[Перевод] Защищаем сайт с помощью ZIP-бомб

Старые методы по-прежнему работают

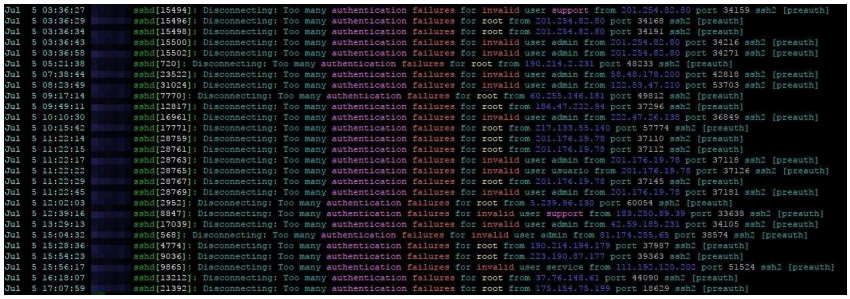
[Обновление] Теперь я в каком-то списке спецслужб, потому что написал статью про некий вид «бомбы», так?

Если вы когда-нибудь хостили веб-сайт или администрировали сервер, то наверняка хорошо знаете о плохих людях, которые пытаются сделать разные плохие вещи с вашей собственностью.

Когда я в возрасте 13 лет впервые захостил свою маленькую Linux-коробочку с доступом по SSH, я смотрел логи и каждый день видел IP-адреса (в основном, из Китая и России), которые пытались подключиться к моей сладенькой маленькой коробочке (которая на самом деле была старым ноутом ThinkPad T21 со сломанным дисплеем, жужжавшим под кроватью). Я сообщал эти IP их провайдерам.

На самом деле если у вас Linux-сервер с открытым SSH, то можете сами посмотреть, сколько попыток подключений происходит ежедневно:

```
grep 'authentication failures' /var/log/auth.log
```



Сотни неудачных попыток авторизации, хотя на сервере вообще отключена авторизация по паролю и он работает на нестандартном порту

Wordpress нас приговорил

Ладно, признаем, сканеры веб-уязвимостей существовали и до Wordpress, но после того, как эта платформа стала настолько популярной, большинство сканеров начали проверять неправильно сконфигурированные папки `wp-admin` и непропатченные плагины.

Так что если маленькая начинающая хакерская банда хочет получить немного свеженьких учёток, они скачают один из этих сканерских инструментов и натравят его на кучу веб-сайтов в надежде получить доступ к какому-нибудь сайту и дефейснуть его.

Информация о хабе

0 комментариев

Образец логов при сканировании инструментом Nikto

Вот почему все серверы и админы веб-сайтов имеют дело с гигабайтами логов, полными попыток сканирования. Так что я подумал…

Можно ли нанести ответный удар?

После экспериментов с возможностью потенциального применения IDS или Fail2ban я вспомнил о старых добрых ZIP-бомбах из прошлого.

Что за штука такая — ZIP-бомба?

Как выяснилось, сжатие ZIP великолепно справляется с повторяющимися данными, так что если у вас имеется гигантский текстовый файл, заполненный повторяющимися данными вроде всех нулей, он очень хорошо сожмётся. В смысле, ОЧЕНЬ хорошо.

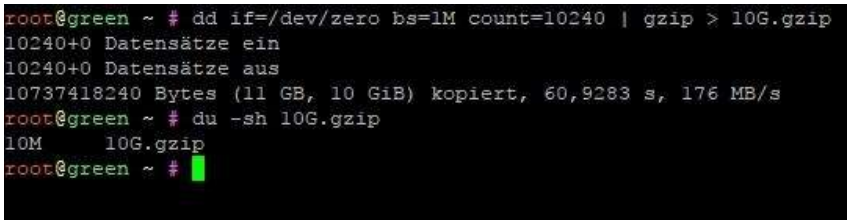
Как показал 42.zip, можно сжать 4,5 петабайта (4 500 000 гигабайт) в 42 килобайта. Когда вы попытаетесь посмотреть содержимое архива (извлечь или разархивировать его), то у вас, вероятно, израсходуется всё дисковое пространство или оперативная память.

Как спустить ZIP-бомбу на сканер уязвимостей?

К сожалению, веб-браузеры не понимают ZIP, но зато они понимают GZIP.

Так что первым делом создадим 10-гигибайтный файл GZIP, заполненный нулями. Можно сделать много вложенных сжатий, но начнём с простого.

```
dd if=/dev/zero bs=1M count=10240 | gzip > 10G.gzip
```



Создание бомбы и проверка её размера

Как видите, её размер 10 МБ. Можно было сжать и получше, но пока хватит.

Теперь установим PHP-скрипт, который доставит её клиенту.

```
<?php
//prepare the client to recieve GZIP data. This will not be suspicious
//since most web servers use GZIP by default
header("Content-Encoding: gzip");
header("Content-Length: ".filesize('10G.gzip'));
//Turn off output buffering
```

```
if (ob_get_level()) ob_end_clean();
//send the gzipped file to the client
readfile('10G.gzip');
```

Готово!

Теперь мы можем использовать её в качестве простой защиты:

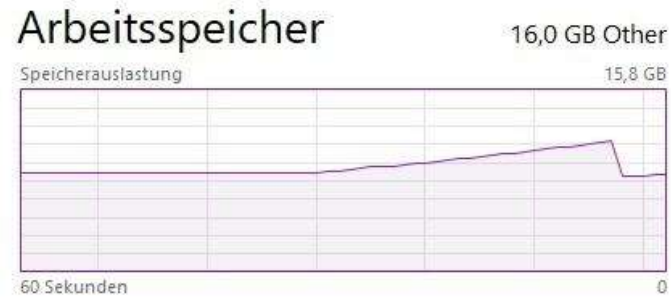
```
<?php
$agent = filter_input(INPUT_SERVER, 'HTTP_USER_AGENT');

//check for nikto, sql map or "bad" subfolders which only exist on wordpress
if (strpos($agent, 'nikto') !== false || strpos($agent, 'sqlmap') !== false ||
startswith($url,'wp-') || startswith($url,'wordpress') || startswith($url,'wp/'))
{
      sendBomb();
      exit();
}

function sendBomb(){
        //prepare the client to recieve GZIP data. This will not be suspicious
        //since most web servers use GZIP by default
        header("Content-Encoding: gzip");
        header("Content-Length: ".filesize('10G.gzip'));
        //Turn off output buffering
        if (ob_get_level()) ob_end_clean();
        //send the gzipped file to the client
        readfile('10G.gzip');
}

function startsWith($a, $b) {
    return strpos($a, $b) === 0;
}
```

Очевидно, этот скрипт не образец элегантности, но он может защитить нас от скрипт-кидди, упомянутых раньше, которые вообще понятия не имеют, что в сканерах можно изменять user-agent.

Итак… Что будет, если запустить этот скрипт?

Клиент	Результат
IE 11	Память расходуется, IE падает
Chrome	Память расходуется, демонстрируется ошибка
Edge	Память расходуется, утекает, грузится вечно
Nikto	Как будто нормально сканирует, но не выдаёт результат
SQLmap	Большой расход памяти, затем падает
Safari	Большой расход памяти, затем падает и перезагружается, затем опять большой расход памяти и так далее...
Chrome (Android)	Память расходуется, демонстрируется ошибка

(если вы проверяли бомбу на других устройствах/браузерах/скриптах, пожалуйста, сообщите мне, и я добавлю результат в таблицу)



Результат загрузки скрипта в Chrome

Если вам нравится рисковать, **попробуйте сами**!

— 07/06/2017 23:32:06 0 habr Аноним

комментарии (0)