Information Security and Computer Fraud, 2014, Vol. 2, No. 3, 48-51
Available online at http://pubs.sciepub.com/iscf/2/3/3
© Science and Education Publishing
DOI:10.12691/iscf-2-3-3

The Security File System in Smart TV Operating System

Wang Xin[1,2,*], **Yao Suying**[1]

[1]School of Electronic Information Engineering, Tianjin University, Tianjin, China
[2]Tianjin Broadcast and Television Network Co., Ltd
*Corresponding author: pyhiloveyou@126.com

Received December 02, 2014; Revised December 19, 2014; Accepted December 23, 2014

Abstract Modern smart TV devices have put a wealth of information and ever increasing opportunities for social interaction at the fingertips of users. At the center of this revolution are smart phone, smart TV and tablet computers, which give people a nearly constant connection to the Internet. As smart TV usage increases, more developers become involved in implementing application for smart TV. Security for application running on the smart TV is important. The smart TV operating system provides great flexibility not only for application developers but also for users. Most of Smart TV Operating Systems are open source operating system designed for Digital TV available on a variety of smart TV devices. As with PCs, there are a variety of security threats that can affect the smart TV device such as application based threat, web based threat, network based threat and file data threat. In this paper, we mainly focus on file data threat, in order to keep the data integrity the information should not be amended, damaged and loss in transmission process. We present the security file system in the Smart TV Operating System.

Keywords: *security file system, Smart TV, DVB, Android*

Cite This Article: Wang Xin, and Yao Suying, “The Security File System in Smart TV Operating System.” *Information Security and Computer Fraud*, vol. 2, no. 3 (2014): 48-51. doi: 10.12691/iscf-2-3-3.

1. Introduction

[12] Broadcasting has been one of the most important means to distribute information in mass media communications. A broadcasting system places information in mass media; any device connected to the medium can receive the information.In the promotion of the policy of three network convergence and the technology of digitization, intelligentize and network, the broadcast and television network is evolved into a comprehensive information communication network. It is not only a one way information network but also a bidirectional transmission network, even more can access the internet. The smart TV terminal is also evolved from one way broadcast terminal to DVB+OTT bidirectional intelligent terminal.

[13] DVB+OTT is a next generation television capable of transmitting, receiving and displaying a video stream. It provides access to on-demand gaming, home security, data services and digital music. DVB+OTT is capable of providing a single stream to multiple users simultaneously and also to a single stream such as Video On-Demand.

[10] Modern smart TV devices have put a wealth of information and ever increasing opportunities for social interaction at the fingertips of users. At the center of this revolution are smart phone, smart TV and tablet computers, which give people a nearly constant connection to the Internet. Applications running on these devices provides users with a wide range of functionality, but vulnerabilities and exploits in their software stacks pose a real threat to the security and privacy of modern smart TV system. The information and data security is becoming more and more complex, more and more important in the broadcast network.

[14] Providing security at the file system level is part of the basic security functionality provided by virtually all general-purpose operating systems; however, the limitations of file system protection only via data structures maintained by the operating system have long been obvious.

In this paper, we present the security file system of the Smart TV Operating System, in order to protect the security and sensitive data and avoid the security and sensitive information leakage.

2. The Security File System Overview

The Smart TV device allows users to store information, data and applications to the storage device. If we do not protect, the copyright product store in the external device will be malicious damaged. It will strongly damage the interest of broadcast and television system. Therefore, in order to protect the copyright of the data we can use the encrypt mechanism of Smart TV Operating System. The encrypted files can only be decrypted by the encrypted Smart TV Operating System. e.g. encrypt the audio and video during the process of recording audio and video, encrypt the file when the file is writing into the SD card.

The Smart TV Operating System provide security file system strategy in order to protect the user data not be malicious leakage. The Smart TV Operating System also provide security file system umount mechanism to provide unauthorized instruction code execution.

The security file system provide the following functions.

A. The File Handle Random Distribution Function

In the I/O file, when reading data from a file, the application must call an operating system function and transfer the file name and select a path to the file to open the file. This function takes a sequence number, called file handles, the file handle is the only basis for identification to open file. To read a block of data from a file, the application needs to call the function ReadFile, and transmit the address of the file handle in the memory and the number of bytes need to be copied to the operating system. When the task is finished, then call the system function to close the file. In the Smart TV Operating System, the system is adopted file handle random distribution strategy.

B. Complete Unmount Function

Through the analysis of normal umount process we can know that it is decided to do what based on the reference count of file system. When the reference count is greater than 2, it will return busy error code directly without any further process. When the reference count equals 2, then umount the file system from the file system tree. Therefore, we adopt the method as following, first set a marker(MNT_FUMOUT) to the file system's mnt_flags to prevent the outside access the file system and return an error code EN-XIO, secondly set FMODE_FUMOUNT to all the open file and then close all files. Write the file content to the device and releasing the document lock and memory map.

C. Signiture and Authentication to the Kernel Module

The kernel module code has the same priority with the kernel in the process of operation. If it is used by the invasive procedure, it will bring serious security hazards. In order to prevent the invasion to the kernel module, it is needed to signature and verification to the kernel module code to avoid to provide an opportunity that can be made use of by invaded code to its advantage.

We use the signature verification algorithm which isin full compliance with the PKCS (The Public-Key Cryptography Standards) series standards and is compatible with the certificate in X509 format, on the basis of RSA asymmetric key system to realize signature authentication to the module file code.

(1). Original RSA

[15] Since the mid-1970s, when public-key cryptography was first developed,the RSA Cryptosystem has become the most popular cryptosystem in the world.

[15] One of the reasons that RSA is so popular is its simplicity. Both encryption and decryption require only one modular exponentiation. However, computing an exponentiation modulo N is very costly because the RSA modulus is much larger than other moduli of public key cryptosystems such as those based on elliptic curves. The other main disadvantage of using RSA is the size of the key pairs.

[15] The original RSA cryptosystem consists of three algorithms: key generation, encryption, and decryption. Below, we describe each algorithm from the original description of RSA, sometimes called the textbook or simplified version of RSA, and then discuss some simple variants of the original presentation. In practice, an appropriate padding scheme, such as OAEP [2], is required to ensure the security of the cryptosystem.

Key Generation: Let N = pq be the product of two randomly chosen large prime numbers p and q that are distinct. Let e be a randomly chosen integer that is relatively prime to $\phi(N) = (p - 1)(q - 1)$, where $\phi(.)$ is Euler's phi function, and let d be its multiplicative inverse modulo $\varnothing$ (N). The pair (e, N) is the public key and the pair (d, N) is the private key.

Encryption: A plaintext message M $\in$ N is encrypted by raising itto the eth power modulo N. The result, $C = M^e$ mod N $\in Z_n$, iscalled the ciphertext of M. All of the different variants of RSA in this correspondence use this method for encryption.

Decryption: A ciphertextC $\in Z_n$, for a given plaintext messageM $\in Z_n$, is decrypted by raising it to the dth power modulo N. From Lagrange's theorem, it follows that

$$C^{d}\text{mod } N = M_{ed}\text{mod } N = M \text{ mod } N = M$$

The integer N is called the RSA modulus or simply the modulus. The integer e is called the public (or encryption) exponent and the integer d is called the private (or decryption) exponent. When computed in the manner described above, the private exponent d will, with high probability, be roughly the same order of magnitude as $\varnothing$ (N). The public and private exponents are defined so that ed $\equiv$ 1 (mod $\varnothing$ (N)). We call this the RSA key relation, or simply the key relation. From the key relation, it follows that there exists a unique positive integer k satisfying

$$\text{ed} = 1 + k\varnothing(N)$$

We call this the RSA key equation or simply the key equation

(2). The theory of Signiture and Authentication

Let the data to be signature is equal to m, its digital abstract is H.

$$\text{h} = \text{Hash}(\text{m})$$

Among them, Hash is the one-way hash algorithms, such as MD5, SHA-1.

Let p, q, d are the private data of signer, they all included in the private key SK; n, e are the public data of the signer, they are all included in the signer's public key PK, these data meet the following requirements:

n = pq, where p $\neq$ q, q, p are the big prime number.

e,d $\in$ RZn, and e=d-1, ed≡lmod(n), here

$$(\text{n}) = (\text{p}-1)(\text{q}-1)$$

Then use the signer's private key to encrypt h can get the signature value s.

$$\text{s} = \text{E}(\text{x}) = \text{hdmodn}$$

(3) Authentication

Let the data to be authenticated is m′, its digital abstract is h′

$$\text{h}' = \text{Hash}(\text{m}')$$

Assumption that has obtained the real public key of the signer, then use the open data e of the PK to decrypt and calculate, obtain the restored digital abstract h′. Here h' is equivalent to h.

$$\text{h}'' = \text{D}(\text{s}) = \text{se mod n}$$

Comparing h′′ with h′, if they are not same, then the authentication is failure.

D. Data Encryption and Decryption

(1) Implementation of the encryption process

By replacing the function address of call table address in the kernel system to realize the redirection of system call, such asthe original reading (sys_read) and write (sys_write) system call is replaced with decryption function read (hacked_read) andDensity functional (hacke_write) function.

In the course of execution, the encryption module first find the sys_writein system call table and save the original system call.

The content of system call table is replaced with the encryption function of hacked_write.

To judge whether the file requires to be encrypted bySmart TV Operating System, if it is, copy the data to the kernel space, and then call a corresponding algorithm for data encryption, after encryption copied the encrypted data to the user space; if else not skip.

Finally call the write function and write data in it.

(2) Implementation of the decryption process

The process call the sys_read through the system, Implementation the replacement of hacked_read. Use the original functionsys_read to read data and judging whether the file is needed to decrypt. Copy the data to the kernel and decrypt the data and copy the data to the user space, then return.

E. File Integrity Check

This module provides file integrity checking function and access interface, so as to ensure the integrity protection of the security of sensitive data, finding the security and sensitive data is tampered promptly.

3. The Technical Scheme of Security File System

In this part we analysis the technical scheme of security file system. The S of security file system is included the file handle random distribution, complete unmounts, signature and authentication to the kernel module, data encryption and decryption, file integrity check.

A. The Technical Realization of File Handle Random Distribution Function

In Smart TV Operating System, the system convert the file handle distribution strategy and adopted the file handle random distribution strategy.

In the file handle random distribution strategy, the kernel is no longer adopted the manner of plus 1 simply. The kernel adopted the manner of plus a random value. After accessing the fdtable table, searching the random file handle value in fdtable. If the random file handle value is not used, then use this file handle and feedback the upper application to use.

B. The Technical Realization of Complete umount function

(1) First of all need to judge, the file system to be uninstalled is not the root file system of unmount process. Whether there is a mandatory option in umount. Retval is not equal to 0, because the retval is equal to 0 represent that the file system has been uninstalled successfully already. The mnt_flags mark of file system has not been set MNT_FUMOUNT. If MNT_FUMOUNT has already been set mark, it means that it has already carried out the process following if. The mnt_mounts of the file system is empty. When the file system is being mounted, it usually add file system to the mnt_mounts of the father file system. Here the judgment is empty represented this file system has no sub file system.

(2) MNT_FUMOUNT has been set to mnt_flags. After being set this mark, no process can open files of this file system any longer.

(3) Callfs_fumount_mark_files () function, set FMODE_FUMOUNT mark to the open file of the file system.

(4) Uninstall the file system from the file system tree.

C. The Technical Realization of Signature and Authentication to the Kernel Module

Signing the module file must be completed through the signature tool. It has nothing to do with the operating system and other platform.

The signature information location:

The format of module file is ELF file, adding signature information to the end of the ELF file.

The signature information includes: version identifier, the size of original file, the identification ID of the signature’s public key, signature algorithm LOGO, the signature time, the basic information of the signer.

The validation process:

When the user request to insert to the module, it is needed to judge whether there has signature information. If hasn’t then feedback failure. If has the signature information, then verify the signature information. If the verification passed, then allowing the module execute insert action, else forbidding the insert action.

D. The Technical Realization of Data Encryption and Decryption

In the course of execution, the encryption module first find the system call table in sys_write, save the original system call; judge whether the file requires encrypted files to Smart TV Operating System, if it is copy the data to the kernel and calla corresponding algorithm for data encryption, else jump. Finally call the write function and write data in it.

In the course of execution, the decryption module first find the system call table in sys_read, save the original system call; judge whether the file requires decrypted files to Smart TV Operating System, if it is copy the data to the kernel and calla corresponding algorithm for data decryption, else jump. Finally call the write function and write data in it.

E. The Technical Realization of File Integrity Check

Configure the check value and file name of integrity-check.conf. The check value is used as reference. Sending the file name which is needed to be checked to the kernel. The kernel module can generate a check value of current file automatically, and comparing with the storage value of configuration file, if successfully then return ‘prompt file has not changed’, else return ‘file has changed’.

4. Conclusion

[10] Modern smart TV devices have put a wealth of information and ever increasing opportunities for social interaction at the fingertips of users. At the center of his revolution are smart phone, smart TV and tablet computers, which give people a nearly constant connection to the

Internet. Applications running on these devices provides users with a wide range of functionality, but vulnerabilities and exploits in their software stacks pose a real threat to the security and privacy of modern smart TV system. The information and data security is becoming more and more complex, more and more important in the broadcast network.

In this paper, we presents the security management and control scheme of Smart TV Operating System (TeleVision Operating System) for DTV (Digital TeleVision). We designed to provide high assurance security mechanisms for security management and control policies, and a verified implementation of the mechanisms which the Smart TV Operating System uses to enforce the security management and control policies.

References

[1] Wang Xin, Chen Delin, Sun Yue, Wang Xu, Yao Suying, "Security control for TVOS", 2014 2nd Asian Pacific Conference on Mechatronics and Control Engineering.

[2] Wang Xin, Yao Suying, Wang Xu, Chen Delin, Sun Jun, Hua Zhong, Wang Dongfei, "TVOS Content Security Analysis and Protection in Smart TV", 2014 2nd International Conference on Mechatronics and Control Engineering.

[3] Wang Xin, Chen Delin, Sun Yue, Wang Xu, Yao Suying, "The Security Model of Broadcast Illtelligent Terminal Application and Technology Realization on TVOS", 2014 2nd International Conference on Mechatronics and Control Engineering.

[4] Felix Rohrer, Nebiyu Feleke, Yuting Zhang, Kenneth Nimley,Lou Chitkushev, Tanya Zlateva1. Android Security Analysis and Protectionin Finance and Healthcare. //Boston University MET

[5] ChenDelin, Li Zheng, Wang Ying, Zhao Liangfu, Zhang Dingjing. "The main technical characteristics and software architecture of NGB TVOS", "Radio and television information", 2013-10National Academy of Broadcasting Science

[6] NGB TVOS1.0 Version of cooperation and development. The next generation of radio and television (NGB) Intelligent Technology TV operating system v1.0.0, the State Press and Publication Administration

[7] GY/T267-2012.NGB Technical specification of terminal middleware, the State Press and Publication Administration

[8] Wang Mingmin, Zhu Yunbin. "To explore the implementation model and techniques of intelligent television terminal security under NGB environment", "Radio and TV Technology", 2012-10.

[9] Ning Hua, Li Wei, Wang Kun, Lei Mingyu. "Research on intelligent terminal security system", "Modern telecommunication technology", 2012-5.

[10] Haohui Mai, Edgar Pek, Hui Xue, Samuel T. King, P.Madhusudan. Verifying Security Invariants in ExpressOS // University of Illinois at Urbana-Champaign

[11] Bernhard J. Berger, Michaela Bunke, and Karsten Sohr. An Android Security Case Study with Bauhaus // 2011 18th Working Conference on Reverse Engineering.

[12] Zhiguo Wan, Jun'e Liu, Rui Zhang, and Robert H. Deng. "A Collusion-Resistant Conditional Access System for Flexible-Pay-Per-Channel Pay-TV Broadcasting" // IEEE TRANSACTIONS ON MULTIMEDIA VOL. 15，NO. 6, OCTOBER 2013

[13] Eun-Jun Yoon, Kee-Young Yoo. "ECC-based Key Exchange Protocol for IPTV Service" // ICIC 2011

[14] Stephen D. WolthusenFraunhofer-IGD. "Security Policy Enforcement at the File System Level in the Windows NT Operating System Family".

[15] Hung-Min Sun, Mu-En Wu, Wei-Chi Ting, and M. Jason Hinek. "Dual RSA and Its Security Analysis" //IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 53, NO. 8, AUGUST 2007.