Gjøvik University College

Software Security

Trusted Path Execution in Linux

Assignment #1

Victor Rudolfsson - 120912

September 3, 2014

1 Trusted Path Execution

Trusted **P**ath **E**xecution is an approach to prevent users from (un)intentionally executing malicious code, by ensuring that only code installed by a trusted user (or user belonging to a trusted group) can be executed. Another criteria for execution is that said code must not be writeable by an unprivileged user. [1]

Linux does not implement Trusted Path Execution by default, but the feature is available as kernel modules from a few different packages (*grsecurity* [2] mainly, but also standalone kernel modules [3]), which restricts execution of writeable binaries.

With *grsecurity* and TPE enabled, users belonging to a specific group are allowed to execute binaries if and only if they reside in directories that are non-writeable and owned by root.

grsecurity implements TPE through one of four different approaches:

1. *Basic TPE*,

2. *TPE with inverted gid*,

3. *TPE with partial restrictions*

4. *TPE with partial restrictions* ***and*** *inverted gid*

With basic TPE in *grsecurity*, only users in a specified group (100 by default) will be restricted to executing files in root owned and write protected directories; whereas other users will be unrestricted.

When *grsecurity* has TPE enabled with inverted gid, the setting is the inverse of basic TPE – trusted path execution is enabled for all users **not** belonging to a specific group (100 by default).

With *partial restrictions*, there's an additional restriction for non-root users that would otherwise be excluded from TPE in the previous rules: Execution is allowed only for executables in root owned directories only writeable by root, and the user's own directories which aren't group or publicly writeable.

The fourth set-up allows **both** TPE with inverted gid **and** partial restrictions, which means that users not belonging to a certain group will only be able to execute files in directories owned by, and writeable only by, root. Additionally, other non-root users may only execute files in directories owned by and writeable only by root, as well as own directories provided they are not world or group writeable. [2]

There's also a version ported from *grsecurity* and then further enhanced, but installed as stand-alone kernel modules hijacking system calls to enforce its TPE. In this regard, it operates similar to a rootkit but in a non-malicious way.

One of the additional features provided by this is *hardcoded_path*, which allows the administrator to further lock down execution to a single path regardless of ownership or permissions, and even root and trusted gids will be restricted to this path if the *paranoid* option is enabled. [4]

More specifically (and admittedly because I am really unsure of what else could possibly be mentioned about grsec's implementation of TPE in Linux), *grsecurity* implements TPE by first checking whether the user is root, in which case execution is always allowed. Second, it checks whether TPE is in whitelist (*inverted gid*) or blacklist mode (*basic TPE*).

It then proceeds to check whether the file is in a non-root-owned, world-writable, or group-writable directory.

If both these checks are true – that is, it's enabled and the file is in any of these kinds of directories, the incident is logged and execution will not be allowed to proceed.

Otherwise grsec proceeds to check whether the directory is owned by the user, and if the file is in a world-writable or group-writable directory, in which case execution will be disallowed. [5]

References

[1] Niki A. Rahimi. Trusted path execution for the linux 2.6 kernel as a linux security module. https://www.usenix.org/legacy/events/usenix04/tech/freenix/full_papers/rahimi/rahimi_html/. Accessed Sep 2, 2014.

[2] Francisco Blas Izquierdo Riera. Hardenedgrsecurity trusted path execution. http://wiki.gentoo.org/wiki/Hardened/Grsecurity_Trusted_Path_Execution. Accessed Sep 2, 2014.

[3] Corey Henderson. Trusted path execution linux kernel module. https://github.com/cormander/tpe-lkm/. Accessed Sep 2, 2014.

[4] Corey Henderson. Trusted path execution linux kernel module faq. https://github.com/cormander/tpe-lkm/blob/master/FAQ. Accessed Sep 2, 2014.

[5] ncopa. grsec_tpe.c. https://github.com/ncopa/linux-stable-grsec/blob/5f12d03978dee3de26cd27657f7aac60dacf780d/grsecurity/grsec_tpe.c. Accessed Sep 2, 2014.