



<http://www.nologin.org>

Defending Passwords Against Hardware Keyloggers And Malicious Keyboards

January 2010

warlord
warlord@nologin.org

Contents

1	Introduction	2
2	Attacks on encrypted hard drives	3
2.1	The Cold Boot Attack	3
2.2	The Hardware Keylogger	3
3	Safe Typing On Unsafe Hardware	5
4	Conclusion	7

Chapter 1

Introduction

Encrypting a hard drive or even an entire Operating System is an easy feat nowadays, and doesn't purely belong in the domain of hardcore nerds anymore. Mass-market operating systems like Ubuntu offer that option in their install process which makes setting up a fully encrypted OS an easy task even for an average computer user. Where encrypting the entire OS is not required, not possible or not wanted, setting up an encrypted partition with tools like Truecrypt is another option to safely encrypt data, allowing a container file to be opened and mounted as a drive of its own whenever access to said data is required.

The reasons for encrypting hard drives are far ranging, reaching from keeping prying parents or spouses out of ones documents to hiding data from law enforcement to life-or-death situations where so-called dissidents cannot allow their confidential information to fall into the wrong hands.

Most often these encrypted setups are unlocked by a passphrase, which should be both long and employ all kinds of different characters. If a common drive-encryption solution with strong cryptography as well as an equally strong passphrase have been selected and employed, and the passphrase hasn't been written down somewhere, the protected data is safe against decryption attacks according to todays state of the art in cryptography. That doesn't mean though there aren't other ways to attack the confidentiality of the protected data.

Chapter 2

Attacks on encrypted hard drives

While there certainly are more ways to attack the content of encrypted hard drives two important ones will be described in this chapter, one in short, one in more detail.

2.1 The Cold Boot Attack

In July 2008 a group from Princeton around J. Alex Halderman proved that it is possible to recover keyphrases out of RAM even minutes after a computer has been turned off. This allows a dedicated attacker to break into an encrypted environment by abusing physical capabilities of RAM chips. While this is a serious threat, this paper will not cover any defensive techniques intended to defeat this attack. To acquire further knowledge on the "Cold Boot Attack" reading the original paper^[2] is strongly suggested.

2.2 The Hardware Keylogger

As described earlier, encrypted hard drives or container files are usually decrypted and mounted by the use of a passphrase. The best cryptographic technology and the most well picked passphrase can't protect their sensitive data though should the passphrase somehow be disclosed, be that by a piece of paper under the keyboard or even someone watching while it is typed in. While these situations are usually easy to spot and defend against, hardware keyloggers are something entirely different.

The most (in)famous of these devices probably is the Keyghost[3], a tiny device which is attached to the cable between keyboard and computer and stores every keystroke typed into the keyboard. The current generation Keyghost stores up to 2 million keystrokes and easily dumps them all should a predefined password be entered. Any passphrases typed into the keyboard while the keyghost was attached to it will thus be in the device, and an attacker can thereby easily recover someone's keyphrase.

Probably the most famous attack of this kind was undertaken by the United States' FBI against alleged mobster Nicodemo Scarfo, Jr. Around 1999 or 2000 the FBI broke into Scarfo's home to install a keylogger which later allowed them to decrypt the content of Scarfo's PGP encrypted hard drive. [1]

While checking the cable of the keyboard every time the passphrase is typed in to make sure no extra plug is attached to it may sound like a way to defend against this kind of attack, it really isn't. Nor is using a wireless keyboard, as that sort of keylogger doesn't necessarily have to be hidden outside of the keyboard. A dedicated attacker may just as well place an extra chip inside of the keyboard or replace it altogether by a manipulated keyboard of the same model to record keystrokes without any obvious visual cues. So defending against hardware keyloggers becomes quite difficult when considering a potentially compromised keyboard that has to be used to provide the passphrase to the decryption routine. What can be done about this?

Chapter 3

Safe Typing On Unsafe Hardware

A keyboard is a very simple piece of hardware which assigns each of its keys a unique number and supplies this number to the computer when key is pressed. It is then the task of the OS to correctly interpret the provided number and display the intended letter to the user. So should a user press the letter 'l' on the keyboard, the OS will in just about all cases display the letter 'l' on the screen. Why may it not do so in all cases? Because of keyboard layouts.

While the average American can easily type the sequence 'qwerty' on his keyboard by typing the 6 keys on his keyboard starting from 'q' from left to right, a similar person in Romania couldn't do so. If he would start at 'q' and type the 5 letters right of this one, the result would be 'qwertz'. The reason for this is a different type of keyboard layout which is in use in Romania, so that keys are not be in the same physical place as on American standard keyboards. Wikipedia has a nice article on keyboard layouts.[4]

One doesn't require a Romanian keyboard though to use the Romanian layout in America. Switching keyboard layouts is an easy thing to do in most operating systems, which means an average keyboard can be used for all sorts of layouts, as it's the interpreting software of the OS which decides what keys were just pressed. So the man in Romania doesn't require special dedicated hardware to type in his native layout. It's the same hardware as the American uses, just interpreted differently by the Romanian's OS. Why is that important?

Consider the case of someone recovering a Keyghost that shows a passphrase of 'qerty'. Surprisingly though, the passphrase doesn't actually open the encrypted hard drive, even though that same passphrase appears to work fine for the person it was grabbed from. The reason may be, that the real password

actually is 'qwertz' as a Romanian keyboard layout is in use. So it is possible to confuse the keylogger, and make it display the wrong password even though it recorded the very keys that were pressed. How can this be used to the defenders advantage?

The attacker will quickly realize that potentially a different keyboard layout was used, 'querty' is replaced with 'quertz' and the attack succeeded. But what if the attacker doesn't know which layout was used by the defender? Trying all the known ones is an easy feat, few as they are, but there could be unknown ones. Consider the following scenario. Prior to entering a passphrase, the decryption software assigns a randomly generated keyboard layout, and displays the new key mapping on the screen. The user then has to employ this mapping to type the password, resulting in a password looking very much different from a hardware keyloggers point of view every single time. If the backspace key is remapped too, one could even use that while typing the password, resulting in various sizes of the passphrase. While a keylogger would be fooled, the OS would know which letter was intended every single time.

As a simple single remapping would still allow to identify the amount of similar letters in a passphrase due to the use of the same keys to type these, a further improvement of the suggested scheme could rearrange the keyboard layout after every keystroke. By displaying the original keymapping alongside of the new one, picking the right keys to hit should be an easy task and not too much of a hassle, allowing for a still relatively fast way to provide a passphrase even on an unknown keyboard layout.

Chapter 4

Conclusion

Security always is a process, and not something that can simply be enabled or disabled (or bought in a box) by the use of tools or software. While the scheme to prevent hardware keyloggers in this paper may not be an all-out solution to protecting sensitive data in a highly technical word, it should serve to once again raise the bar for attackers and make their offensive arsenal less effective. The kind of defense proposed in this paper may be seen as a hassle and surely is for passwords that have to be typed in often, but especially boot passwords for machines that are rarely turned off should this scheme be highly effective and the trade-off between hassle and security quite worthwhile.

Bibliography

- [1] One
<http://epic.org/crypto/scarfo.html>
- [2] Two
<http://citp.princeton.edu/pub/coldboot.pdf>
- [3] Two
<http://www.keyghost.com>
- [4] Two
http://en.wikipedia.org/wiki/Keyboard_layout