Gjøvik University College

Linux Operating System

Software Security

Assignment #1 - Reviewed

Student:
Victor Rudolfsson - 120912
Review of Tommy Ingdal -
120913

September 9, 2014

1 Original

1 Trusted Path

Trusted Path is some kind of mechanism that ensures that the user is communicating with what they intended to communicate with. By using TP (Trusted Path) attackers and e.g. malware can not intercept or listen in on the communication.
A really good example of this mechanism is Windows' CTRL + ALT + DEL keystroke combination.
Since there's no way for e.g. malware to recognize this combination, you can be 100% sure that the login prompt you see is the real deal, and not some form of malware designed to steal your credentials.

2 Linux and Trusted Path

Many stock Linux distributions today don't implement a TP for their login prompts [3], but some kernels do support a mechanism that's equivalent to Windows' CTRL + ALT + DEL keystroke combination called Secure Attention Key (SAK) [2].
In order to use SAK, some kernels need to be compiled with sysrq support [1], and it is recommended that you map SAK to CTRL + ALT + PAUSE, since other applications, e.g. malware can't recognize this keystroke combination.

2.1 Secure Attention Key

If you want to log in on a Linux box you should always activate and trigger the SAK. When SAK is triggered the running X-server will be killed, running programs is killed, the user is logged out of the system and the login prompt appears, allowing the user to log in without the risk of something intercepting the communication.

2.2 Trusted Path Execution

When you have multiple users on a system, you should always have some sort of security implementation which can limit what each user can do. This is especially important when we talk about running arbitary code, i.e. binaries.

Trusted Path Execution (TPE) can be use to control or minimize this security risk.

The TPE module uses a kernel hook in the Linux Security Module framework [4] which perform a check at the exact time a program is executed. If the user executing that specific program is not found in module's access control list and the path the program is executing from is not trusted, the execution of the program is terminated.
This way the administrator(s) of the system can add or remove users based on wether they are trusted or not. And because of this, the risk of a system compromise is minimized.

References

[1] kernel.org. Linux 2.4.2 Secure Attention Key (SAK) handling. https://www.kernel.org/doc/Documentation/SAK.txt. [Online; accessed 02-September-2014].

[2] lwn.net. SAK. http://lwn.net/2001/0322/a/SAK.php3. [Online; accessed 02-September-2014].

[3] tldp.org. 7.12. Set up a Trusted Path. http://www.tldp.org/HOWTO/Secure-Programs-HOWTO/trusted-path.html. [Online; accessed 02-September-2014].

[4] usenix.org. Trusted Path Execution for the Linux 2.6 Kernel as a Linux Security Module. https://www.usenix.org/legacy/event/usenix04/tech/freenix/full_papers/rahimi/rahimi_html/index.html. [Online; accessed 02-September-2014].

2 Review

Although it briefly introduces what a Trusted Path is with a clear clear example that any Windows-user not previously familiar with the concept would be sure to understand, additional examples or implementation details would have been rad. I believe centered text to be a deadly sin which should be punished accordingly, but I found few other flaws – I did find things I had overlooked in my own report, though (such as SAK), and feel that although this report had left out many things I considered important to include, it also includes things I overlooked.

Grade: Passed