УДК 539.2:548

С.В. Тищенко

ЕКСПЛУАТУВАННЯ ROOT-ВРАЗЛИВОСТІ У РОУТЕРАХ ВІД КОМПАНІЇ ASUS

Анотація. Виявлено критичну вразливість у програмному забезпеченні роутерів компанії Asus. Визначено умови успішного експлуатування вразливості. Визначено параметри та можливості, які надає вразливість.

Ключові слова: роутери, маршрутизатори, вразливість, експлойт, infosvr.

Вступ. На початку 2015 року компанія Asus усунула критичну вразливість у програмному забезпеченні своїх роутерів. Вразливість давала можливість неавторизованого виконання будь-яких системних команд з привілеями суперкористувача. Компанія Asus випустила оновлення програмного забезпечення, у якому цю вразливість вже усунуто, але все ще багато користувачів працюють із вразливими пристроями.

Основна частина. Вразливість виявлено в службі під назвою *infosvr*. Ця служба відповідає за взаємодію з програмним забезпеченням для операційної системи Windows, яке допомагає користувачам налаштовувати роутер. Служба *infosvr* чекає на вхідні UDP пакети на порті під номером 9999. Коли приходить певний пакет, вона аналізує його тип та код операції і виконує відповідну дію. Наприклад, якщо код операції NET_CMD_ID_GETINFO (*0x1F*), то служба збирає інформацію про мережеві налаштування роутера та відправляє цю інформацію на той же порт іншим пристроям (широкомовний пакет). Але є інший код операції - NET_CMD_ID_MANU_CMD (*0x33*). Якщо служба виявляє цей код, то вона зчитує текстову команду, яка міститься у тілі того ж пакету та змушує систему виконати її. Система виконує команду з найвищими правами, адже сама служба *infosvr* запущена від імені суперкористувача.

 ISSN 1562-9945

Така ситуація стала можливою завдяки помилці у програмному коді служби *infosvr*. Була помилково використана функція *memcpy* замість *memcmp*. Функція *memcmp* мала перевірити поле заголовку пакета на відповідність вмісту аналогічного поля в самому роутері. У разі невідповідності вмісту цих полів – обробка пакету припиняється. Блок коду з помилкою:

```
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);
}
```

У другій умові замість порівняння комірок пам'яті відбувається їх копіювання – `if (memcpy(phdr_ex->MacAddress, mac, 6)==0)`. Після копіювання функція *memcpy* повертає адресу призначення, яка не буде дорівнювати нулю. Тобто умова ніколи не виконається, а отже, обробка таких пакетів буде продовжуватися. Імовірно там мало би бути `if (memcmp(phdr_ex->MacAddress, mac, 6)!=0)`. Але навіть за таких обставин цю перевірку можна було б подолати записавши в поле пакета MAC-адресу роутера. Оце й уся аутентифікація. Далі команда копіюється із поля даних пакету в заздалегідь підготовлені буфери, форматується та виконується (показано лише найбільш значущі рядки, справжній код складніший):

```
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);
        }
```

Якщо перейти на офіційний сайт компанії за посиланням [1] та вибрати будь-який роутер, а потім перейти на сторінку завантаження прошивок для цього роутера, то ймовірно, що в описі прошивки за січень 2015 року буде рядок «Fixed infosvr security issue». Це означає, що вразливим був чи не кожен роутер.

Щоб змусити роутер виконати потрібну нам команду необхідно сформувати UDP пакет із полем даних такої структури:

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

Поле ServiceID має дорівнювати NET_SERVICE_ID_IBOX_INFO (0x0*C*).

Поле PacketType – NET_PACKET_TYPE_CMD (0x*15*). Роутер відповідає, змінивши значення цього поля на NET_PACKET_TYPE_RES (0x*16*).

Поле OpCode для виконання команд має дорівнювати NET_CMD_ID_MANU_CMD (0x*33*).

Поле Info може бути довільним. Використовується як мітка, щоб розрізняти відповіді служби infosvr на різні пакети-запити.

Поле MacAddress завдяки помилці, що розглянута вище, не має значення.

Поле Password не використовується. Імовірно розробники програмного забезпечення роутера планували використовувати його для коректної аутентифікації, але так і не реалізували задумане.

Поле len містить довжину команди, яку необхідно виконати.

Поле cmd містить саму команду.

Було розроблено програму-експлойт, що формує такий пакет, налаштовує його, приймає рядок-команду від користувача та відправляє на роутер-ціль.

У процесі відлагодження експлойту виявлено такі особливості вразливості:

- Максимальна довжина користувацької команди – 238 символів;
- Максимальна довжина відповіді – 420 символів.

Максимальна довжина відповіді обумовлена розміром буферу на який вказує поле cmd.

Демонстрацію роботи експлойту можна побачити на рисунку 1.

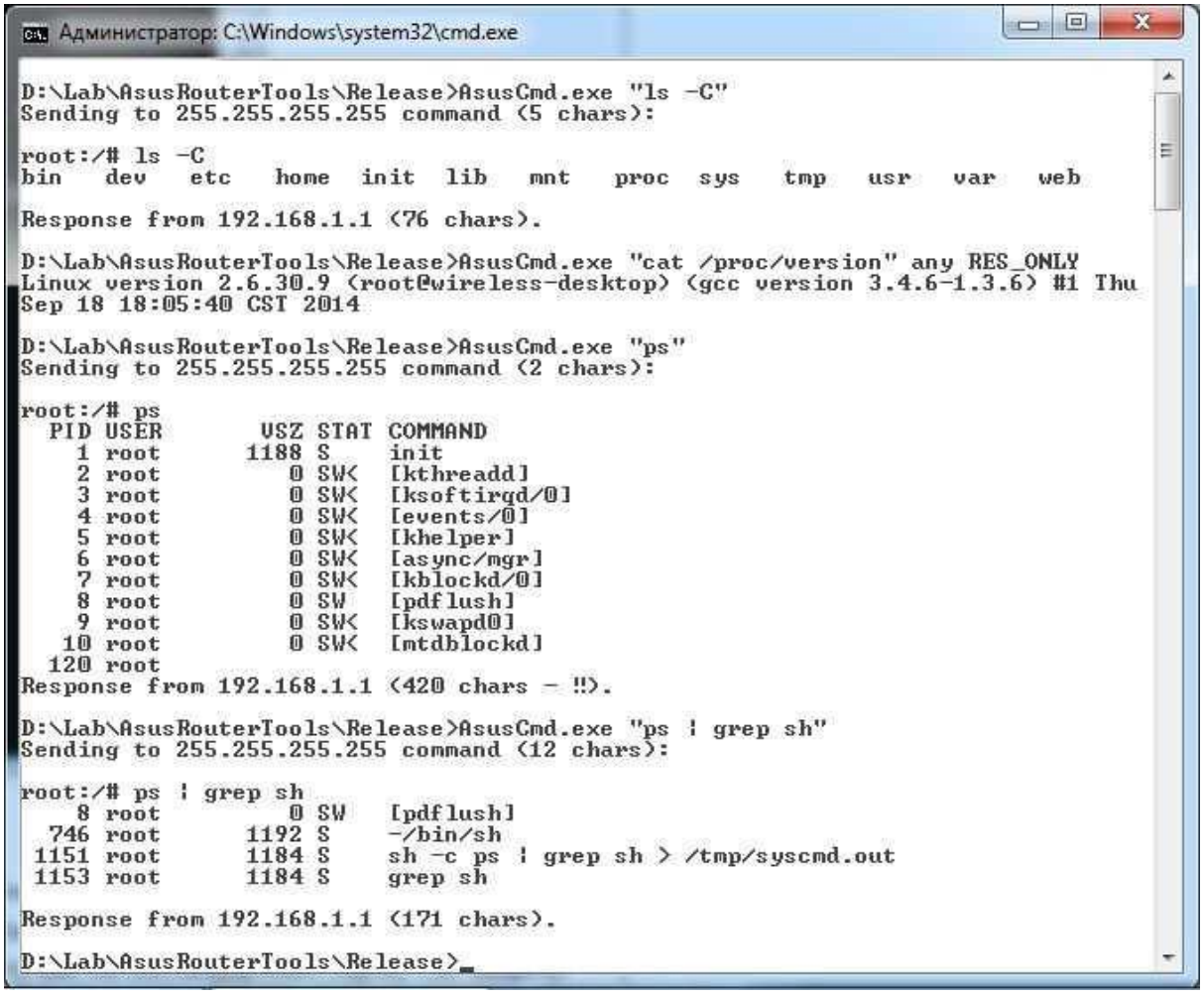

Рисунок 1 – Демонстрація роботи експлойту

Але так працювати незручно. Використовуючи експлойт можна ввімкнути службу *telnet* і дозволити собі підключення до роутера за допомогою програми-термінала без запиту паролю. Це здійснюється виконанням команди «telnetd -l/bin/sh -p777». Ця команда запустить службу, що чекає на вхідні з'єднання на порті під номером 777. Тепер немає жодних обмежень.

Отже, маємо повний контроль над системою маршрутизатора. Роутер можна перезавантажити, вивести з ладу (пошкодити прошивку) чи подивитись/змінити його налаштування. А ще можна завантажувати на нього сторонні файли (у тому числі двійкові програми). Ідея така:

- Зчитуємо порцію байт з файла-джерела;
- Перетворюємо їх в текстовий вигляд, що зрозумілий для команди «echo -e»;
- Формуємо пакет із системною командою;
- Відправляємо його на розтер;
- Повторюємо попередні дії поки не перешлемо весь файл.

На перший погляд нічого складного. Але є дві перешкоди. Перша – це обмеження на максимальну довжину користувацької команди. Це не критично. Просто доведеться розбивати файл-джерело на більшу кількість частин. Друга перешкода – ненадійність протоколу UDP. Треба вживати додаткових заходів, щоб забезпечити надійність передачі. Проблему вирішено наступним чином: окремі частини файлу-джерела записуються в окремі файли на роутері. Потім перевіряється вміст цих файлів на відповідність файлу-джерелу. Якщо якась частина відсутня або пошкоджена – її передача повторюється. Коли всі частини передано та перевірено, виконується їх об'єднання за допомогою команди «cat» в один файл.

Якщо було передано виконуваний файл (*Executable and Linkable Format*), то відразу його запустити не вийде. Як відомо, в системі *Linux* для запуску виконуваних файлів потрібно, щоб був встановлений біт дозволу на виконання. Проблема в тому, в операційній системі роутера не встановлено програму *chmod*, а отже, немає можливості змінювати режим доступу до файлів. Але проблему можна обійти. Треба скопіювати файл, у якого заздалегідь є дозвіл на виконання, в каталог для тимчасових файлів. Потім – скопіювати в той же каталог під тим же іменем файл, що завантажено з комп'ютера.

Тепер програму, що завантажена з комп'ютера можна запускати на виконання.

Більше можна дізнатися зі статті, що опублікована мною на електронному ресурсі за адресою [2] під назвою «Эксплуатируем root-уязвимость в роутерах Asus».

Висновки. Програмісти компанії Asus допустили помилку в коді програмного забезпечення для своїх маршрутизаторів, що стало причиною виникнення серйозної вразливості. Вразливість дозволяла неавторизованим користувачам виконувати будь-які системні команди з найвищими правами. Це давало їм повну владу над роботою пристрою. У статті запропоновані програмні засоби, які дозволяють скоригувати програмний код та виключити потенційну небезпеку неавторизованого виконання команд.

ЛІТЕРАТУРА

1. Все продукты Asus [Електронний ресурс]. – Режим доступу: URL: http://www.asus.com/ua/Networking/AllProducts/ – назва з екрану.
2. Эксплуатируем root-уязвимость в роутерах Asus - Хабрахабр [Електронний ресурс]. – Режим доступу: URL: http://habrahabr.ru/post/253013/ – назва з екрану.
3. Got an Asus router? Someone on your network can probably hack it [Електронний ресурс]. – Режим доступу: URL: http://arstechnica.com/security/2015/01/got-an-asus-router-someone-on-your-network-can-probably-hack-it/ – назва з екрану.
4. ASUS Router infosvr UDP Broadcast root Command Execution [Електронний ресурс]. – Режим доступу: URL: https://github.com/jduck/asus-cmd – назва з екрану.