Mandatory assignment 1
Trusted path

- From: Jan Petter Berg Nilsen
- Studnr: 120505
- Subject: IMT3501 Software security

Trusted Path(TP) in Linux

Trusted Path(TP) is a security mechanism that is built inside a operating system. The meaning of the TP is to provide a protected channel that will give privacy and authentication under data transfers, between a user‘s input/output and the trusted programs on that device. That means that no program that is not trusted can access, change and listen to input/output between user and the device. An example of a trusted path in different devices is Ctrl+Alt+Del keystroke combo in Windows and the home button on smartphones [1].

In Linux there is no implementation by default when it comes to TP, but it does exist a framework built into the Linux kernel called Linux Security Modules. There are many different types of extensions for the Linux Security Modules. Some of the extensions are, Security-Enhanced Linux (SELinux), AppArmor and TOMOYO Linux. There are also many kernel patches that will implement this future, like: grsecurity. There is also a security mechanism that is called Secure Attention Key (SAK), that helps to replace the Trusted Path [2].

Secure Attention Key (SAK)

Secure Attention Key is available on Linux version 2.2 and later versions. SAK is a special key or a key combination that you need to press to get into a TP. An example of a key combination for Linux is SysRq+K or Ctrl+Alt+Pause [3]. SAK is implemented to make attacks like login spoofing, impossible. Because when it's enabled, it will suspend the X-server and all programs before it starts a trusted login sequence, so the user can access the system in a more secure way [4] [3] .

The SAK is hard-coded into the Linux kernel and will always respond. It is also impossible for other programs to intercept the SAK when it's enabled [4] .

Trusted Path Execution (TPE)

When we have a system with multiple users, it would be nice to have a security function that will restrict users, in where to execute a program or a file. This is what Trusted Path Execution(TPE)is. So when a user executes a program or a file with malices code in its home directory, this could resolve in big damage to the system. This is what TPE is trying to prevent, by letting only trusted users execute outside the TP like: /bin or /usr/bin. When the program or the file is executed inside the system, the TPE will check if either the path or the user is trusted. If not one of them is trusted, the system will execute -EACCES error. But if either the user or the path is trusted, the program or file will execute as normal [5].

It is possible to be added as a user to the trusted user list, if it's necessary and the administrator trusts the user. Because of THE the risk will minimize when it comes to this type of severity threats [5].

Bibliography

[1] Building Verifiable Trusted Path on Commodity x86 Computers http://users.ece.cmu.edu/~jmmccune/papers/ZhGlNeMc2012.pdf

[2] Linux kernal security http://www.linux.com/learn/docs/727873-overview-of-linux-kernel-security-features/

[3] Wikipedia: SEA https://en.wikipedia.org/wiki/Secure_attention_key

[4] Kernel: Secure Attention Key (SAK) https://www.kernel.org/doc/Documentation/SAK.txt

[5] Usenix https://www.usenix.org/legacy/event/usenix04/tech/freenix/full_papers/rahimi/rahimi.pdf