Firewire-based Physical Security Attacks on Windows 7, EFS and BitLocker

Benjamin Böck
Security Research Lab
Secure Business Austria
bboeck@securityresearch.at

With kind support from David Huemer

V 1.0, 2009-08-13

Latest version at
http://www.securityresearch.at/publications/windows7_firewire_physical_attacks.pdf

1 Overview

This paper discusses Firewire-based physical security attacks on Microsoft Windows 7. In the course of my research, I was successfully able to bypass the Windows 7 RTM[1] authentication check and logon with any password.

While the attack vector itself is not new, I also describe the impact of Firewire-based Windows authentication bypassing on Microsoft's full-disk encryption solution BitLocker, the Windows Encrypted File System (EFS) and Windows domains. A comprehensive section on countermeasures on different layers concludes this paper.

Moreover, David Huemer and I have developed a software solution to protect against Firewire-based physical security attacks on Windows systems which is discussed in a separate paper [01] and can be downloaded from [02].

[1] RTM: Release to Manufacturing

2 Content

1 Overview 1

2 Content 2

3 Introduction (skip to next Section for actual Attack) 2

4 Firewire-based Physical Security Attacks and Windows 7 3

4.1 Vulnerable Windows System States 4

4.2 On-the-fly Installation of Firewire Interfaces through PCMCIA/CardBus 4

5 Firewire-based Authentication Bypassing and BitLocker 4

6 Firewire-based Authentication Bypassing and EFS 6

7 Firewire-based Authentication Bypassing and Windows Domains 6

8 Countermeasures 7

8.1 Firewire 7

8.1.1 FirewireBlocker 7

8.2 Logon Settings 7

8.3 User Behavior 8

8.4 BitLocker and EFS 8

9 Annex 9

9.1 About the Author 9

9.2 About the Security Research Lab 9

9.3 Acknowledgement 9

10 References 10

3 Introduction (skip to next Section for actual Attack)

Law #3 of Microsoft's *10 Immutable Laws of Security* [03] states: "If a bad guy has unrestricted physical access to your computer, it's not your computer anymore". They list several simple attacks an adversary with physical access to a system might launch, like stealing the hard disk to gain direct access to the data. More advanced physical attacks include Cold Boot Attacks, as described by Halderman et al. [04], which involve copying of the system memory once the system has been powered off. In their paper, they also describe how to recover hard disk encryption keys (for example, for Microsoft's Windows Vista BitLocker) from such memory dumps.

Via **Firewire**, it is possible to gain read/write access to a machine's memory by means of Direct Memory Access (DMA). Ironically, it is often the more expensive (management) notebooks which come with a Firewire interface.

In 2004, Maximillian Dornseif presented a talk "0wned by an iPod" in which he described reading from and writing to arbitrary memory locations via Firewire. He also illustrated benign (memory dumping for digital forensics) and malign (cracking) applications of the aforementioned technique [05].

In 2006, Adam Boileau described how to target Windows XP via Firewire by disguising the attacking computer as an iPod [06]. He released *Winlockpwn* [07], a tool which allows mounting several attacks against Windows XP SP2, and accompanying Python bindings for easy Firewire access.

In 2008, Boileau's technique – in particular an authentication bypassing attack at the logon screen – was successfully extended to target Windows Vista [08].

The situation for Windows 7 is described in the next section.

4 Firewire-based Physical Security Attacks and Windows 7

In the course of my research, I tested Windows 7's resilience to Firewire-based physical security attacks. As a proof-of-concept, I chose the authentication -bypass attack which was first described for Windows XP SP2 by Adam Boileau in his Winlockpwn tool [06][07]. The attack works by utilizing read/write memory access via Firewire to patch the code which compares the entered to the correct password. After the in-memory patch has been applied, any password can be used to log on or unlock the system.

As described in [08], adapting the attack to Windows Vista essentially boiled down to finding the correct code location to apply the patch. Fortunately, the situation for Windows 7 is quite similar.

The comparison takes place in routine **MsvpPasswordValidate** in the "Microsoft Authentication Package" (msv1_0.dll) and is realized as a call to RtlCompareMemory. The msv1_0.dll file details for or English versions of Windows 7 Enterprise RTM were as follows:

- Windows 7 Enterprise 32 bit RTM msv1_0.dll
  - File size: **257,024 bytes**
  - File version **6.1.7600.16385**
  - Date modified **13.07.2009 18:15**
- Windows 7 Enterprise 64 bit RTM msv1_0.dll
  - File size: **311,296 bytes**
  - File version **6.1.7600.16385**
  - Date modified **13.07.2009 18:41**

Figure 1 shows the concerned code portion for Windows 7 32 bit RTM; I also successfully implemented the attack for the 64 bit version.

While Winlockpwn [07] patched the return value of MsvpPasswordValidate to always return "correct password", I use an approach similar to [08], and **NOP** out the jump to the code which sets the "incorrect password" return value. **Once the patch is applied, one can log on or unlock the system with any password**. This also holds true for local logon with a domain user.

```
.text:6D48E919                  loc_6D48E919:                           ; CODE XREF: MsvpPasswordValidate
.text:6D48E919                                                          ; MsvpPasswordValidate(x,x,x,x,x,
.text:6D48E919 6A 10                            push    10h             ; Length
.text:6D48E91B 83 C3 34                         add     ebx, 34h
.text:6D48E91E 53                               push    ebx             ; Source2
.text:6D48E91F 56                               push    esi             ; Source1
.text:6D48E920 FF 15 88 11 48 6D                call    ds:__imp__RtlCompareMemory@12 ; RtlCompareMemory(
.text:6D48E926 83 F8 10                         cmp     eax, 10h
.text:6D48E929 75 13                            jnz     short loc_6D48E93E
```

Figure 1: Password Verification within the Microsoft Authentication Package

The authentication -bypass attack is just an example of what is possible with Firewire-based attacks. With full write access to memory, one could also easily spawn a SYSTEM shell, for example.

4.1 Vulnerable Windows System States

The following table provides an overview of system states in which Windows systems with active Firewire interfaces are either vulnerable or invulnerable to Firewire-based physical security attacks. "pre-boot authentication" refers to the capability of full-disk encryption solutions like Microsoft's BitLocker to require credentials before actually booting up Windows (discussed in detail in section 5).

Windows State	Full disk encryption	
	With Pre-boot Authentication	**Without Pre-boot Authentication**
Running/locked	Vulnerable	Vulnerable
Standby	Vulnerable	Vulnerable
Powered off	*Invulnerable*	Vulnerable
Hibernated	*Invulnerable*	Vulnerable
Running/unlocked	Vulnerable*	Vulnerable*

* This is the trivial case where attackers would likely not go for Firewire-based attacks as the running system is unlocked anyway.

4.2 On-the-fly Installation of Firewire Interfaces through PCMCIA/CardBus

It should be noted that systems without Firewire interfaces are not automatically immune to Firewire-based physical security attacks as interfaces might be added to existing systems. Concerning notebooks, for example, PCMCIA/CardBus interfaces allow on-the-fly installation of Firewire interfaces. Drivers are installed in the background, even while the system is locked.

After installation, the new Firewire interface can be used to attack the system as described. This has been verified for Windows 7.

5 Firewire-based Authentication Bypassing and BitLocker

BitLocker is Microsoft's full-disk encryption solution included with some versions of Windows Vista and Windows 7.

In most configurations (if the hardware is available on the particular system), BitLocker uses a Trusted Platform Module (TPM) to store key material and ensure system integrity during the boot

process before the operating system (OS) comes up. It can also be used without a TPM but this mode does not include BitLocker's pre-OS boot integrity validation.

BitLocker (Vista) offers different modes of operation [09]:

- Basic Mode:
  - TPM only
- Advanced Modes:
  - TPM + PIN
  - TPM + USB Dongle
  - USB Dongle
- Windows 7 seems to offer an additional fourth advanced mode:
  - TPM + PIN + USB Dongle

If BitLocker is used in "basic mode" (TPM only), the encrypted disk is mounted automatically without requiring users to enter any secrets. It is presumed that most BitLocker installations use basic mode [10]. Ironically, this mode is specifically susceptible to Firewire-based attacks as the key for hard disk encryption and decryption is loaded into memory automatically before a user logs on. When the system is on the logon screen, an attacker might proceed with a Firewire-based attack as described above to bypass authentication. Optionally, memory modules could also be removed to recover key material ("cold boot attack" [04]).

According to [04], **all** BitLocker modes are vulnerable to cold boot attacks (and thus also the Firewire attack), if an attacker can gain physical access to a system while

- the screen is locked
- or the computer is in standby mode (as opposed to hibernated or shut down).

I have successfully tried my Firewire method on two Windows 7 systems with two different BitLocker advanced mode configurations:

- Windows 7 Enterprise 32bit RTM
  - BitLocker without TPM (only USB Dongle)
- Windows 7 Enterprise 64bit RTM
  - BitLocker with TPM + PIN

In both configurations, I could successfully bypass the Windows authentication password check and logon with any password if the BitLocker pre-boot authentication had already passed – that is,

- the system was at the logon screen,
  - either with a logged on user who locked the screen
  - or (e.g., right after booting the system up) with no user logged on
- or the system was brought back from standby mode (no pre-boot authentication necessary).

Conclusion: If Windows 7 is running or can be brought back into running state from standby mode, Firewire-based physical security attacks succeed, even if BitLocker is enabled.

6 Firewire-based Authentication Bypassing and EFS

The Windows Encrypted File System (EFS) enables encryption of data so that only the legitimate owner can access them. In short, a user has to log on using correct credentials; else the data will be inaccessible. Authentication bypass schemes will thus *not* allow direct access to EFS-encrypted data (This has not always been the case, however: in Windows 2000, the password obviously was not used in the EFS algorithm which allowed for simple bypass attacks).

In the course of my experiments, I used my Firewire-based attack to target a Windows 7 RTM box displaying the logon screen. Suppose there exists a user with EFS-protected data. Using the technique, we can log on as the legitimate user with any password of our liking, including a blank one. However, two cases have to be distinguished:

1. The system is at the logon screen with no logged-on user (e.g., directly after booting up)
    - After bypassing the authentication password check, as expected, we cannot access the user's EFS data, as a wrong password is used for EFS access.
2. The user is logged on but the system is locked or in standby (e.g., while the user is away)
    - Bypassing password entry for a *locked* system provides us with *full access to the user's EFS-protected data*.
    - This has been verified on our Windows 7 test boxes.

Conclusion: if a user is logged on, the system is running or in standby mode and Firewire-based authentication-bypass attacks are possible, adversaries can gain instant access to EFS-protected data.

It should be noted that other attacks on EFS are possible once physical access is given.

7 Firewire-based Authentication Bypassing and Windows Domains

Bypassing the authentication password check and gaining *local* access to a system is one thing. Usually, however, most machines found in the wild are members of Windows domains. As a result, part of my tests focused on the impact on Windows domain users. While the system has **network connection to a domain controller**, authentication-bypass attacks as described above on domain users **fail**.

However, I was able to successfully logon to or unlock such systems through above mentioned authentication-bypassing attacks after *interrupting* the network connection. Doing so presents attackers with local access to a system with the identity of a domain user. Nevertheless, after enabling network access again, access to network resources within the domain fails due to incorrect credentials. For example, not even network shares accessible by “everyone” can be accessed by a domain user with incorrect credentials (this is in fact an interesting case where a user outside the domain with arbitrary user name/password actually has more privileges than a domain user with incorrect credentials).

Conclusion: domain users’ authentication can be bypassed as well while systems cannot communicate with domain controllers. However, only local access to systems is granted.

8 Countermeasures

8.1 Firewire

As the Firewire vulnerability is hardware-based and related to the Firewire specification, it seems unlikely that a future software patch will fix it. This is not a Microsoft Windows issue; according to [06], other operating systems were even easier to target (without having to disguise as a legitimate Firewire device).

Currently, the only effective defense for Windows is to disable all Firewire controllers in the device manager – and also disable all PCMCIA/CardBus controllers as well, so attackers cannot add Firewire interfaces on the fly.

8.1.1 FirewireBlocker

As a result of the research described in this paper, we developed **FirewireBlocker**, **a software-based protection against Firewire-based physical security attacks**. Basically, it is a Windows service that disables Firewire and PCMCIA/CardBus controllers when a system is unattended. It was successfully tested on Windows Vista and Windows 7 RTM. Design and implementation of FirewireBlocker are described in a separate paper [01]. The software can be downloaded from [02].

8.2 Logon Settings

If a system shows the user name while it is locked, shows the user name of the last successful logon or a list of existing user accounts, attacks as described above are easier to conduct. Windows 7 possesses two distinct security policies which can be set locally via `secpol.msc` or rolled out via group policies. Figure 2 displays the correct settings to prohibit unwanted leakage of user name information. As most of the times, this is a trade-off between usability and security: users now have to enter not only their password but also their user name whenever they want to log on or unlock their system.

By all means, giving away your username is an invitation to attackers to try and guess the right password.

These settings present best practice and will not protect per se against a Firewire-based attack.

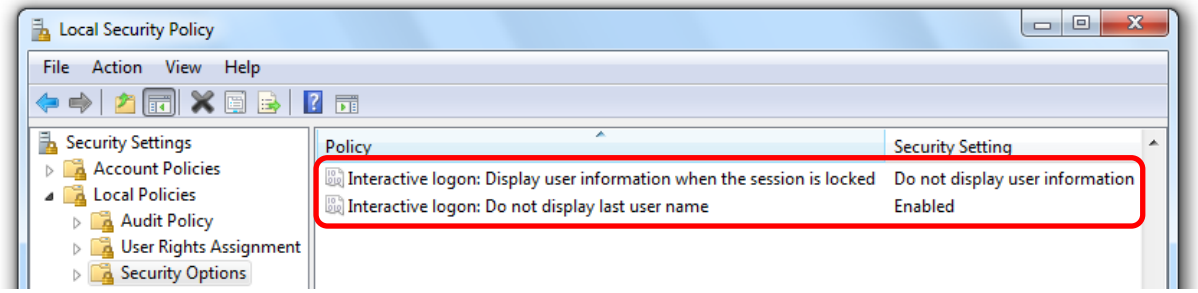

Figure 2: Windows 7 security policy settings to prohibit unwanted disclosure of user names

8.3 User Behavior

When a user leaves the computer, the system should be hibernated or turned off as opposed to locked or put into standby (of course, this does not apply to server systems which are supposed to run unattended). At the same time, the full-disk encryption solution should be set to require the supply of user credentials when booting or returning from hibernation, as described in the next section.

Turning off and rebooting a machine requires quite some time and does not preserve the current system state. On the other hand, it should become best practice to regularly reboot machines. I often see users who put their systems into standby or hibernate mode over and over again for weeks or even months without ever rebooting, thus unintentionally preventing the installation of updates which require a reboot.

Leaving an unattended system display-locked or in standby mode not only leaves it open for Firewire-based and cold boot attacks [04]. For example, every now and then, a vulnerability that is remotely exploitable is discovered. The Conficker virus exploited such a vulnerability in 2009, for example. Once the system is in a running state – for example when only the display is locked or it can be brought back from standby into running state – the system is potentially vulnerable for such attacks, until the vulnerability is fixed. If a new exploits comes out, adversaries could attack unpatched systems via the network.

8.4 BitLocker and EFS

Full-disk encryption solutions like Microsoft's BitLocker can add to overall data security. However, they must be configured to require user credentials before booting to the logon screen ("pre-boot authentication"). Once Windows is running, the full-disk encryption key is present in memory and can be stolen, as shown by [04].

As described in section 6 (page 6), Firewire-based authentication-bypass attacks on locked systems with a logged-on user enable instant access to EFS protected data. Unattended systems should thus be turned off or hibernated (with full disk encryption installed and pre-boot authentication turned on).

9 Annex

9.1 About the Author

Benjamin Böck is employed at Secure Business Austria, an industrial research center for IT-Security based in Vienna/Austria and is also an independent information security consultant. He holds several IT-related master's degrees from Vienna University of Technology where he is currently a Ph.D. student. Benjamin lectures on information security related topics at the Vienna University of Technology and several universities of applied sciences; he is also an instructor for professional penetration testing courses. He has a background in IT auditing from one of the Big Four and also explored other information security related domains such as digital forensics. His main interests are in the area of penetration testing with a focus on web applications.

9.2 About the Security Research Lab

Members of the Security Research Lab conduct security research on various subject areas of information security. It is situated in Vienna, Austria.

For more information please refer to http://www.securityresearch.at

Security Research is a strategic partner of Secure Business Austria, an industrial research center for IT-Security founded by the Vienna University of Technology, Graz University of Technology and University of Vienna. http://www.sba-research.org/

9.3 Acknowledgement

The work described in this paper has partially been supported by Secure Business Austria.

10 References

[01] **Benjamin Böck, David Huemer.** FirewireBlocker – A Software Defense against Firewire-based physical Security Attacks on Windows Systems. http://www.securityresearch.at/publications/windows_firewire_blocker.pdf (accessed August 13, 2009)

[02] **FirewireBlocker (software).** http://www.securityresearch.at/publications/firewireblocker.zip (accessed August 13, 2009)

[03] **Microsoft.** 10 Immutable Laws of Security. http://technet.microsoft.com/en-us/library/cc722487.aspx (accessed August 13, 2009).

[04] **J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten.** Lest we remember: cold-boot attacks on encryption keys. May 2009. Communications of the ACM, Volume 52 , Issue 5. http://citp.princeton.edu/pub/coldboot.pdf (accessed August 13, 2009)

[05] **Maximillian Dornseif.** 0wned by an iPod. Presented at PacSec 2004. http://md.hudora.de/presentations/Firewire/PacSec2004.pdf (accessed August 13, 2009)

[06] **Adam Boileau.** Hit by a Bus: Physical Access Attacks with Firewire. Presented at Ruxcon 2006. http://storm.net.nz/static/files/ab_Firewire_rux2k6-final.pdf (accessed August 13, 2009)

[07] **Winlockpwn (tool).** http://storm.net.nz/static/files/winlockpwn (accessed August 13, 2009)

[08] **Peter Panholzer.** Physical Security Attacks on Windows Vista. March 5, 2008. https://www.sec-consult.com/files/Vista_Physical_Attacks.pdf (accessed August 13, 2009)

[09] **Douglas MacIver.** Penetration Testing Windows Vista BitLocker Drive Encryption. Presented at Hack In The Box 2006. http://www.packetstormsecurity.org/hitb06/DAY_2_-_Douglas_MacIver_-_Pentesting_BitLocker.pdf (accessed August 13, 2009)

[10] **Niels Ferguson.** Microsoft. AES-CBC + Elephant diffuser A Disk Encryption Algorithm for Windows Vista. August 2006. http://download.microsoft.com/download/0/2/3/0238acaf-d3bf-4a6d-b3d6-0a0be4bbb36e/BitLockerCipher200608.pdf (accessed August 13, 2009)