2011 International Conference on Computer Science and Information Technology (ICCSIT 2011)
IPCSIT vol. 51 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V51. 109

Research on Cisco IOS Security Mechanisms

Xiaoyan Su[a,*], Dongying Wu[a], Da Xiao[a], Yuxiang Han[a],

[a] Zhengzhou Institute of Information Science and Technology, Zhengzhou, 450002, China

Abstract. As Cisco Systems' routing and switching equipment occupy a large market share, so their security is critical for network. Firstly, this paper analyzes the security mechanisms of Cisco IOS, such as the integrity check mechanism, the memory protection mechanism, the authentication mechanism, the access control mechanism and so on. And then, research the vulnerability of these security mechanisms when resisting attack. At last, in order to better protect the security of Cisco IOS, some advices about how to strengthen the anti-attack capability of Cisco IOS are given.

Keywords: network security, Cisco IOS, security mechanism, vulnerability

1. Introduction

Routers and switches are important network infrastructure. So the security of these equipments is an important guarantee for network security. Cisco Systems' routers and switches gear are currently the most widely used network equipment. Cisco IOS (Internetwork Operating System) is the software used on the vast majority of Cisco network equipment. At present, almost all Cisco routers and switches are running Cisco IOS. With in-depth study of Cisco IOS, new attack technologies against Cisco IOS are constantly emerging, and pose a great threat to network equipments and networks. Therefore, research on Cisco IOS security mechanisms is of great important to equipments and network security.

Currently, researches on Cisco IOS security mechanisms are not many. The earliest research of Cisco IOS security is from FX. He analyzes much vulnerability of Cisco IOS [1] [2], introduces shellcode and binary exploitation techniques [3] [4], and analyzes the challenges with the explotion of memory corruption software vulnerabilities in Cisco IOS [5]. In [6], Michael Lynn describes the method of bypassing Cisco IOS memory check, and run a “shellcode” program on a router without authority. Lynn's work is also focus on shellcode and exploitation techiniques. Muniz and Arial Futoransky do much research on Cisco IOS rootkit, they analyze vulnerability in IOS image file and described their techniques for IOS rootkit [7] [8]. In [9], Gyan Chawdhary and Varun Uppal introduce shellcode techniques for Cisco IOS. Gyan Chawdhary explains the way of bypassing Check heaps by modifying one part of the router's memory [10].

The research above is from the attacking aspect, mainly analyzes the vulnerability of the IOS management mechanism. At present, there is little comprehensive study on the IOS security mechanism. Not yet form an improved research system. This paper will introduce the Cisco IOS security mechanisms comprehensively, and analyze its vulnerability when resisting attack.

2. Integrity check mechanism

In order to protect the integrity and security of its products, Cisco added integrity check mechanism in the IOS image file. This approach makes the attacker can not tamper with the IOS image file or embed malicious code in the file easily, thus ensuring network security. Image integrity check is the first security barrier to protect the Cisco IOS.

* Corresponding author.
E-mail address: xysu11@126.com.

2.1. Image file structure

Cisco IOS is a monolithic system, A software system is called "monolithic" if it has a monolithic architecture, in which functionally distinguishable aspects (for example data input and output, data processing, error handling, and the user interface), are not architecturally separate components but are all interwoven. IOS image file contains a lot of information which is used for system debugging. This information makes the IOS image become very large. In order to save storage space, Cisco has used a specific compression method to compress the IOS image. At start-up phase, the equipment first extracts IOS image through the self-extracting code, and then loads the decompressed IOS image.

2.2. Integrity check mechanism

IOS file is the ELF file structure. The compressed IOS image is in a separate section of the IOS file. There is a header structure in the beginning of the section [7], here we call it Header. Header contains the information that is used for integrity check. The compressed IOS image is after the Header. Header's structure can be shown as figure 1.

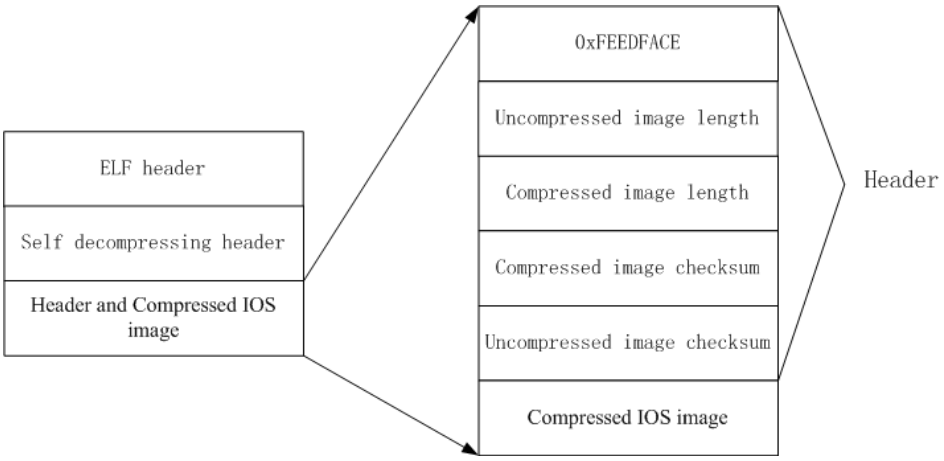

Fig. 1. Structure of an IOS image file and it's Header

The contents of Header are a Magic Number “0xFEEDFACE”, uncompressed image length, compressed image length, compressed image checksum, and uncompressed image checksum. Checksum values of image are calculated through the contents of the compressed and uncompressed IOS image. Cisco calculates the Length values and checksum values of compressed image and uncompressed image, and saved them in the Header. By this header structure, IOS can protect an image's Integrity and security. Contents of a Header taken from a 2600 router image as shown in figure 2.

```
00004280  47 49 43 24 24 0A 43 57  5F 45 4E 44 24 2D 67 73
00004290  2D 69 2D 6D 7A 24 0A 00  FE ED FA CE 00 E7 70 84
000042A0  00 59 01 22 7C E6 3A 15  D6 69 00 EC 50 4B 03 04
```

Fig. 2. Header of an 2600 router image

Once the IOS image file is tampered, no matter how many modifications, it will result in the change of the checksum values and the compressed image length value, which can not pass the integrity check. Therefore, the integrity check mechanism can effectively protect the integrity of Cisco IOS image, and improve the system security.

The integrity check mechanism plays an important role in protecting the Cisco IOS. But the integrity check algorithm is not secure, and the information of the header structure is also easy to get. In [7], Muniz has broken the integrity check mechanism by the weakness of the integrity check algorithm. Finally he has tampered with the IOS image and embedded rootkit code successfully.

3. Memory protection mechanisms

3.1. Memory Management

For the operating system, memory management is an important component. Cisco IOS maps the entire physical memory into a large flat virtual address space. IOS divides the address space into different areas called regions, which mostly correspond to the various types of physical memory, and regions also can be nested in a parent-child relationship [11]. IOS command 'show region' can be used to display regions defined on a particular system as demonstrated in figure 3 (taken from a Cisco 2600 router).

```
router#show region
Region Manager:

      Start         End      Size(b)  Class  Media  Name
 0x01B00000  0x01FFFFFF     5242880  Iomem  R/W    iomem
 0x60000000  0x60FFFFFF    16777216  Flash  R/O    flash
 0x80000000  0x81AFFFFF    28311552  Local  R/W    main
 0x80008074  0x80A2C2AF    10633788  IText  R/O    main:text
 0x80A2C2B0  0x80E7EE6B     4533180  IData  R/W    main:data
 0x80E7EE6C  0x81042167     1848060  IBss   R/W    main:bss
 0x81042168  0x81AFFFFF    11263640  Local  R/W    main:heap
```

Fig. 3. Output of the command "show region" from an 2600 router

Figure 4 illustrates this memory map and its regions.

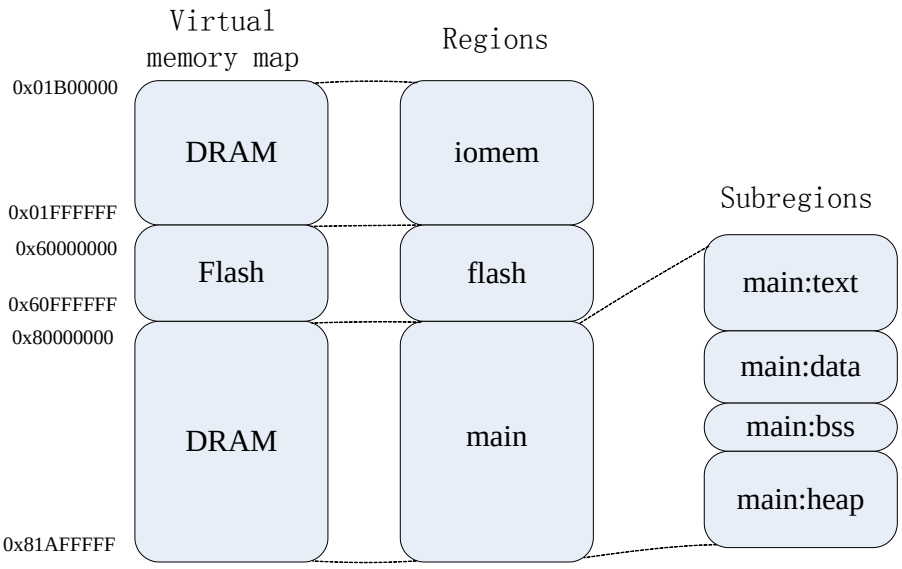

Fig. 4. memory map of an 2600 router

Every process of IOS has its own stack, which is an allocated heap block. The heap of IOS is shared by all processes. In Cisco IOS, memory blocks of all processes are linked together one after another by a doubly linked list. This is visible when inspecting the memory allocation on a router. You can obtain information of IOS blocks by using the 'show memory' command as demonstrated by figure 5.

```
          Processor memory

 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
81042168 0000001500 00000000 81042770 001  -------- -------- 803D0DCC  List Elem
ents
81042770 0000005000 81042168 81043B24 001  -------- -------- 803D0E08  List Head
ers
81043B24 0000009000 81042770 81045E78 001  -------- -------- 803EC3E0  Interrupt
 Stack
81045E78 0000000044 81043B24 81045ED0 001  -------- -------- 80A279E8  *Init*
81045ED0 0000000092 81045E78 81045F58 001  -------- -------- 807F9C9C  Init
81045F58 0000000208 81045ED0 81046054 001  -------- -------- 803E690C  *Init*
81046054 0000004248 81045F58 81047118 001  -------- -------- 803305D4  TTY data
81047118 0000002000 81046054 81047914 001  -------- -------- 803339D4  TTY Input
```

Fig. 5. Part of memory block taken from an 2600 router

Memory blocks for entirely different tasks are following each other as shown in figure 5. And the entire heap of IOS is one big doubly linked list. In the heap, every block has a header structure for management. The relationship of blocks can be shown as figure 6.

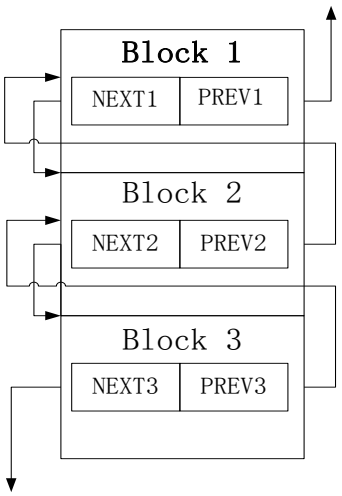

Fig. 6. Relationship of memory blocks

Memory block's header can be defined as shown in figure 7.

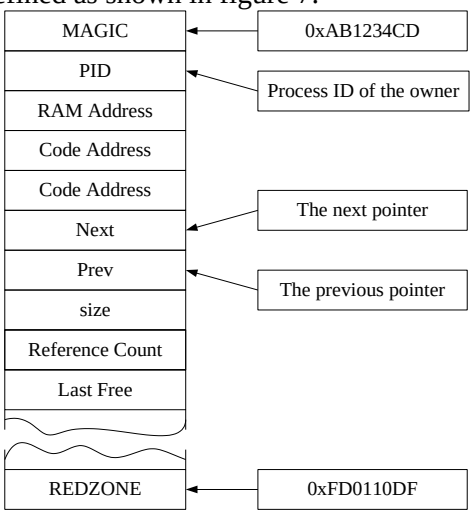

Fig. 7. Structure of an block header

3.2. Memory protection mechanisms

3.2.1. Address space protection

In the space of IOS virtual address, the address space is not continuous, and there are intervals. As shown in figure 3 and figure 4, you can see that the Iomem's address is from 0x01B00000 to 0x01FFFFFF, and the Flash's address start from 0x60000000. There is an address interval from 0x02000000 to 0x5FFFFFFF.

Intervals in IOS virtual address space are not system errors. These address intervals are critical for system security. There are two main effects of these intervals. First, the address interval facilitates the expansion of regions. When a region needs to expand the address space, it can be extended directly on the base of the original address, and other regions' address space will not be impacted. Second, intervals of these virtual addresses will prevent damage to memory caused by thread errors. If a running thread over its address space in the region and reached an address interval between the regions, an address error occurs and it will be forced to stop.

3.2.2.

Cisco IOS doesn't employ a full virtual memory scheme. To reduce overhead, Cisco IOS do not take any effective mechanism to isolate processes' address space. Thus, though each process has its own memory space, but other processes can also access other's memory space. It means that all processes share the same memory space, and can arbitrarily change the data in memory [10]. This makes the software vulnerabilities debugging and memory leak vulnerability detection becomes very difficult.

For security reasons, a process named Check heaps is used to overcome the weakness of IOS memory management. Check heaps walks the heap lists on a regular basis and verifies the integrity of them. If an error is found, Check heaps will force the device to reboot in order to protect the security of the system. Check heaps performs roughly the following checks [4]:

- Verify that the MAGIC value is “0XAB1234CD”
- If the block is in use, verify that the REDZONE value is “0xFD0110DF”
- Verify that the Prev is not NULL
- Verify that the Prev pointer's Next pointer points to this block
- If the Next is not NULL, verify it points exactly behind the REDZONE field of this block
- If the Next is not NULL, verify that the block it points to has a Prev ptr point back to this block
- If the Next is NULL, verify that it does end on a memory boundary.

IOS sets a Boolean variable crashing_already. The initial value of crashing_already is “0”. Once Check heaps found exceptions of memory block, it will first check the value of crashing_already, if the value of rashing_already is a non-positive value, Check heaps will change the value of crashing_already to a positive value, and reboot the system. However, if the value of crashing_already is a positive value, Check heaps will simply return to the caller without performing any operation [10]. The crashing_already can prevent unpredictable errors that generated by two current processes from crashing simultaneously. Check heaps can also provide useful information to administrators; this information can help tracking reasons of IOS memory leaks and buffer overflow.

By examining the memory block, Check heaps has the ability to detect heap overflow. However, there is still weakness in Check heaps, so that an attacker can use specially crafted data to bypass it. FX successfully deceived Check heaps process with fake memory block header, and achieved the purpose of embedding arbitrary code and executing it [4]. Though crashing_already can be used to protect IOS, its weaknesses also pose a threat to the security of the system. Gyan Chawdhary introduced the method of preventing system to reboot by change the value of crashing_already to a positive value [10]. When the Check heaps detects a memory corruption, it tries to reboot the device. But as crashing_already has been changed to a positive value, Check heaps would simply return to the caller.

4. Authentication mechanism

Authentication mechanism is the most basic security mechanism of all operating system. In complex networks with a large number of network devices, a centralized authentication server should be deployed for authentication. But in fact, most of Cisco Systems' device running IOS stores a configuration file in its nonvolatile memory (NVRAM), and the configuration file contains all user names and passwords. Cisco IOS

uses two types of passwords: user password and privilege password (enable password). User password is used for user to login authentication, and Privilege password is used to enter privilege mode.

In the configuration file, the password can be stored in 3 types: plain text password, type 7 password and type 5 password. Type 7 password and type 5 password are cipher text passwords. Type 7 password's encryption algorithm is designed by Cisco Systems, and this encryption algorithm is reversible. Type 5 password is encrypted by salt and MD5 algorithm which is a one-way hash algorithm and almost impossible to illegal crack. Type 5 password provides a strong protection for sensitive information. Type 7 password can be used for user password and privilege password, and type 5 password can only be used for privilege password. Administrators are allow to choose the form of a password that is stored in configuration file, either in plain text form, or in cipher text form.

An encrypted password makes the password can not be obtained easily, which prevents malicious operation from attackers. However, the configuration file, we often find that privilege password are stored both in the form of type 5 and type 7， as shown in figure 8, or both in the form of type 5 and plain text, as shown in figure 9. Priority of type 5 is higher than type 7 and plain text, which lead to the invalidation of the latter.

```
hostname boy
!
enable secret 5 $1$h5fW$TU.FfqnX56Sb0c/GKuF1z.
enable password 7 135445415F5952
```

Fig. 8. Fragment of an configuration file

```
hostname boy
!
enable secret 5 $1$03J7$GghbTWT8mKbOUgngaTnK/.
enable password 123456
```

Fig. 9. Fragment of an configuration file

Administrators of Cisco Systems' devices may change the type of password for some reason. For example, IOS can re-encrypt the original plain text password or type 7 password with MD5. Although the encryption method changed, in the configuration file, plain text or type 7 password will not be automatically removed, As shown in figure 8 and figure 9.The existence of this defect, allow an attack to easily gain access to the system by cracking a type 7 password or read a plain text password from the configuration file. Therefore, for security reasons, administrators should try not to use plain text password, and privilege password show be stored in the form of type 5. Privilege password in the form of type 7 and plain text should be removed from the configuration file, to prevent attackers' stealing and cracking it.

5. Access control mechanism

In most multi-user operating systems (such as Linux, UNIX, etc.), the administrator has privileges, the user does not have privileges. A process or have privileges (super-user process), or does not have any privilege (non-root user process). Such access control mechanism brings convenience to management and maintenance of the system, but it is not conducive to system security.

For Cisco equipments security, Cisco IOS adopts principle of least privilege (POLP) to manage user's privileges. POLP requires that in a particular abstraction layer a computing environment, every module (such as a process, a user or a program on the basis of the layer we are considering) must be able to access only such information and resources that are necessary for its legitimate purpose. Such as super-user privileges are divided into a set of fine-grained privileges and are granted to different system user / administrator. This strategy give the system user / administrator only the required permissions to complete its work, Thereby reducing the loss due to malicious action, privileged user's password is lost or caused by misuse.

Cisco IOS provides the user with level 0 to level 15 total of 16 different privilege levels, to restrict user's access to the system. In Cisco IOS, the higher the level of the user's privileges, the more operation can be carried out. In all 16 levels, level 1 and level 15 are the most commonly used. In the default configuration, log

on Cisco routers, the user will enter the privilege level 1. At this level, the user can view the router information, such as interface status, but can not make any changes or view the running configuration file. Enter the ‘enable’ command and verify the password, if the password is verified successfully, the user would enter the privilege level 15. In Cisco IOS, level 15 is equivalent to UNIX's root authority or the administrator rights of Windows, so you can fully control the system.

For the principle of POLP, in the case of several operators, Cisco IOS will give different privilege to different operator base on their management authority. In Cisco IOS, there are two ways to define user's privilege level: One is to set a specific password for each privilege level, when log in, users first enter the level 1, and then they can use appropriate password to enter other corresponding privilege levels. For example, command ‘enable secret level 5 cisco5’ will set “cisco5” as level 5's password. Another is defining user's privilege when the user is defined, with its user name and password, the user can enter its privilege level directly. For example, with command "username anna privilege 5 password cisco5”, user anna's privilege will be set at level 5 and its password is “cisco5”. Cisco IOS can also set its command can only be used in a special level, other levels that are lower than this level can not use this command.

Cisco IOS control user's access to the system through different privileges. The POLP strategy provides effective access control restriction for Cisco IOS, it limits user's misuse and malicious operation, and prevent the possible threat posed to the system, and protect the security of the Cisco Systems' device.

6. Others

6.1. Exception handling mechanism

Windows, Linux and other operating systems have perfect exception handling mechanism, when the system exception occurs, these systems allow the exception handler for exception processing, and allow part functions of the system to be rebooted. Cisco IOS has no complicated exception handling mechanism, and part of system functions reboot is forbidden. The only exception handling mechanism of Cisco IOS is to reboot the system completely, because an exception of system is probably a mis-operation of a process, and this mis-operation may have written some data in the writable memory section. Therefore the only safe handling of the exception is rebooting the system completely.

This exception handling mechanism does nothing except to force the system to reboot directly. This not only simplifies the Cisco IOS exception handling, but also a system security protection mechanism. In the Windows operating system, use CPU exception handling to trigger malicious code execution is already a very mature technology, but in Cisco IOS, this approach is not feasible, because any CPU exception will result in a system reboot, resulting in failure of the attack.

6.2. Third-party software is forbidden

Cisco IOS does not allow any third-party software running in the system. And Cisco Systems' devices are not allowed to update the IOS image stored in the device in the way of online patching. Cisco IOS image must be updated in the way of replace the full IOS image. The restrictions on the software running, and the way of updating IOS image provides a good security protection, and reduce the possibility of effective malicious attacks.

7. Conclusion

Cisco IOS security mechanisms plays a significant role in protecting the security of Cisco IOS. However, these security mechanisms, there are still weaknesses that allow attackers to take advantage of, resulting in threats to the security of the system, thus threatening the security of the network. Doing research on Cisco IOS security mechanism will enable us to have an in-depth understanding of Cisco IOS security, and know more about security vulnerabilities of Cisco IOS. Mastered these, we will be able to predict where the attacker might have to attack, and what methods can be used to attack, so that we can effectively prevent malicious attacks, and protect the security of networks.

At present, researches on Cisco IOS mechanisms are all one-sided and crude, There is not an improved research system. In the next step, we are going to do more comprehensive and in-depth research on Cisco IOS mechanisms. We will analyze possible methods that may be used to attack Cisco IOS by malicious attackers, and put forward a sound response strategy to protect network security.

8. Acknowledgements

The authors would like to thank our many colleagues and the anonymous referees that pointed us to related work and helped us improve the presentation of the material.

9. References

[1] Felix 'FX' Lindner, “Cisco Vulnerabilities--Yesterday,Today and Tomorrow”, 2003, http://www.blackhat.com/presentations/bh-federal-03/bh-fed-03-fx.pdf.

[2] Felix 'FX' Lindner, “More vulnerable embedded system”, 2003, http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-FX.pdf.

[3] Felix 'FX' Lindner, “Cisco IOS—Attack&Defence,the State of Art”, 2008, http://www.phenoelit-us.org/stuff/FX_Phenoelit_25c3_Cisco_IOS.pdf.

[4] Felix 'FX' Lindner, “Burning the Bridge: Cisco IOS Exploits”, Phrack Magazine, 2007, http://www.phrack.org/issues.html?issue=60&id=7#article.

[5] Felix 'FX' Lindner, “Cisco IOS Router Exploitation”, 2009, http://www.blackhat.com/presentations/bh-usa-09/LINDNER/BHUSA09-Lindner-RouterExploit-PAPER.pdf.

[6] Micheal Lynn, “The Holy Grail:Cisco IOS shellcode And Exploitation Techniques”, 2005 http://securityvulns.com/files/lynn-cisco.pdf.

[7] Sebastian 'topo' Muniz, “Killing the myth of Cisco IOS Rootkit: DIK (Da Ios rootKit)”, 2008, http://eusecwest.com/esw08/esw08-muniz.pdf.

[8] Ariel Futoransky, “Viral Infection in Cisco IOS”, 2008, http://www.coresecurity.com/files/attachments/blackhat2008_cisco.pdf.

[9] Gyan Chawdhary and Varun Uppal, “Cisco IOS Shellcodes/Backdoors”, 2008, http://www.blackhat.com/presentations/bh-usa-08/Chawdhary_Uppal/BH_US_08_Chawdhary_Uppal_Cisco_IOS_Shellcodes.pdf.

[10] Gyan Chawdhary, “IOS Exploitation Techniques”, 2007, http://www.irmplc.com/downloads/whitepapers/Cisco_IOS_Exploitation_Techniques.pdf.

[11] BOLLAPRAGADA V, WHITE R, and MURPHY C, “Inside Cisco IOS Software Architecture”, Cisco Press, 2000.

[12] Felix 'FX' Lindner, “Developments in Cisco IOS Forensics”, 2008, http://www.blackhat.com/presentations/bh-usa-08/Lindner/BH_US_08_Lindner_Developments_in_IOS_Forensics.pdf.