



How to defend your website with ZIP bombs

the good old methods still work today

Jul 5th, 2017

© Christian Haschek

[update] I'm on some list now that I have written an article about some kind of "bomb", ain't I?

If you have ever hosted a website or even administrated a server you'll be very well aware of bad people *trying* bad things with your stuff.

When I first hosted my own little linux box with SSH access at age 13 I read through the logs daily and report the IPs (mostly from China and Russia) who tried to connect to my sweet little box (which was actually an old ThinkPad T21 with a broken display running under my bed) to their ISPs.

Actually if you have a linux server with SSH exposed you can see how many connection attempts are made every day:

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



```

Jul 5 03:36:27 sshd[15498]: Disconnecting: Too many authentication failures for invalid user support from 201.254.82.80 port 34159 ssh2 [preauth]
Jul 5 03:36:29 sshd[15496]: Disconnecting: Too many authentication failures for root from 201.254.82.80 port 34168 ssh2 [preauth]
Jul 5 03:36:34 sshd[15498]: Disconnecting: Too many authentication failures for root from 201.254.82.80 port 34191 ssh2 [preauth]
Jul 5 03:36:43 sshd[15500]: Disconnecting: Too many authentication failures for invalid user admin from 201.254.82.80 port 34216 ssh2 [preauth]
Jul 5 03:36:50 sshd[15502]: Disconnecting: Too many authentication failures for invalid user admin from 201.254.82.80 port 34271 ssh2 [preauth]
Jul 5 05:21:38 sshd[720]: Disconnecting: Too many authentication failures for root from 190.214.2.231 port 48233 ssh2 [preauth]
Jul 5 07:38:44 sshd[23522]: Disconnecting: Too many authentication failures for invalid user admin from 58.48.178.200 port 42818 ssh2 [preauth]
Jul 5 08:23:49 sshd[31024]: Disconnecting: Too many authentication failures for invalid user admin from 122.53.47.210 port 53703 ssh2 [preauth]
Jul 5 09:17:14 sshd[7770]: Disconnecting: Too many authentication failures for root from 60.255.146.181 port 49812 ssh2 [preauth]
Jul 5 09:49:11 sshd[12817]: Disconnecting: Too many authentication failures for root from 186.47.232.94 port 37296 ssh2 [preauth]
Jul 5 10:10:30 sshd[16961]: Disconnecting: Too many authentication failures for invalid user admin from 222.47.26.138 port 36849 ssh2 [preauth]
Jul 5 10:15:42 sshd[17771]: Disconnecting: Too many authentication failures for root from 217.133.55.140 port 57774 ssh2 [preauth]
Jul 5 11:22:14 sshd[28759]: Disconnecting: Too many authentication failures for root from 201.176.19.78 port 37110 ssh2 [preauth]
Jul 5 11:22:15 sshd[28761]: Disconnecting: Too many authentication failures for root from 201.176.19.78 port 37112 ssh2 [preauth]
Jul 5 11:22:17 sshd[28763]: Disconnecting: Too many authentication failures for invalid user admin from 201.176.19.78 port 37118 ssh2 [preauth]
Jul 5 11:22:22 sshd[28765]: Disconnecting: Too many authentication failures for invalid user usuario from 201.176.19.78 port 37126 ssh2 [preauth]
Jul 5 11:22:29 sshd[28767]: Disconnecting: Too many authentication failures for root from 201.176.19.78 port 37145 ssh2 [preauth]
Jul 5 11:22:45 sshd[28769]: Disconnecting: Too many authentication failures for invalid user admin from 201.176.19.78 port 37181 ssh2 [preauth]
Jul 5 12:02:03 sshd[2952]: Disconnecting: Too many authentication failures for root from 5.239.96.130 port 60054 ssh2 [preauth]
Jul 5 12:39:16 sshd[8847]: Disconnecting: Too many authentication failures for invalid user support from 183.250.89.39 port 33638 ssh2 [preauth]
Jul 5 13:29:13 sshd[17039]: Disconnecting: Too many authentication failures for invalid user admin from 42.59.185.231 port 34105 ssh2 [preauth]
Jul 5 15:04:32 sshd[568]: Disconnecting: Too many authentication failures for invalid user admin from 81.174.255.65 port 38574 ssh2 [preauth]
Jul 5 15:28:36 sshd[4774]: Disconnecting: Too many authentication failures for root from 190.214.194.179 port 37987 ssh2 [preauth]
Jul 5 15:54:23 sshd[9036]: Disconnecting: Too many authentication failures for root from 223.190.87.177 port 39363 ssh2 [preauth]
Jul 5 15:56:17 sshd[9865]: Disconnecting: Too many authentication failures for invalid user service from 111.192.120.302 port 51524 ssh2 [preauth]
Jul 5 16:18:07 sshd[13212]: Disconnecting: Too many authentication failures for root from 37.76.148.61 port 44090 ssh2 [preauth]
Jul 5 17:07:59 sshd[21392]: Disconnecting: Too many authentication failures for root from 175.154.75.199 port 18629 ssh2 [preauth]

```

Hundreds of failed login attempts even though this server has disabled password authentication and runs on a non-standard port

Wordpress has doomed us all

Ok to be honest, web vulnerability scanners have existed before Wordpress but since WP is so widely deployed most web vuln scanners include scans for some misconfigured wp-admin folders or unpatched plugins.

So if a small, new hacking group wants to gain some hot cred they'll download **one** of **these** scanner **things** and start testing against many websites in hopes of gaining access to a site and **defacing** it.

```

-- [14/Jul/2015:00:47:32 +0200] "GET /modules/fckeditor/fckeditor/license.txt HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006469)"
-- [14/Jul/2015:00:47:32 +0200] "GET /class/fckeditor/license.txt HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006469)"
-- [14/Jul/2015:00:47:32 +0200] "GET /inc/fckeditor/license.txt HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006469)"
-- [14/Jul/2015:00:47:32 +0200] "GET /sites/all/libraries/fckeditor/fckeditor/license.txt HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /FCKeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /Script/fckeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /sites/all/modules/fckeditor/fckeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /modules/fckeditor/fckeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /class/fckeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /inc/fckeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006470)"
-- [14/Jul/2015:00:47:32 +0200] "GET /sites/all/libraries/fckeditor/fckconfig.js HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:32 +0200] "GET /FCKeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:32 +0200] "GET /Script/fckeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /sites/all/modules/fckeditor/fckeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /modules/fckeditor/fckeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /inc/fckeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /class/fckeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /sites/all/libraries/fckeditor/whatsnew.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /FCKeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /Script/fckeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /sites/all/modules/fckeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /modules/fckeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /class/fckeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /inc/fckeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /sites/all/libraries/fckeditor/editor/filemanager/browser/default/browser.html HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006471)"
-- [14/Jul/2015:00:47:33 +0200] "GET /reportservr/ HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006473)"
-- [14/Jul/2015:00:47:34 +0200] "GET /j2ee/examples/servlets/ HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006474)"
-- [14/Jul/2015:00:47:34 +0200] "GET /j2ee/examples/jsp/ HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006475)"
-- [14/Jul/2015:00:47:34 +0200] "GET /Messages/ HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006476)"
-- [14/Jul/2015:00:47:34 +0200] "GET /console-selfservice/ HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006478)"
-- [14/Jul/2015:00:47:34 +0200] "GET /axis2/axis2-web/HappyAxis.jsp HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006479)"
-- [14/Jul/2015:00:47:34 +0200] "POST /search.php HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006480)"
-- [14/Jul/2015:00:47:34 +0200] "POST /private.php HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006481)"
-- [14/Jul/2015:00:47:34 +0200] "GET /en-GB/debug/sso HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006482)"
-- [14/Jul/2015:00:47:34 +0200] "GET /en-US/debug/sso HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006483)"
-- [14/Jul/2015:00:47:34 +0200] "GET /default.htm HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006484)"
-- [14/Jul/2015:00:47:34 +0200] "GET /axis2/axis2-web/HappyAxis.jsp HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006485)"
-- [14/Jul/2015:00:47:34 +0200] "GET /axis2/services/VersionTxsad...etc/passwd HTTP/1.1" 500 625 "-" "Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:006485)"

```

Sample of a log file during a scan using the tool Nikto

This is why all server or website admins have to deal with gigabytes of logs full with scanning attempts. So I was wondering..

Is there a way to strike back?

After going through some potential implementations with [IDS](#) or [Fail2ban](#) I remembered the old [ZIP bombs](#) from the old days.

WTH is a ZIP bomb?

So it turns out ZIP compression is really good with repetitive data so if you have a really huge text file which consists of repetitive data like all zeroes, it will compress it really good. Like REALLY good.

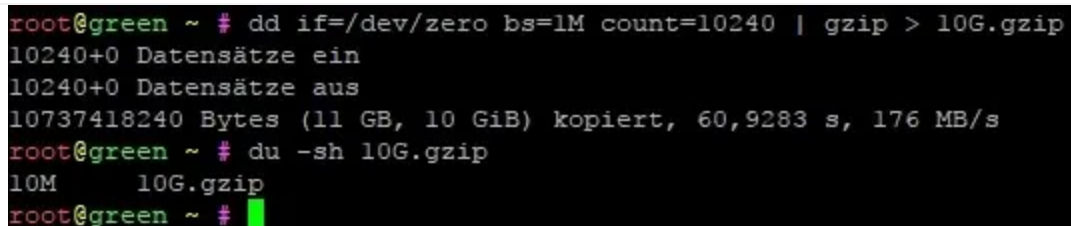
As [42.zip](#) shows us it can compress a 4.5 peta byte (4.500.000 giga bytes) file down to 42 kilo bytes. When you try to actually look at the content (extract or decompress it) then you'll most likely run out of disk space or RAM.

How can I ZIP bomb a vuln scanner?

Sadly, web browsers don't understand ZIP, but they do understand GZIP.

So firstly we'll have to create the 10 giga byte GZIP file filled with zeroes. We could make multiple compressions but let's keep it simple for now.

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

A terminal window with a black background and green text. The prompt is 'root@green ~ #'. The user enters 'dd if=/dev/zero bs=1M count=10240 | gzip > 10G.gzip'. The output shows '10240+0 Datensätze ein', '10240+0 Datensätze aus', and '10737418240 Bytes (11 GB, 10 GiB) kopiert, 60,9283 s, 176 MB/s'. Then the user enters 'du -sh 10G.gzip' and the output is '10M 10G.gzip'. The prompt returns to 'root@green ~ #'.

```
root@green ~ # dd if=/dev/zero bs=1M count=10240 | gzip > 10G.gzip
10240+0 Datensätze ein
10240+0 Datensätze aus
10737418240 Bytes (11 GB, 10 GiB) kopiert, 60,9283 s, 176 MB/s
root@green ~ # du -sh 10G.gzip
10M    10G.gzip
root@green ~ #
```

Creating the bomb and checking its size

As you can see it's 10 MB large. We could do better but good enough for now.

Now that we have created this thing, let's set up a PHP script that will deliver it to a client.

```

<?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');

```

That's it!

So we could use this as a simple defense like this:

```

<?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 ||
{
    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;
}

```

This script obviously is not - as we say in Austria - the yellow of the egg, but it can defend from script kiddies I mentioned earlier who have no idea that all these tools have parameters to change the user agent.

Sooo. What happens when the script is called?

Client	Result
IE 11	Memory rises, IE crashes
Chrome	Memory rises, error shown
Edge	Memory rises, then dripps and loads forever
Nikto	Seems to scan fine but no output is reported
SQLmap	High memory usage until crash
Safari	Hight memory usage, then crashes and reloads, then memory rises again, etc..
Chrome (Android)	Memory rises, error shown

(if you have tested it with other devices/browsers/scripts, please [let me know](#) and I'll add it here)

Reaction of the script called in Chrome

If you're a risk taker: **Try it yourself**

Tags: [security](#) [php](#) [compression](#)

views 171,450

Comment using SSH! [Info](#)

ssh f2fda@ssh.blog.haschek.at

Comments

Get new posts by email

(~ one email every couple of months & no spam)

Your email address

Sign up



1ChrisHMgr4DvEVXzAv1vamkviZNLPS7yx

0x1337C2F18e54d72d696005d030B8eF168a4C0d95