Implementation of trusted path in Linux

1. What is trusted path?

Trusted path is an additional security layer. It should limit the execution rights from files in special cases.[1] A typical example in which trusted path should protect the user is a fake log-in screen. That means an other program on the PC simulates the log-in prompt to capture the password.[2] Trusted path should avoid these attack scenario.

2. Secure attention key (SAK)[3][4]

The security attention key is a way to solve the problem with the fake log-in screen. It should protect the user against password capturing programs. If the user press the SAK sequence on the keyboard than SAK kills the running X server (and with that all applications that could be camouflaged as log-in prompt) and restart it. It exist different SAK sequences e. g. with the Seed Ubuntu from the laboratory “alt+print+k” works fine. But in Linux Mint “ctrl+alt+backspace” works. In the windows world is it comparable with “crtl+alt+del” key sequence.
In Linux exist two ways to supporting SAK. The first and not recommended way is to compile the kernel with “sysrq” support. Then you are allowed to press the key sequence “alt+print+k”. The second and recommended way is to use the function “loadkeys”. This should work in any case.

3. Trusted path execution (TPE)

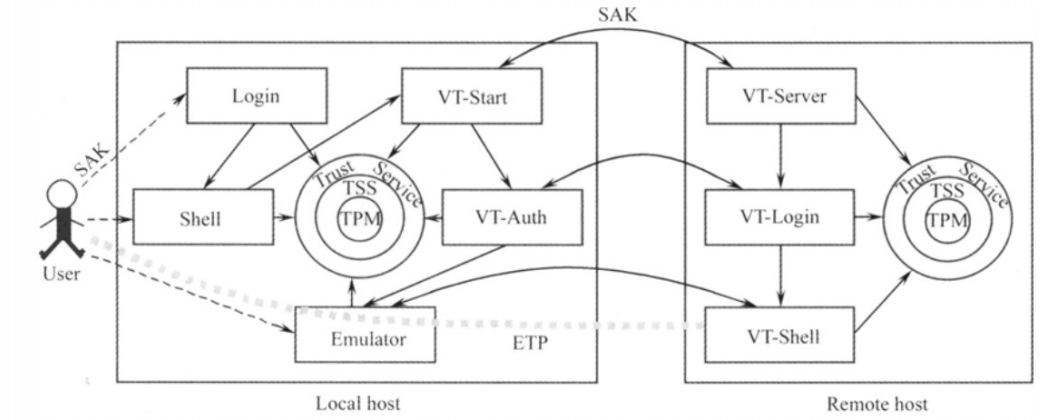
It exist different ways of implement a TPE. The one is called “partially restrict all non-root users”. That means if I download a file (and I am not root) with malicious content and want to execute that and this file want access a file that is owned by an other user, the execution will be stopped by the system.
A second way is using different user groups in Linux and make special groups for untrusted users to limit their execution rights.[5] For me it seems to be similar in the windows world with “normal users” and “guest users” that have strong limited access to system files.

Trusted Path Execution is not default activated in Linux kernel, but you have the possibility to patch Linux and compile a new kernel. There exist a various library’s for Linux to do that e. g. with “Grsecurity” [6][7] or with the elrepo project.[8]

4. Extended Trusted Path (ETP)[9]

A problem that you have very often especially in business applications is that nearly all of them are network applications. That’s why you need (e. g. to authenticate a user) the ability for trusted path over networks. This system is called extended trusted path. In the following graphic on the next page you will see a use case diagram about the architecture overview of ETP

1 https://itsecblog.de/trusted-path-execution-des-grsecurity-kernel-patches/#more-870
2 http://www.dwheeler.com/secure-class/Secure-Programs-HOWTO/trusted-path.html
3 http://www.dwheeler.com/secure-class/Secure-Programs-HOWTO/trusted-path.html
4 https://www.kernel.org/doc/Documentation/SAK.txt
5 http://www.insanitybit.com/2012/06/26/trusted-path-execution-and-antiexecutables-10/
6 https://itsecblog.de/einen-gehaerteten-linux-kernel-mit-pax-und-grsecurity-kompilieren/
7 https://itsecblog.de/tag/trusted-path-execution/
8 http://elrepo.org/tiki/kmod-tpe
9 Shi Wenchang, Wuhan University Journal of Natural Sciences Voll 11, No 6, 2006



Working Principles of ETP in the USE CASE diagram

→ Pressing SAK (e. g. “ctr+print+k”) to establish a trusted path connection to the local console

→ Shell invokes the VT-Start module

→ this will create a TCP connection to the VT-Shell from the server (in the graphic “Remote host”)

→ VT-Start (on the host) sends the SAK to the VT-Server

→ Operation system on the host verifies the integrity of the VT-Server

→ Operation system on the host check that VT-Server invokes the correct VT-Login component

→ VT-Start invokes the VT-Auth (Auth = Authentication) component

→ VT-Login communicates with VT-Auth

→ to authenticate the user

→ If authentication was successful

→ VT-Login invokes the VT-Shell

→ VT-Auth invokes the Emulator

→ Emulator communicates with the user and with the VT-Shell

→ Extended trusted path is now between the user and the VT-Shell established

Conclusion

Trusted path seems to be a powerful way to have a more secure system, but it doesn’t protect you in all situations e. g. when the attacker can manipulate a program that runs as “Trusted Program” than he is able to work in the address space of the program, because he inherit the access rights from the application.[10]

In the end TPE is only one way for a more secure system and it can't replace computer trainings for the workers and using the computer with open eyes.

10 http://www.insanitybit.com/2012/06/26/trusted-path-execution-and-antiexecutables-10/