
8. Policy recommendations
9. Enterprise considerations

This guidance is applicable to any device running Ubuntu 14.04 LTS. This version of Ubuntu is the latest Long Term Support (LTS) version available at the time of writing and is due to be supported by Canonical until 2019.

1. Usage scenario

Ubuntu devices will be used remotely over any network bearer, including Ethernet, Wi-Fi and 3G to connect back to the enterprise over a VPN. This enables a variety of remote working approaches such as:

- accessing OFFICIAL email
- creating, editing, reviewing and commenting on OFFICIAL documents
- accessing the OFFICIAL intranet resources, the internet and other web-resources

To support these scenarios, the following architectural choices are recommended:

- All data should be routed over a secure enterprise VPN to ensure the confidentiality and integrity of the traffic, and to benefit from enterprise protective monitoring solutions.
- Users should not be allowed to install arbitrary applications on the device. Applications should be authorised by an administrator and deployed via a trusted mechanism.


2. Summary of platform security

This platform has been assessed against each of the 12 security recommendations, and that assessment is shown in the table below. Explanatory text indicates that there is something related to that recommendation that the risk owners should be aware of. Rows marked [!] represent a more significant risk. See [How the platform can best satisfy the security recommendations](#) for more details about how each of the security recommendations is met.

Recommendation	Rationale
1. Assured data-in-transit protection	The built-in VPN has not been independently assured to Foundation Grade.
2. Assured data-at-rest protection	LUKS and dm-crypt have not been independently assured to Foundation Grade.
3. Authentication	
4. Secure boot	Secure boot is not fully supported on this platform.
5. Platform integrity and application sandboxing	
6. Application whitelisting	
7. Malicious code detection and prevention	
8. Security policy enforcement	
9. External interface protection	
10. Device update policy	
11. Event collection for enterprise analysis	
12. Incident response	

3. Significant risks

The following significant risks have been identified:

- The VPN has not been independently assured to Foundation Grade. Without assurance in the VPN there is a risk that data transiting from the device could be compromised.
- Users may choose to ignore certificate warnings leaving data in transit vulnerable to interception and alteration.
- The LUKS / dm-crypt disk encryption solutions have not been independently assured to Foundation Grade, and do not support some of the [mandatory requirements expected from assured full disk encryption products](#) . Without assurance there is a risk that data stored on the device could be compromised. However, the tpm-luks project can enable usage of Trusted Platform Modules (TPMs) by LUKS which may help meet more of these requirements.

- Ubuntu does not use any dedicated hardware to protect its disk encryption keys. If an attacker can get physical access to the device, they can perform an offline brute-force attack to recover the encryption password.
- Encryption keys protecting sensitive data remain available to an attacker when the device is locked. This means that if the device is attacked while powered on and locked, keys and data on the device may be compromised without the attacker knowing the password.
- Whilst not specific to Ubuntu, many devices which can run Ubuntu have external interfaces which permit Direct Memory Access (DMA) from connected peripherals. This presents an opportunity for a local attacker to exfiltrate keys and data.

4. How the platform can best satisfy the security recommendations

This section details the platform security mechanisms which best address each of the security recommendations.

4.1 Assured data-in-transit protection

Use the StrongSwan IPsec VPN client until a Foundation Grade VPN client for this platform becomes available.

4.2 Assured data-at-rest protection

Use LUKS/dm-crypt to provide full volume encryption. CESG recommend the use of a complex password of at least 9 characters in length, or of at least 6 characters in length when used in conjunction with a second factor.

4.3 Authentication

The user has a different authentication password to authenticate themselves to the device once they have entered the decryption password.

Alternatively, the user can implicitly authenticate to the device by decrypting the disk at boot time with their LUKS/dm-crypt password. This password unlocks a key which encrypts certificates and other credentials, giving access to enterprise services.

4.4 Secure boot

At present, Ubuntu cannot be configured to fully boot securely. Minor security benefit can be obtained by applying the configuration given in the [Policy recommendations](#) section below, but this will not fully satisfy the secure boot recommendation.

4.5 Platform integrity and application sandboxing

These requirements are met implicitly by the platform. Where available, AppArmor profiles limit applications' access to the platform. Other applications can be configured to use AppArmor if required.

4.6 Application whitelisting

Permissions can be configured at install time to ensure users cannot execute applications from any disk partition that they can write to. All application installation should be performed by an administrator.

4.7 Malicious code detection and prevention

The platform implicitly provides some protection against malicious code being able to run when configured as recommended.

Several third-party anti-malware products exist which attempt to detect malicious code for this platform. Content-based attacks can be filtered by scanning capabilities in the enterprise.

4.8 Security policy enforcement

The enforcement of security policies will be conducted by various operating system components and third-party products, based upon configuration files contained in specific directories. These include Policy Kit rules and LightDM settings, DConf settings, PackageKit rules, gksu settings and gksudo settings.

These configuration changes can be managed centrally through the use of Ubuntu packages, which can be deployed from the Software Configuration Management server.

Settings applied by the administrator cannot be modified by the user.

4.9 External interface protection

Interfaces can be configured using standard platform configuration files.

DMA is possible from some external interfaces including FireWire and Thunderbolt. As this platform does not control access via DMA it is advisable to procure hardware which does not have external DMA interfaces present if possible.

4.10 Device update policy

Operating system security updates can be configured to be automatically applied. Using the recommended automatic setting, application updates are installed automatically when the device is

switched on and fully booted. Kernel updates are applied when the user restarts their Ubuntu device.

4.11 Event collection for enterprise analysis

By default the majority of applications on Ubuntu will use RSyslog to output event logs. RSyslog can forward logs to a remote server which is most reliable when using TCP or the RELP protocol and can be secured with TLS encryption. RSyslog can also be configured to queue log messages when the remote server is unavailable and flush this queue to disk when the system is shutdown, which can avoid messages being discarded when connectivity to the central log server is unavailable.

Additional auditing can also be performed with auditd for specific events of interest to an administrator.

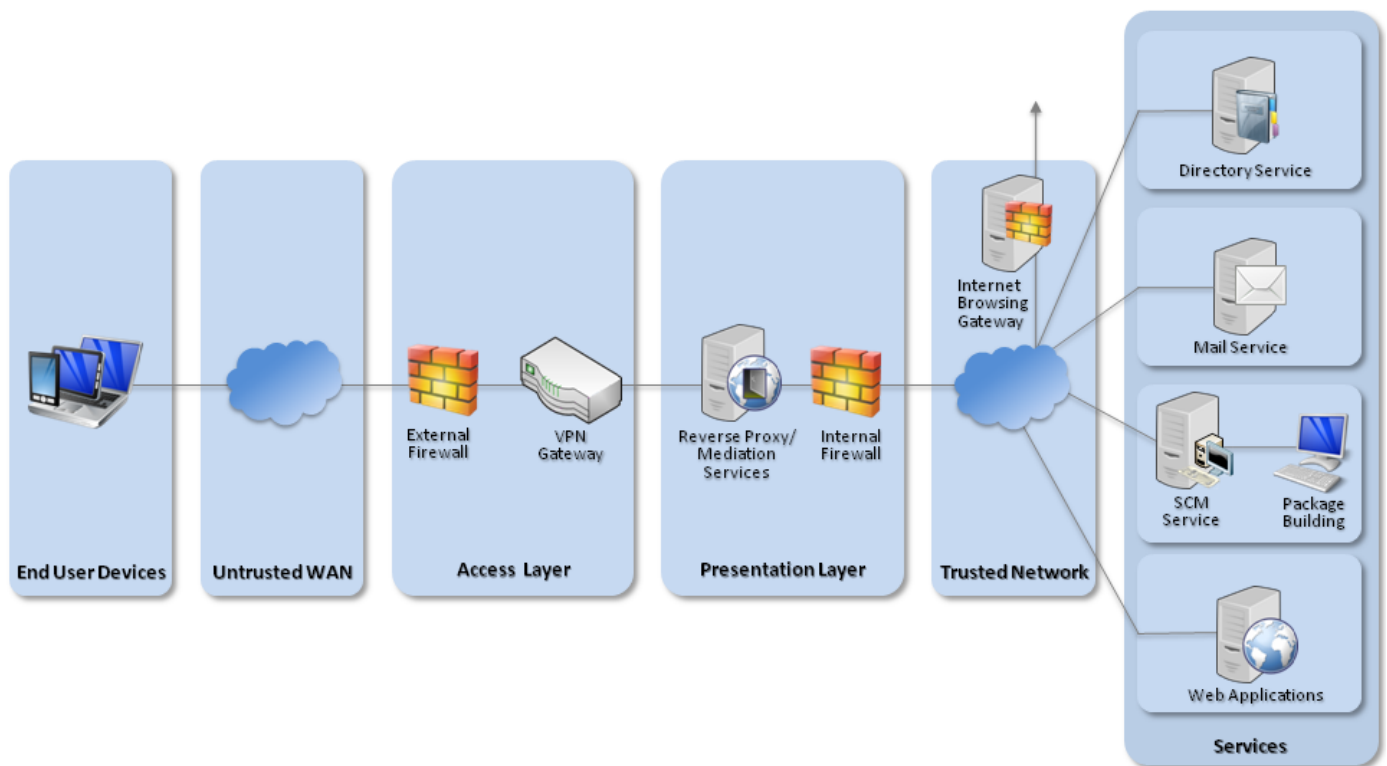
4.12 Incident response

There is no native remote wipe functionality available for Ubuntu, but remote wipe functionality can be implemented with a configuration management system such as Puppet. This system could also destroy key material for encrypted hard drives or use a secure erase feature of the drive if present.

Access to the enterprise network can be prevented by revoking the VPN client certificate associated with a lost or stolen device. Additionally, the client certificates for any other enterprise servers (such as email) that are stored on the device should be revoked.

5. Network architecture

CESG recommend that for remote or mobile working scenarios, organisations use a Software Configuration Management (SCM) service as part of a typical remote access architecture based on the Walled Garden Architectural Pattern as shown below.



Recommended network architecture for Ubuntu 14.04 LTS deployments

6. Deployment process

It is possible to deploy Ubuntu with the recommended configuration either via a Software Configuration Management service or to deploy this configuration with the install and post-install scripts provided with this guidance. An SCM service can help manage a large number of machines with differing requirements from a central location, although deploying simply with a script at install time can be simpler for smaller scale deployments.

If using a Software Configuration Management (SCM) service, the following steps should be taken:

1. Procure and provision an SCM server, such as Puppet or Chef. Optionally install Landscape as the system management tool.
2. Produce and provision an Ubuntu repository mirror and custom repository. This can be installed on the same host as the Software Configuration Management server.
3. Install and configure Ubuntu 14.04.1 LTS x86_64 in accordance with the requirements on a dedicated system for the purpose of building configuration packages.
4. Create signed packages to push the security configuration settings, and upload them to the custom repository. Update the list of packages in the Software Configuration Manager by re-synchronising with the repositories if required.

7. Provisioning steps

1. With the device configured to use UEFI mode, with no support for Legacy booting, and Secure Boot enabled, the device should be booted to the latest x86_64 Ubuntu 14.04 LTS Desktop Live CD.
2. At the GRUB menu, the “Try Ubuntu without installing” option should be selected.
3. Once the desktop has loaded, installation can proceed with the provided installation script (which can be copied from a USB stick). This script will:
 - clear any existing partitions from the disk and create a GUID Partition Table (GPT) with a 512MB EFI partition at the start of the disk followed by a 512MB Linux Native boot partition and the remaining free space as an encrypted Linux Logical Volume Manager (LVM) partition.
 - format the LVM partition with Linux Unified Key Setup (LUKS) to enable disk encryption with the specified password and open it ready for use.
 - initialise and create a Logical Volume Group in the LUKS encrypted volume.
 - within the group, create volumes for a swap, /tmp, /home and root (/).
 - launch the Ubiquity installer, in which the volumes should be selected and associated with the proper mount points and set to be used as EXT4 partitions, with the exception of swap which should simply be specified as used for swap space. The boot partition should also be selected to mount as /boot and also be EXT4. After the installer, “Continue Testing” should be selected to allow the script to finish its tasks.
 - create a valid /etc/crypttab in the newly installed system (achieved by mounting the relevant partitions).
 - with the new system mounted and in a chroot environment execute `# update-initramfs -u -k all` to allow LUKS to operate correctly at boot.
4. If opting not to use an SCM service, the script will apply as much of the recommended configuration as possible at install time. The final steps can be applied with the post-install script. If using an SCM, the new system can be added to it now and the relevant configuration packages applied to it.
5. Create a VPN certificate for the device, along with a certificate for the intended user and copy both of these to the device over a secure network connection.
6. Configure StrongSwan to connect to the VPN gateway using the relevant profile and certificate files, and then restart the StrongSwan daemon. Example StrongSwan configuration files are provided with this guidance but these will usually need to be customised for the specific environment.

8. Policy recommendations

This section details important security policy settings which are recommended for an Ubuntu deployment. Other settings (eg server address) should be chosen according to the relevant network configuration.

8.1 Secure boot

Whilst Secure Boot hardware can verify the first step of the boot chain, Ubuntu does not continue verifying the booting system so enabling and configuring Secure Boot offers no additional security benefit.

8.2 Automatic updates

Automatic updates should be enabled by executing `# dpkg-reconfigure unattended-upgrades` and selecting “Yes”. Alternatively, the `/etc/apt/apt.conf.d/20auto-upgrades` file can be created with the following content:

```
APT::Periodic::Update-Package-Lists "1";  
APT::Periodic::Unattended-Upgrade "1";  
APT::Periodic::AutocleanInterval "7";
```

The default configuration will only permit security related updates to be automatically installed. This can be changed by editing the `/etc/apt/apt.conf.d/50unattended-upgrades` file to enable additional sources for automatic updates.

To enable additional non-security automatic updates, uncomment the line `"${distro_id}:${distro_codename}-updates";`. The `Unattended-Upgrade::Allowed-Origins` should then contain the `-security` and `-updates` origins. Note that `-security` updates are also added to the `-updates` archive shortly after release in `-security`.

Other configuration settings are available and are detailed in `/etc/cron.daily/apt` and `/etc/apt/apt.conf.d/50unattended-upgrades`.

8.3 Software restriction

A separate partition should be created for `/home` and this, along with `/run/shm` and a `tmpfs /tmp` directory, should be configured in `/etc/fstab` to not allow execution of any files they contain. Users should only be able to create and edit files on these partitions. For example, this could be achieved by adding `/home noexec,nosuid,nodev 0 0,none /tmp tmpfs noexec,nosuid,nodev 0 0` and `none /run/shm tmpfs rw,noexec,nosuid,nodev 0 0` to `/etc/fstab`

Any additional locations on the file system that the non-administrator user can both write to and execute from should be identified and locked down by changing group membership or directory permissions where possible.

Known problematic locations on a 14.04.1 LTS install configured as per this guidance are `/var/crash`, `/var/metrics`, `/var/tmp` and `/home/lightdm-data/<username>`. All but the latter can be addressed by removing “other” write permission. The `lightdm-data` subdirectory however is manipulated by a root level LightDM process to always be writable and executable by the user, so

instead create a replica `lightdm-data` directory on the `/home` partition (which is mounted with `noexec`) and create a symlink to a user subdirectory there as in the example below:

```
# mkdir /home/lightdm-data
# chmod 755 /home/lightdm-data
# mkdir /home/lightdm-data/someuser
# chown someuser:lightdm /home/lightdm-data/someuser
# chmod 770 /home/lightdm-data/someuser
# ln -s /home/lightdm-data/someuser /var/lib/lightdm-data/someuser
```

Any remaining locations, if any, should be added to the deny rules in the AppArmor configuration as shown below. The command `# find / -type d -writable` will identify directories where a user can write files when run as that user, where locations on partitions where execution is not possible can be ignored.

Install additional AppArmor profiles and set ones for software that is installed into enforce mode. The extra profiles can be obtained by executing `# apt-get update` followed by `# apt-get install apparmor-profiles apparmor-utils`. Then execute the following to put specific profiles for software that is installed in the recommended configuration into enforce mode:

```
# aa-enforce /etc/apparmor.d/usr.bin.firefox
# aa-enforce /etc/apparmor.d/usr.sbin.avahi-daemon
# aa-enforce /etc/apparmor.d/usr.sbin.dnsmasq
# aa-enforce /etc/apparmor.d/bin.ping
# aa-enforce /etc/apparmor.d/usr.sbin.rsyslogd
```

Also note that ensuring the end user's account does not have `sudo` access is also important to ensure proper software restriction. More details on this can be found in the User Setup section below.

If shell access is not required it can be disabled. To do this, before creating any users, set the default shell to `/usr/sbin/nologin` in both `/etc/default/useradd` and `/etc/adduser.conf`. This prevents users gaining access to the shell via the console, SSH, or the GUI.

8.4 User setup

1. The default Ubuntu installer, Ubiquity, will create a user account during install. For the recommended configuration this user account should be considered to be the administrator of the system and not be assigned to the end user of the device. This user account will have `sudo` access with which it can perform administration tasks. After install, another user account should be created with `adduser` with default options. This user account should not have `sudo` access and the password can be provided to the end user of the device. As a further step, this user account can be prevented from executing the `su` command by running `# dpkg-statoverride -`

-update --add root adm 4750 /bin/su to change the commands permissions. Similar protection can also be applied to gksu, gksudo, pkexec, pkcon and so on, as appropriate.

2. Disable guest login through LightDM by creating `/etc/lightdm/lightdm.conf.d/50-no-guest.conf` with the following content (leaving a blank line at the end of the file):

```
[SeatDefaults]
allow-guest=false
```

1. Remove read access to user home directories and set a more secure default umask:
 - For any existing home directories, use `chmod 750`.
 - In `/etc/adduser.conf` ensure the `DIR_MODE1` setting is set to `0750`.
 - In `/etc/login.defs` ensure the line `UMASK` setting is set to `027`.

8.5 Privacy

By default Ubuntu has some features enabled which can be a privacy concern. To disable these features take the following steps:

1. Disable Apport error reporting by ensuring that `/etc/default/apport` contains `enabled=0`.
2. To prevent what is typed into the Dash from triggering online searches, go to System Settings, Privacy, Search, and set `Include online results in Dash` to disabled. Alternatively, online scopes can be disabled by executing `$ gsettings set com.canonical.Unity.Lenses remote-content-search 'none'` and `$ gsettings set com.canonical.Unity.Lenses disabled-scopes "['more_suggestions-amazon.scope', 'more_suggestions-u1ms.scope', 'more_suggestions-populartracks.scope', 'music-musicstore.scope', 'more_suggestions-ebay.scope', 'more_suggestions-ubuntushop.scope', 'more_suggestions-skimlinks.scope']"`

These settings should be locked so users cannot unset them by doing the following steps:

- Create a `/etc/dconf/profile/user` file containing:

```
user-db:user
system-db:local
```

- Create a `/etc/dconf/db/local.d/unity` file containing:

```
[com/canonical/unity/lenses]
remote-content-search=false
```

- Create a `/etc/dconf/db/local.d/locks/unity` file containing:

```
# Do not allow remote content searching in Unity
/com/canonical/unity/lenses/remote-content-search
```

Then run `# dconf update`.

The provided installation script runs these commands each logon via a LightDM hook to ensure such scopes are always disabled.

8.6 VPN

StrongSwan VPN is capable of supporting the PSN Interim IPsec Profile and the PSN End-State IPsec profile. StrongSwan has not completed the CESG CPA process so at this time it provides technical, but not assured, data in transit protection.

Recommended PSN IPsec profile configuration summary:

PSN end-state IPsec profile

ESP

Encryption	AES-128 in GCM-128
------------	--------------------

IKEv2

Encryption	AES-128 in GCM-128 (and optionally CBC*)
------------	--

Pseudo-random function	HMAC-SHA256-128
------------------------	-----------------

Diffie-Hellman group	256-bit random ECP (RFC5903), Group 19
----------------------	--

Authentication	ECDSA-256 with SHA-256 on P-256 curve
----------------	---------------------------------------

*If supporting CBC for IKEv2 encryption, the integrity algorithm that should be used is HMAC-SHA256-128

A sample strongSwan default section for the above is:

```
conn %default
    keyexchange=ikev2
    ike=aes128gcm128-prfsha256-ecp256!
    esp=aes128gcm128-ecp256!
    ikelifetime=60m
```

```
lifetime=20m
margintime=3m
keyingtries=%forever
closeaction=restart
dpdaction=restart
```

PSN interim IPsec profile

IKEv1

Encryption	AES-128 in CBC mode
Pseudo-random function	SHA-1
Diffie-Hellman group	Group 5 (1536 bits)
Authentication	RSA with X.509 certificates

8.7 Firewall

The firewall should be configured to block incoming connections, external connections can also be limited to only allow access to provisioned enterprise services. This can be achieved using an ufw configuration such as:

```
# ufw limit 22/tcp # Limit incoming SSH connections to 6 connections per IP address per 30 seconds
# ufw enable
```

Restrict outgoing access solely to the VPN server (172.99.99.2 in this example):

```
# ufw allow out proto udp to 172.99.99.2 port 500
# ufw allow out proto udp to 172.99.99.2 port 4500
# ufw default deny outgoing
```

Normal traffic will be able to flow down the IPsec tunnel once the security association is established (leftfirewall=yes must be set in the StrongSwan configuration file).

In order for DHCP on the LAN to function, this must be allowed:

```
# ufw allow out from any port 67 to any port 68 proto udp
# ufw allow out from any port 68 to any port 67 proto udp
```

Alternatively for more fine-grained configurations, the `iptables-persistent` package may be used:

```
# apt-get install -y iptables-persistent
```

9. Enterprise considerations

The following points are in addition to the common enterprise considerations, and contain specific issues for Ubuntu deployments.

9.1 Application whitelisting

Ubuntu can be configured such that users cannot run programs from areas where they are permitted to write files. This ensures users can only access programs provisioned by an administrator, although this also prevents users from installing pre-approved software by themselves.

In addition, it is recommended that users do not have access to script interpreters such as Python, Perl, or shells including bash. Access to these can be restricted using a combination of AppArmor, file permissions, and file attributes.

9.2 Auditing

Auditing can be enabled and configured via an application called `auditd`. Rules can be configured which enable auditing of various system events.

This can be installed with the following command:

```
# apt-get install -y auditd
```

Additional configuration is required in order for `auditd` to monitor for events. The following examples can be used.

Monitoring changes and execution within `/tmp`

In `/etc/audit/rules.d/tmp-monitor.rules`, place the following configuration:

```
# Monitor changes and executions within /tmp
-w /tmp/ -p wa -k tmp_write
-w /tmp/ -p x -k tmp_exec
```

Monitoring administrator access to /home directories

In `/etc/audit/rules.d/admin-home-watch.rules`, place the following configuration:

```
# Monitor administrator access to /home directories
-a always,exit -F dir=/home/ -F uid=0 -C audit!=obj_uid -k admin_user_home
```

If files are created or modified within `/etc/audit/rules.d/` then `# augenrules` must be run to merge the changes to the main `auditd` configuration. Once run, restart `auditd` with `service auditd restart`

Audit events are then recorded in `/var/log/audit/audit.log`. This file can be parsed with various tools, such as `aureport`. Example usage of this command is `# aureport --input /var/log/audit/audit.log`

To test the above configuration, the following commands will trigger the auditing rules:

```
$ touch /tmp/audit_test_file
$ chmod u+x /tmp/audit_test_file
$ /tmp/audit_test_file
$ sudo -i
# ls /home
```

Legal information

This guidance is issued by CESG, the UK's National Technical Authority on Information Assurance. One of the roles of CESG is to provide advice to UK government entities and organisations providing services to UK government. The guidance found here is provided and intended for use by this audience. It is provided 'as-is' as an example of how specific requirements could be met. It should be used to help inform risk management decisions on the use of the products described, but it should not be used for procurement decisions; it is not intended to be exhaustive, it does not act as an endorsement of any particular product or technology, and it is not tailored to individual needs. It is not a replacement for independent, specialist advice. Users should ensure that they take appropriate technical and legal advice in using this and other guidance published by CESG. This guidance is provided without any warranty of any kind, whether express or implied. It is provided without any representation as to the accuracy, completeness, integrity, content, quality, or fitness for purpose of all or any part of it. CESG cannot, then, accept any liability whatsoever for any loss or damage suffered or any costs incurred by any person as a result of, or arising from, either the disclosure of this guidance to you, or your subsequent use of it. This guidance is UK Crown Copyright. All Rights Reserved.

