Guidance

End User Devices Security Guidance: Ubuntu 12.04

Updated 14 October 2013

Contents

This guidance is applicable to any device running Ubuntu 12.04. 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 2017.

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 twelve 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. Open source patches exist for LUKS to enable usage of Trusted Platform Modules (TPMs) which may help meet more of these requirements.
- Whilst Ubuntu will boot on hardware configured to use Secure Boot, the integrity of the kernel is not chained to the bootloader in order to guarantee its integrity.
- Ubuntu does not use any dedicated hardware to protect its keys. If an attacker can get physical access to the device, they can extract password hashes and 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 strong 9-character password (or 6-character plus second factor) to authenticate themselves to the device. This password unlocks a key which encrypts certificates and other credentials, giving access to enterprise services.

The user can implicitly authenticate to the device by decrypting the disk at boot time.

4.4 Secure boot

There is currently no secure boot mechanism in a standard Ubuntu platform.

4.5 Platform integrity and application sandboxing

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

4.6 Application whitelisting

Devices 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, possibly remotely using Landscape or an alternative.

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, which are applied by core operating system components, and DConf settings, which applied by the Operating System based upon files created by the DConf toolkit.

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

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, eSATA, and Thunderbolt. As this platform does not control access via DMA it is advisable to procure hardware which does not have 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. 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 a variety of “syslog” options to output event logs. Whilst syslog supports the use of remote servers (via UDP messages) if the remote log server is not available then those log messages are silently discarded.

The ability to locally cache log events when connectivity to the log server is not present is a common feature of Software Configuration Management systems such as Landscape.

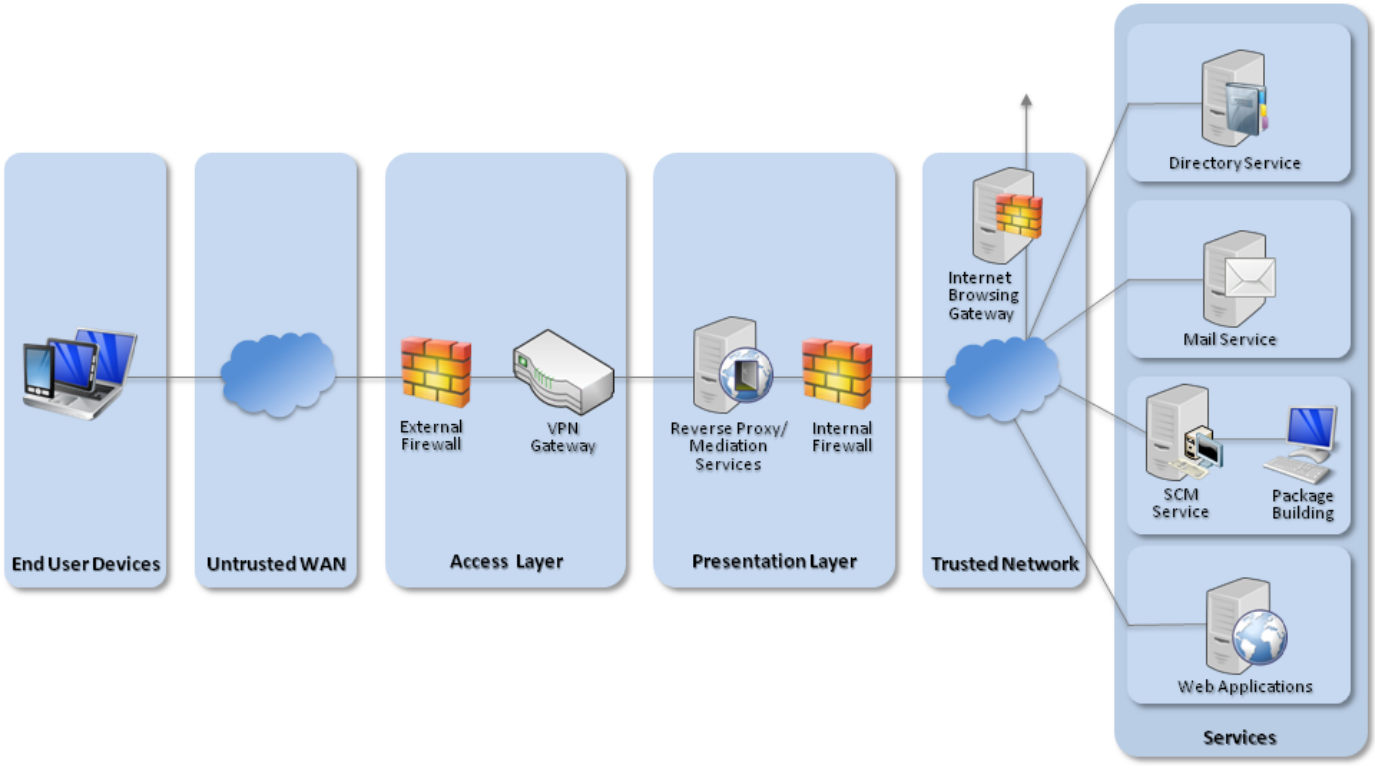
4.12 Incident response

There is no native remote wipe functionality available for Ubuntu.

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

All remote or mobile working scenarios use a typical remote access architecture based on the Walled Garden Architectural Pattern.



Recommended network architecture for Ubuntu deployments

6. Deployment Process

1. Procure and provision a Software Configuration Management server, such as Landscape.
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 12.04.2 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 x86_64 Ubuntu 12.04 Desktop Live CD.
2. At the GRUB menu, the “Try Ubuntu without installing” option should be selected.
3. Once the desktop has loaded, a GUID partition table (GPT) should be created on the installed hard disk. Within

this, there should be a 100MB FAT32 partition, a 2048MB EXT4 partition, and an EXT4 partition occupying the rest of the disk.

4. LVM should then be installed into the live CD environment, and a LUKS partition created on `/dev/sda3` using aes-xts-plain as the cipher.
5. The LUKS partition should then be opened and a LVM volume group created.
6. An LVM volume should then be created within the mounted LUKS partition and made as part of the volume group, and an ext4 file system created on it.
7. Ubuntu should then be installed with the Ubiquity installer, with the 100MB partition set to be an EFI boot partition, the 2048MB partition set to be an EXT4 partition mounted at `/boot`, and the encrypted partition set to be mounted at `/`.
8. Once the installation process has finished, a `crypttab` file should be created, and lvm2 and cryptsetup installed into the new installation prior to rebooting. Additionally, the relevant modules should be set to be loaded by initramfs to ensure the encrypted volume can be opened and the operating system loaded.
9. 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.
10. Configure StrongSwan to connect to the VPN gateway using the relevant profile and certificate files, and then restart the StrongSwan daemon.
11. Connect to the VPN and enrol the device with the Software Configuration Management server.
12. On the SCM server, place the newly added device into the appropriate groups and deploy the configuration packages.
13. Create a new local account for the user, login and add any required user certificates from the domain CA into the appropriate certificate store(s).

8. Policy Recommendations

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

8.1 Software Restriction

1. Remove all shell access. Before creating any users, set the default shell to `/bin/false` in both `/etc/default/useradd` and `/etc/adduser.conf`. This prevents users gaining access to the shell via the console, ssh, or the GUI.
2. Separate partitions should be created for `/tmp` and `/home` and 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` and `/tmp noexec,nosuid,nodev` to `/etc/fstab`
3. Any additional locations on the filesystem that the user can both write to and execute from should be identified and locked down by changing group membership or directory permissions. Any remaining locations should be added to the deny rules in App Armor configs as shown below. The command `find / -type d -writable` will identify directories where a user can write files when run as that user.

4. Interpreters included with Ubuntu are required by the system for normal operation and cannot be uninstalled, however they increase the attack surface significantly as users can download or create scripts and run them. The following is an example App Armor configuration file which prevents Python from accessing files in the user writable locations defined above:

```
#File: /etc/apparmor.d/usr.bin.python2.7
/usr/bin/python {
  #include <abstractions/base>
  /usr/bin/python2.7 mr,
  deny /home/** rw,
  deny /tmp/** rw,
  deny /some/user/writable/directory/** rw,
  /** rw,
}
```

Similar App Armor configurations can be used for Perl, Ruby or other interpreters required only by the system.

8.2 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 must be used is HMAC-SHA256-128

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

8.3 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 achieved using an iptables configuration such as:

```
/sbin/iptables -F
/sbin/iptables -X

/sbin/iptables -P OUTPUT ACCEPT

/sbin/iptables -P FORWARD DROP

/sbin/iptables -P INPUT DROP

# allow incoming ssh
/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Allow stateful return traffic
/sbin/iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# allow all traffic on loopback interface
/sbin/iptables -A INPUT -i lo -j ACCEPT
/sbin/iptables -A OUTPUT -o lo -j ACCEPT
```

When iptables has been configured with the above rules, it's configuration should be saved to a file using `iptables-save > /etc/fw-rules` or similar.

Finally, the permissions on this file should be changed to root read only: `chmod 400 /etc/fw-rules`

For iptables to load rules at startup, create the following script in `/etc/init/netfilter.conf`

```
# netfilter - start netfilter packet firewall
#
# Initialise netfilter from the predefined rules files

description "netfilter firewall"
start on (starting networking)
```

```
pre-start script
 iptables-restore < /etc/fw-rules
end script
```

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 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 App Armor, file permissions, and file attributes.

Legal Information