**Уязвимость**

Формально, уязвимостью 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. 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, которая позже была опубликована под названием «Common Vulnerabilities and Exposures».

В 1996 году, 4 июня на космодроме во Французской Гвиане был произведён первый запуск ракеты-носителя Ariane 5. Уже через 34 секунде 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. Возникшая ошибка в программном модуле ИСО произошла во время [конвертации](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D1%82%D0%B8%D0%BF%D0%B0) [64-битного](https://ru.wikipedia.org/wiki/64_%D0%B1%D0%B8%D1%82%D0%B0) [вещественного числа](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%89%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) в [16-битное](https://ru.wikipedia.org/wiki/16_%D0%B1%D0%B8%D1%82) [знаковое целое](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D0%B5_(%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)#.D0.A6.D0.B5.D0.BB.D1.8B.D0.B5_.D1.81.D0.BE_.D0.B7.D0.BD.D0.B0.D0.BA.D0.BE.D0.BC), и при этом случилось [арифметическое переполнение](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5) последней. Этот неудачный запуск стал одной из самых дорогостоящих [компьютерных ошибок](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B0) в истории — оценки только материальных потерь варьируются от 360 до 500 млн [долларов](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BB%D0%BB%D0%B0%D1%80_%D0%A1%D0%A8%D0%90).

**Уязвимость интерпретатора Bash**

24 сентября 2014 г. в интерпретаторе Bash французским специалистом по Unix/Linux Стефаном Чазеласом была обнаружена критическая уязвимость, позволяющая удаленно выполнять произвольный код на целевой системе. Этот недостаток в интерпретаторе существовал с 1994 г. Уязвимости присвоен уровень «10 из 10».  Такой высокий балл был присвоен уязвимости за лёгкость осуществления атаки, а также за отсутствие необходимости в аутентификации для использования Bash с помощью скриптов. Было даже придумано персональное имя для этой уязвимости – Shellshock.

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 том, что Bash позволяет экспортировать функции как переменные окружения. Рассмотрим фрагмент программного кода для языка bash, который приводит к эксплуатации уязвимости:

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.

**CGI**

В течение часа после публикации уязвимости начали появляться сообщения об атаках (DDoS) на компьютерные системы по всему миру. В частности, на подсети Министерства обороны США. Такие молниеносные действия злоумышленников оказались возможными из-за простоты эксплуатации уязвимости. Вектор первоначальных атак заключался в том что веб-сервер Apache может работать в качестве обработчика CGI-скриптов.

**CGI** (Common Gateway Interface — «общий интерфейс шлюза») — стандарт [интерфейса](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), используемого для связи внешней [программы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) с [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

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

**SSH**

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

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

**ssh testuser@localhost ‘() { :;}; echo MALICIOUS CODE’**

**«MALICIOUS CODE»**

**DHCP-СЕРВЕР**

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.

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.