УДК 004.457

АНАЛИЗ УЯЗВИМОСТИ В ИНТЕРПРЕТАТОРЕ BASH – SHELLSHOCK

Бахтин А.О., Шерстнёв В.С., Шерстнёва А.И.

ФГБОУ ВПО «Национальный исследовательский Томский политехнический университет», Томск, e-mail: vss@tpu.ru, sherstneva@tpu.ru

В данной статье описывается уязвимость командной оболочки bash, поставляемой в качестве базового интерпретатора для большинства дистрибутивов операционной системы Linux. Широкая распространенность интерпретатора bash позволяет говорить о многочисленных потенциально уязвимых системах. В статье описываются векторы, которыми может воспользоваться злоумышленник для атаки на целевую систему. Такими системами могут быть веб-сервера, DNS-сервера, особым образом сконфигурированные SSH-сервера, файловые-сервера на основе операционных систем семейства Unix и Linux. Целью данных атак может быть захват системы, выполнение на ней произвольных команд, дальнейшее распространение на другие системы и их последующий захват. Отмечается простота эксплуатации уязвимости, не требующая высокой квалификации злоумышленника. В статье также приведено описание мер для противодействия использования данной уязвимости.

Ключевые слова: bash, Shellshock, уязвимость, программное обеспечение, информационная безопасность, ТПУ

SHELLSHOCK BASH INTERPRETER VULNERABILITY ANALYSIS

Bakhtin A.O., Sherstnev V.S., Sherstneva A.I.

National Research Tomsk Polytechnic University, Tomsk, e-mail: vss@tpu.ru, sherstneva@tpu.ru

This article describes vulnerability in bash command shell that comes as a default interpreter in most Linux operating system distributives. There are a lot of potentially vulnerable systems on Internet because of bash interpreter prevalence. This article also describes vectors that could be exploited by malicious user to attack target system. Such systems are: web servers, DNS servers, improperly configured SSH servers, file servers. Target of such types of attack could be system takeover, arbitrary commands execution, further spread on other systems and takeover of them. Article notes simplicity of vulnerability exploitation that does not require malicious user's high qualification. Article also describes countermeasures of vulnerability exploitation.

Keywords: bash, Shellshock, vulnerability, software, information security, TPU

В 1996 году, 4 июня на космодроме во Французской Гвиане был произведён первый запуск ракеты-носителя Ariane 5. Уже через 40 секунд после старта она взорвалась на высоте 50 метров. 19 июля, через полтора месяца после катастрофы, специальной комиссией был опубликован доклад, в котором говорилось о том, что взрыв произошел в результате переполнения одной из переменных в программном обеспечении компьютера ракеты. Казалось бы, небольшая уязвимость в программном обеспечении привела к такому значительному ущербу.

Формально, уязвимостью (vulnerability) можно назвать любой параметр информационной системы, изменение которого сторонним пользователем может привести к появлению каких-либо угроз. При этом не существенны мотивы использования уязвимости, так как её наличие уже представляет значительную угрозу функционирования информационной системы. Проблема уязвимостей и их обнаружения берет свое начало практически с самого появления программного обеспечения. За время их существования предпринимались различные попытки классификации по различным признакам. В 1999 году компания MITRE Corporation предложила свое решение проблемы классификации, которое не зависело от различных производителей сканеров безопасности. Данное решение было реализовано в виде специализированной базы данных с перечнем всех известных уязвимостей (Common Vulnerability Enumeration), которая позже была опубликована под названием «Common Vulnerabilities and Exposures».

Уязвимость интерпретатора Bash. 24 сентября 2014 г. в интерпретаторе Bash французским специалистом по Unix/Linux Стефаном Чазеласом (Stephane Chazelas) была обнаружена критическая уязвимость, позволяющая удаленно выполнять произвольный код на целевой системе. Этот недостаток в интерпретаторе существовал с 1994 г. [3]. Уязвимости присвоен уровень «10 из 10» и целый ряд номеров CVE: CVE-2014-7169, CVE-2014-7186, CVE-2014-7187, CVE-2014-6277, CVE-2014-6278. Было даже придумано персональное имя для этой уязвимости – Shellshock. Именно она станет предметом исследования в данной статье. Такой высокий балл был присвоен уязвимости за лёгкость осуществления атаки, а также за отсутствие необходимости в аутентификации для использования Bash с помощью скриптов.

Данная статья ставит перед собой цель описать эту уязвимость, понять причину, по которой она возникает, определить векторы

атаки на уязвимые системы и сервисы, привести список мер по противодействию эксплуатации.

В наиболее простом понимании, Bash – это командный интерпретатор (оболочка), распространённый в операционных системах семейства Unix/Linux. Оболочка Bash часто используется при подключении к операционной системе посредством протоколов SSH или Telnet. В то же время, оболочка Bash может использоваться для обработки CGI-скриптов на веб-серверах Apache. Благодаря распространённости данной оболочки уязвимость Shellshock является столь опасной. Не будет преувеличением сказать, что в течение ближайших нескольких лет многие компьютеры под Linux и Mac OS всё ещё будут открыты для эксплуатации этой уязвимости из-за того, что огромное количество старых устройств так и не получит обновления.

Рассмотрим причину, по которой интерпретатор Bash стал уязвим и почему эксплуатация этой уязвимости не требует серьёзной квалификации у злоумышленника.

Основная опасность заключается в возможности произвольно задать переменные среды внутри интерпретатора Bash, которая задаёт определение функций. Рассмотрим фрагмент программного кода для языка bash, который приводит к эксплуатации уязвимости:

```
$ myfunc() { echo «Hello»; }
$ export -f myfunc
$ env | grep -A1 ^myfunc
myfunc=() { echo «Hello»}.
```

Уязвимость состоит в том, что если после тела функции (после последнего символа «}») добавить ещё какую-нибудь команду, и экспортировать её, она будет выполнена при вызове дочернего интерпретатора:

```
$ env x='() { :; }; echo «Oh...»' /bin/bash -c 'echo vulnerable'
Oh...
vulnerable
```

[1].

Рассмотрим векторы эксплуатации данной уязвимости.

В течение часа после публикации уязвимости начали появляться сообщения об атаках (DDoS) на компьютерные системы по всему миру. В частности, на крупнейшую в мире распределенную сеть доставки контента (CDN) – Akamai Technologies, а также на подсети Министерства обороны США. Такие молниеносные действия злоумышленников оказались возможными из-за простоты эксплуатации уязвимости. Вектор первоначальных атак заключался в следующем. Как мы сказали выше, веб-сервер Apache может работать в качестве обработчика CGI-скриптов. Когда веб-сервер получает запрос от клиента, появляется сразу три вектора для эксплуатации уязвимости: URL, заголовки, посылаемые вместе с URL, и то, что мы знаем под названием «аргументы», передаваемые в дальнейшем скрипту.

Так выглядит легитимный HTTP запрос клиента серверу:

```
GET / HTTP/1.1
Accept-Language: en-US,en;q=0.8,fr;q=0.6
Host: example.com.
```

В данном случае URL это «/» (главная страница) и заголовки: Accept-Language и т.д. Эти заголовки предоставляют веб-серверу информацию о возможностях браузера клиента, предпочитаемый язык, а также тип веб-браузера. Для веб-сервера не является чем-то необычным в дальнейшем превратить полученные заголовки в переменные для дальнейшего разбора и анализа (например сайт может сразу отображаться на необходимом языке). Таким образом, эти заголовки посимвольно копируются в переменные следующим образом:

```
HTTP_ACCEPT_LANGUAGE=en-US,en;q=0.8,fr;q=0.6
HTTP_HOST=example.com.
```

До тех пор, пока эти переменные хранятся внутри ПО веб-сервера и не передаются сторонним программам, сервер не уязвим. Но как только эти переменные передаются интерпретатору bash, и возникает уязвимость Shellshock. Так, атакующий может модифицировать запрос и добавить к нему строку вида: () { :; }; а следом команду, которую необходимо выполнить на сервере.

Пример такого запроса, который получает веб-сервер от злоумышленника:

```
GET / HTTP/1.1
Accept-Language: en-US,en;q=0.8,fr;q=0.6 () { :; }; /bin/bash -c 'ping 8.8.8.8'
Host: example.com.
```

Таким образом, скопированный посимвольно в переменную HTTP_ACCEPT_LANGUAGE заголовок «Accept-Language» при разборе CGI-скриптом приведет к выполнению интерпретатором Bash команды ping на сервер с адресом 8.8.8.8 (рис. 1).

Ответ веб-сервера отдаётся пользователю вместе с результатом выполнения команды. Конечно же, такая команда является самым безобидным, что может сделать злоумышленник по отношению к цели.

Представим более реалистичные варианты. Вместе с запросом пользователь может отправить строку вида: «*() {:;}; /bin/cat /etc/passwd*».

Команда прочитает содержимое файла passwd, в котором хранятся имена пользователей в системе и вернет вместе в ответом веб-сервера.

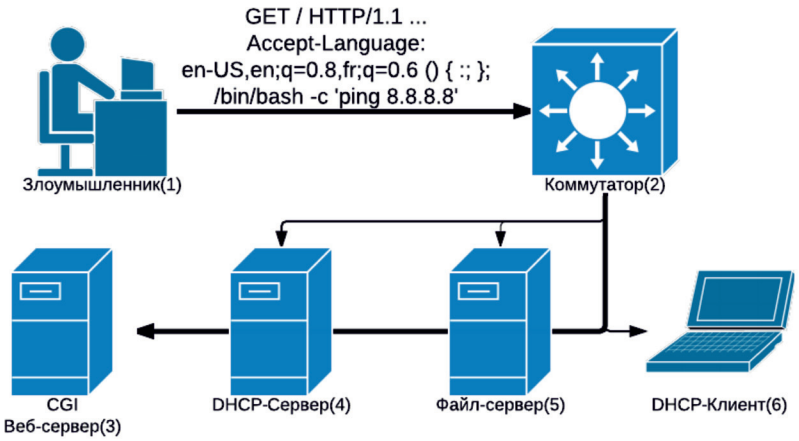

Рис. 1. Пример HTTP запроса, составленного злоумышленником, с модифицированными заголовками

В качестве еще одного сценария эксплуатации данной уязвимости можно привести следующий пример. Злоумышленник отправляет вместе с запросом строку вида:

«() { :;}; /bin/bash -c \»whoami | mail -s ‘example.com l’ xxxxxxxxxxxxxx@gmail.com»

В данном примере выполняется команда whoami, которая покажет пользователя, от которого работает веб-сервер и результат работы данной утилиты отправляется злоумышленнику на почту. Таким образом, у него накапливается база уязвимых хостов. Особое внимание затем следует обратить на цели, на которых веб-сервер запущен от суперпользователя – root.

Самым вероятным сценарием использования данной уязвимости является захват целевой системы. Злоумышленник отправляет на сервер запрос вида:

«() { :;}; /bin/bash -c \»cd /tmp;wget http://192.x.x.x/ji;curl-O/tmp/jihttp://192.x.x.x/ji ; perl /tmp/ji;rm -rf /tmp/ji\»».

Результатом выполнения данной команды будет загрузка на сервер файла скрипта, который запускается интерпретатором perl, и, вероятнее всего, затем этот скрипт открывает обратное соединение с командным сервером и переводится в состояние ожидания для последующего принятия команд от злоумышленника. Исходный файл скрипта удаляется.

Уязвимость SSH-сервера. Ещё одним вектором эксплуатации уязвимости является атака на сервер, на котором запущена служба SSH. SSH – сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов) [5].

Выглядит такой вектор следующим образом. Предположим, что в файл «authorized_keys», расположенный на удаленном сервере, который служит для авторизации пользователей по паре ключей, дописана команда, которую пользователь выполнит в любом случае при входе на сервер. Например, будет дописана строка вида «command=’echo 1’». Таким образом, каждый раз при входе на сервер на экран пользователя будет выдаваться строка ‘1’. Если в качестве интерпретатора будет задан интерпретатор Bash, то злоумышленник может воспользоваться этим и передать вместе со строкой подключения вредоносный код следующим образом (рис. 2):

«ssh testuser@localhost ‘() { :;}; echo MALICIOUS CODE’».

Результат выполнения команды:

«MALICIOUS CODE».

Команда после «;» выполнится перед командой, выводящей на экран пользователя строку «1» [2].

Уязвимость DHCP-клиента. Наконец еще одним вектором эксплуатации уязвимости может стать атака на клиента DHCP-сервера. В рамках стандартного взаимодействия DHCP-клиент и сервера, клиент запрашивает аренду IP-адреса у сервера. DHCP-сервер в свою очередь имеет право отправить клиенту не только основные параметры конфигурации ip-адреса, но и ряд дополнительных параметров, которые клиент сохранит в переменных среды окружения, и в конце концов может привести к использованию уязвимости Shellshock в операционной системе, подключенной к локальной сети. К тому же злоумышленник может представиться поддельным DHCP-сервером и перехватить запрос на получение параметров от клиента, предназначенных легитимному DHCP-серверу.

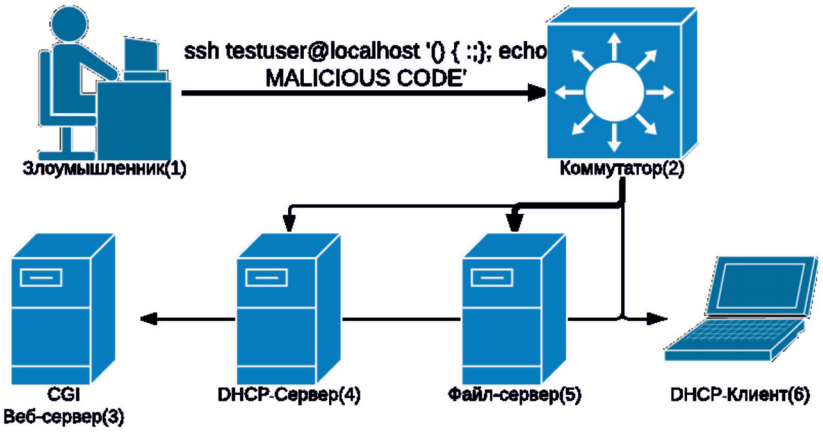

Рис. 2. Передача вредоносного кода SSH-серверу в строке подключения

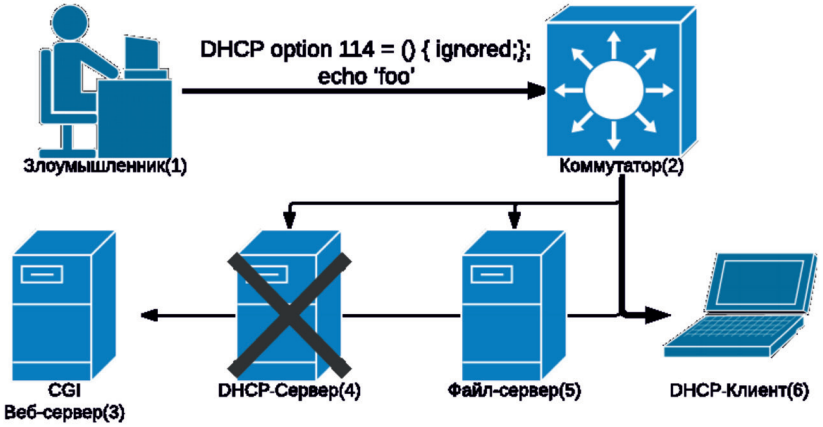

Рис. 3. Отправка вредоносного кода DHCP-клиенту

Злоумышленнику необходимо настроить собственный DHCP-сервер на локальной машине. Установить в качестве значения опции DHCP-сервера 114 (URL по умолчанию) строку вида: *() { ignored;}; echo 'foo'*. Вероятнее всего, подойдут и другие опции, но данная является более надежной. Таким образом, при запросе параметров DHCP-сервера клиента вместе с необходимыми полями получит вредоносный код и выполнит его (рис. 3). Скорее всего, даже с правами суперпользователя, т.к. изменение параметров DHCP на большинстве систем происходит от пользователя root [4].

Столь серьезная уязвимость затронула большое количество производителей программного обеспечения. Большинство из таких производителей отреагировали незамедлительно и оперативно исправили уязвимость в своих системах. Но стоит отметить, что первые обновления для исправления уязвимости не были эффективными. Специалистам по информационной безопасности некоторое время удавалось обходить эти обновления. Можно говорить о том, что большинство крупных вендоров справилось с задачей устранения уязвимости за 3–4 дня. Также стоит отметить, что для данной бреши не существует быстрого решения проблемы, кроме как обновление от производителя. В качестве крайней меры защиты можно, например, отключить от компьютерной сети веб-сервер или DHCP-сервер. Но такие меры не являются допустимыми в большинстве организаций.

Заключение

Рассмотренная проблема является только второй по счёту уязвимостью, за всю историю существования программного обеспечения, которая получила свое персональное имя, что говорит нам о её серьёзности. Десятки и сотни тысяч систем были скомпрометированы за считанные дни после её публикации. Ещё многие и многие годы специалисты по информационной безопасности будут находить неисправленную брешь в системах. Особенно уязвимыми остаются офлайн системы, не имеющие возможности обновления, а также встраиваемые системы.

Список литературы

1. CVE-2014-6271, CVE-2014-7169. [Электронный ресурс]. – Режим доступа: http://habrahabr.ru/post/238021/ (22.03.2015).

2. How can shellshock be exploited over SSH? [Электронный ресурс]. – Режим доступа: http://unix.stackexchange.com/questions/157477/how-can-shellshock-be-exploited-over-ssh (22.03.2015).

3. Shellshock BASH vulnerability tester. [Электронный ресурс]. – Режим доступа: https://shellshocker.net/ (22.03.2015).

4. Shellshock DHCP RCE Proof of Concept. [Электронный ресурс]. – Режим доступа: https://www.trustedsec.com/september-2014/shellshock-dhcp-rce-proof-concept/ (22.03.2015).

5. SSH. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SSH (22.03.2015).