

System Configuration as a Privilege

Glenn Wurster, Paul C. van Oorschot
School of Computer Science
Carleton University, Canada

HotSec 2009 — 11 Aug 2009

The Configuration Privilege

Separate configuration privilege from traditional root

Why separate the two?

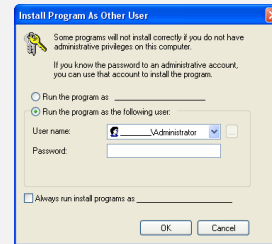
- ① System is normally used for performing work
 - e.g., reading e-mail, coding, writing papers
- ② Prevent stealthy configuration changes
- ③ Restrict the abilities of installers

1. Compromised daemons/root applications
 - No limit to configuration changes
2. Dubious installers
 - Sony DRM, Kazaa

Installing Applications

Typical procedure

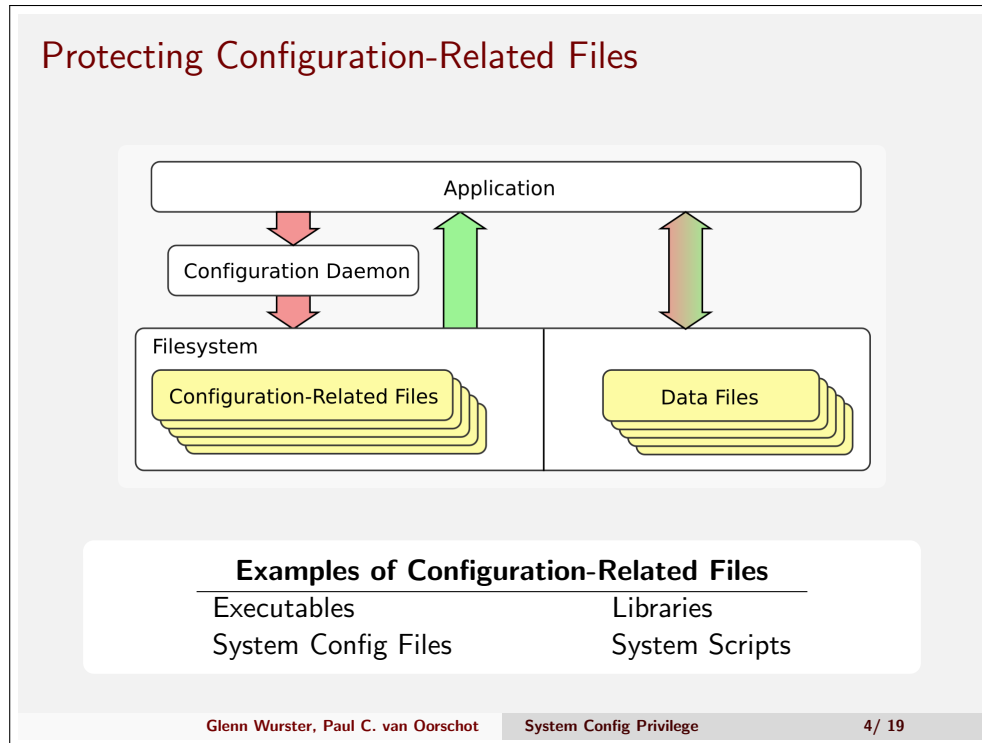
- 1 Become the superuser
- 2 Run the installer



Problem:

- Granting unrestricted access to the file system

1. User knows when they want to install something
2. Explicitly grants superuser privileges
 - Can run arbitrary code
 - Can overwrite arbitrary files
3. Source of installer is downloaded code?
 - The complete opposite of what we're trying to tell them



1. Information Flow

- Read and write to data files
- Only read from configuration-related files

2. Application Includes Installers

3. Regardless of whether the program is run as root

4. Example of Data file:

- Your pictures of Montreal

Primary Existing Mechanisms for File-System Protection

Many proposals

- ① Discretionary
 - e.g., UNIX and ACL's
- ② Mandatory Access control
 - e.g., SELinux, AppArmor
- ③ Physical
 - e.g., read-only media
- ④ Reactive
 - e.g., Tripwire

What about software installs/updates?

1. Assumption is the system is in a steady state
 - No new applications being installed/removed
2. Need very-high privileges to install/upgrade
 - Read-only media needs to be made writable
 - Tripwire needs manual merging of changes
3. My Observations
 - All handle installs, but were not designed for it.
 - Even MAC systems don't directly tackle the problem of how you get new/updated software onto the system.

The Alternative Install Approaches Have Their Own Problems

- ① Do everything manually
 - Technical expertise required
- ② Application bundles/packages
 - Caveat: binaries/scripts run during application install
- ③ Track system changes
 - User surveillance

1. Manual Install:

- Thick book of instructions
- No installer run with superuser privileges

2. Application bundles:

- Linux package managers - still run scripts with root
- Apple bundles - don't know about scripts

3. Tracking Changes:

- System Checkpointing, Sandboxing
- User must run tools before install

Classifying Installers

- ① What is an installer?
 - Is a FTP server an installer?
- ② Not all installers are created equal.
 - Game vs. OS upgrade
- ③ Update vs. Upgrade vs. Install

1. What is an installer
 - Hard question
 - Prone to subversion by malware
2. Different installers need to do different things
 - Security Updates change little
 - OS Upgrades change much
3. Any attempt to subclass the installer space?
 - Apparently not
4. Not sure we need to classify installers
 - Configuration privilege still a contribution without classification

Limiting Configuration Actions

Two options:

- ① Identify installers and limit them ☹️
 - Identifying installers is probably hard
 - What about non-installers?
- ② Limit all applications 😊
 - Also protects against other attacks
 - The approach we use

1. From Previous: Identifying what is an installer is hard
2. Alternative install approaches (manual, packages, tracking) required identifying installers
3. Non-installers should not be allowed to configure the system
4. Want to limit dangerous configuration operations regardless

Limiting all Applications

Prevent modifications to system configuration

- 1 Create a new privilege, the *configuration privilege*
 - Tied to the ability to modify system configuration

Modifying system configuration

Operations having a direct visible effect on configuration files or applications on disk

- Scripts
- Startup Files (OS and application related)
- Executable files

1. Privilege required to modify system configuration
 - Files on disk
2. Identify configuration changes and limit them
3. Privilege required to modify configuration files on a system

Limiting Configuration-Related Changes

Enforcement points

- 1 Kernel enforcement
 - Not the best fit
- 2 Proxy enforcement (Configuration Daemon - configd)
 - Our proposal - a single choke-point

Ways to limit configuration operations

- 1 User input
- 2 What files are modified
- 3 How the files are modified
- 4 The previous contents of the modified files
- 5 ...

Access control on non-traditional properties.

1. Kernel enforcement limits choices
 - Rootkit-resistant disks
 - Code-signing
2. Configd
 - A daemon which responds to configuration requests
3. User-input is intentionally broad
 - Traditional Keyboard/Display
 - USB Keys
 - Location Sensor
 - Biometrics

The Crux

*This application requires configuration privileges in order to install.
Please re-run this application after logging in with configuration
privilege.*

❗ The privilege alone is insufficient

1. Problem:

- Developers pop up a dialogue box
- Users follow the instructions
- We've just shifted the problem.

2. Not sufficient to just create the privilege

- Must specify how the privilege is used

3. Restrict in a way that developers can't convince the naive user to run with elevated privileges.

Preventing a Shell Game

- ① Don't let the user grant configuration privilege
- ② Restrict configuration privilege to a single system daemon
 - Enforce access-control protections in the daemon

1. Reduce the effectiveness of social engineering
2. A single system daemon
 - Make it hard to get around

How Much Complexity Should Be Exposed?

Desired target audience:

Non-technical users



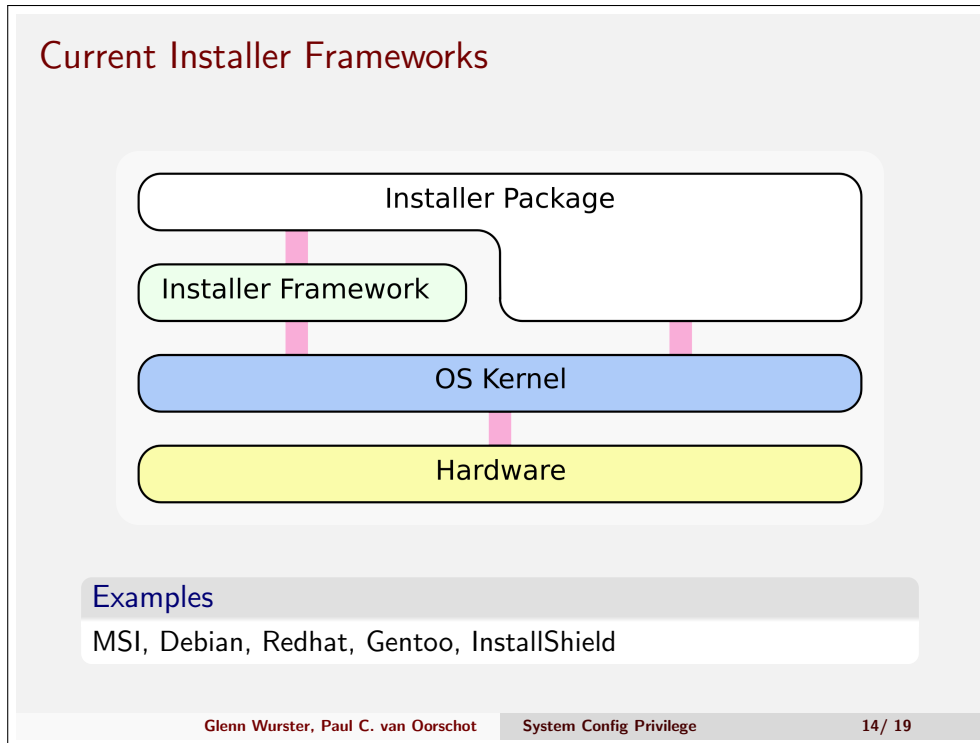
Fine-grained access control solutions which require users to get involved with the fine grains usually fail due to usability issues.

Glenn Wurster, Paul C. van Oorschot

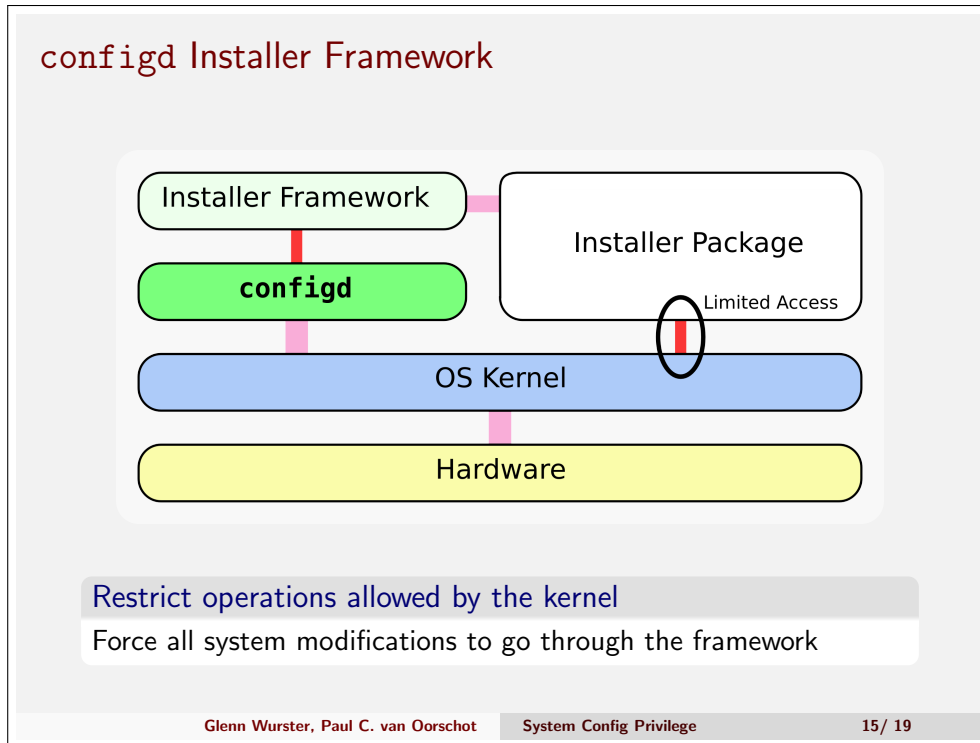
System Config Privilege

13 / 19

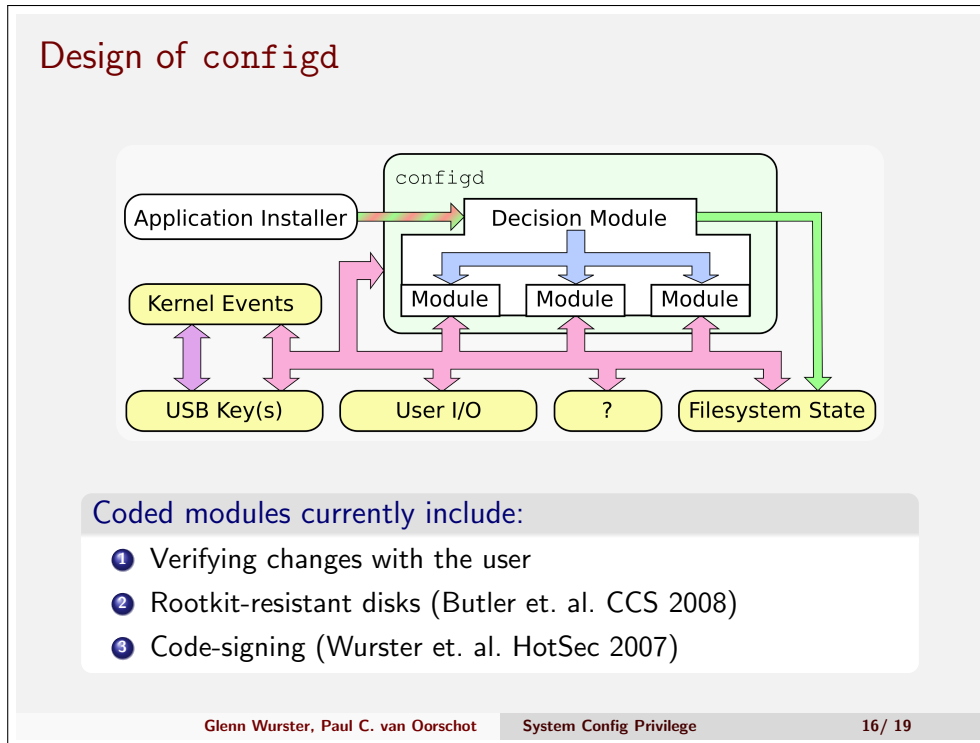
1. The collection of parts in a drill
 - More useful than the drill, but harder to use
2. Going past determining all the pieces we need
3. Involves determining how the pieces work together
4. Decrease the complexity exposed to the user



1. Installers can bypass most frameworks by talking directly to the OS kernel
 - Including all frameworks I'm aware of
2. OS does not restrict activities
 - Installer is run as superuser



1. Applies to all applications
2. Interface with OS kernel is limited
3. Configd can reject requests for configuration changes



1. Each Module:
 - Examines elements of system state
 - Rejects, Allows, or Postpones making a decision
2. Base Configd also responds to system events
3. Rejected requests do not modify file-system state
4. Modules work together
 - e.g., Code-signing combined with Rootkit-resistant disks
5. Mix of modules is not right yet.
6. Experimenting with asking the user
7. Extensible to try out new ideas
 - Module list is not designed to be modified at run-time

Additional Design Details

Kernel responsibilities

- 1 Suspend other programs when requested by `configd`
- 2 Prevent root from modifying `configd`
- 3 Restrict configure permission to `configd`

`configd` responsibilities

- 1 Respond to configuration change requests
- 2 Queue up requests until a specific USB key is inserted
- 3 Perform allowed changes to system configuration
- 4 Notify the kernel what configuration-related files to protect

1. Kernel:

- Prevents others from getting configure permission
- Protects the `configd` process
- Protects configuration-related files

2. `Configd`:

- Deals with requests for configuration changes
- Notifies the kernel what files are configuration related

3. Prevent race conditions by suspending other tasks

Current Status

Progress made

- 1 Created `configd`
- 2 Modified the Linux kernel to restrict file-system modifications

Next steps

- 1 Test `configd` on a Debian system
- 2 Reduce the technical expertise required to use `configd`

Glenn Wurster, Paul C. van Oorschot System Config Privilege 18 / 19

1. Done:

- Config framework and modified Linux kernel

2. In progress:

- Integrating `configd` into a Debian system
- Start to determine what the right mix of modules should be

Questions