

Nat McHugh

An important site on a minor internet.

Home	About	PGP Key
------	-------	---------

Thursday, February 5, 2015

Create your own MD5 collisions

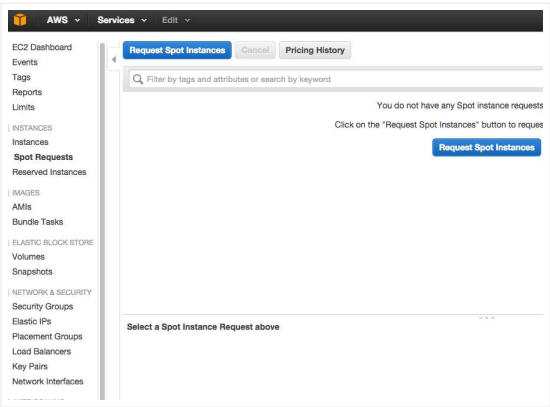
A while ago a lot of people visited my site (~ 90,000) with a post about how easy it is to make two images with same MD5 by using a chosen prefix collision. I used [Marc Steven's HashClash](#) on AWS and estimated the the cost of around \$0.65 per collision.

Given the level of interest I expected to see cool MD5 collisions popping up all over the place. Possibly it was enough for most people to know it can be done quite easily and cheaply but also I may have missed out enough details in my original post.

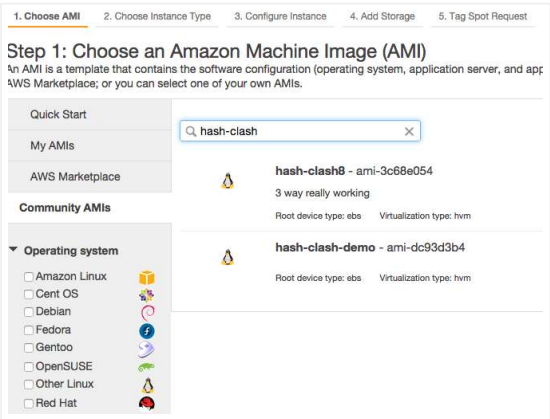
In this further post I've made an AWS image available and created a step-by-step guide so that you too can create MD5 chosen prefix collisions and amuse your friends (disclaimer: they not be that amused). All you need to do is create an AWS instance and run a few commands from the command line. There is a explanation of how the chosen prefix collision works in Marc Steven's [Masters thesis](#).

Here are the steps to create a collision.

1) Log on to AWS console and create a spot request for an instance based on my public Amazon Machine Image (AMI). Spot requests are much cheaper than creating instances directly, typically \$0.065 an hour. They can be destroyed, losing your data, if the price spikes but for fun projects they are the way to go.



I have created a public AMI called hash-clash-demo. It has the id ami-dc93d3b4 and is in the US East (North Virginia) region. It has all the software necessary to create a collision pre-built. Search for it with ami-dc93d3b4 in community AMIs and then choose a GPU2 instance. I promise it does not mine bitcoins in the background although thinking about it this would be a good scam and I may introduce this functionality.



Twitter

Tweets by @natmchugh

natmchugh

@natmchugh

@katie_fenn ok 🔥🔥🔥

17 May

natmchugh

@natmchugh

@katie_fenn I have one if you need to borrow one.

17 May

natmchugh

@natmchugh

Ah yes short Weierstrass form versus Edwards form, very amusing.

12 May

natmchugh

@natmchugh

@katie_fenn Sadly not available beta.companieshouse.gov.uk/search/compani...+

12 May

Embed View on Twitter

Popular Posts

2) Once your request has been created and evaluated hopefully you will have a running instance to connect to via SSH. You may need to create a new key pair, follow the instructions on AWS to do this and install on your local machine. Once you have your key installed log onto instance via ssh as ec2-user.

```

mechugh-mac:~$ ssh -i Virginia.pem ec2-user@ec2-184-73-62-37.compute-1.amazonaws.com
Last login: Mon Feb 2 16:05:33 2015 from 10.150.76.230
ec2-user@ip-10-150-76-230:~$ cat /etc/issue
Amazon Linux AMI
https://aws.amazon.com/amazon-linux-ami/2014.03-release-notes/
25 package(s) needed for security, out of 188 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2014.09 is available.
ec2-user@ip-10-150-76-230 ~$

```

3) The shell script for running hash clash is located at /home/ec2-user/hashclash/src/scripts. Change into that directory and download some data to create a collision. Here I download a couple of jpeg images from tumblr.

```

ec2-user@ip-10-150-76-230 ~$ cd hashclash/src/scripts/
ec2-user@ip-10-150-76-230 scripts$ wget -O plane.jpg http://41.media.tumblr.com/9b01398b1
-2015-02-04 10:30:06-- http://41.media.tumblr.com/9b01398b124aaa795d43e47719e04d00/tumblr
Resolving 41.media.tumblr.com (41.media.tumblr.com)... 209.197.3.20, 2001:4de8:acd0:1:1:1:14
Connecting to 41.media.tumblr.com (41.media.tumblr.com)[209.197.3.20]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 227041 (222K) [image/jpeg]
Saving to: 'plane.jpg'
100%[=====] 227041 10:30:06 (36.7 MB/s) - 'plane.jpg' saved [227041/227041]

ec2-user@ip-10-150-76-230 scripts$ wget -O ship.jpg http://41.media.tumblr.com/e4918bed3d
-2015-02-04 10:30:12-- http://41.media.tumblr.com/e4918bed3d33f2a72cd5d3f6f09/tumblr
Resolving 41.media.tumblr.com (41.media.tumblr.com)... 209.197.3.20, 2001:4de8:acd0:1:1:1:14
Connecting to 41.media.tumblr.com (41.media.tumblr.com)[209.197.3.20]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 334497 (327K) [image/jpeg]
Saving to: 'ship.jpg'
100%[=====] 334497 10:30:12 (44.1 MB/s) - 'ship.jpg' saved [334497/334497]

ec2-user@ip-10-150-76-230 scripts$

```

4) It is best to run the shell script in a screen session so you can detach from it and do other stuff. Start a screen session by typing

screen

Once you are in the screen session kick off the cpc.sh shell script with your two files. Send the outputs to a log file in this case I called it demo.output.

```
ec2-user@ip-10-150-76-230 scripts$ ./cpc.sh plane.jpg ship.jpg &> demo.output
```

Detach from the screen session with Ctrl A + D

```

ec2-user@ip-10-150-76-230 scripts$ screen
[detached]
ec2-user@ip-10-150-76-230 scripts$

```

5) Tailing the log file you should be able to see the birthday attack to get the hash differences into the correct locations starting.

tail -f demo.output



Create your own MD5 collisions

A while ago a lot of people visited my site (~ 90,000) with a post about how easy it is to make two images

with same MD5 by using a chos...



How I created two images with the same MD5 hash

I posted the following images the other day which although looking totally different have exactly the same MD5 hash (e06723d4961a0a3f950e...

How I made two PHP files with the same MD5 hash

I recently posted a link on twitter to two PHP scripts which have different behaviours but the same MD5 hash. To verify this download the fi...



How to make two binaries with the same MD5 hash

One question I was asked when I demo'd creating two PHP files with the same hash is; does it work on compiled binaries? Well the answ...



Three way MD5 collision

Previously I explained how I created two images one of James Brown the other of Barry White with the same MD5 hash. At the end of

the post I...



The Magic Words are Squeamish Ossifrage - factoring RSA-129 using CADO-NFS

This thing got long and can basically be summarised as: I factored smaller RSA numbers on free cloud computing using excellent open sour...



Hash Collisions Reading List

Lately in an effort to code up and properly understand the Wang attack on the MD4 family of hash functions I've been reading a lot of pa...



Images with colliding MD5 hash

These images of James Brown and Barry White have the same MD5 hash e06723d4961a0a3f950e7736f3766338. Don't believe me. You can...



Bug Bounty - Remote Code Execution in Magento 1.x

Magento is a popular ecommerce solution written in PHP. It is widely used for web shops both large and small. The most current product is Ma...



MD5 collisions in ssh keys

You can insert MD5 collision blocks in many data formats and if you do it right the result will be 2 objects that differ but which when pass...

Labels

- Cloud
- Hash functions
- Maths
- MD5
- PHP
- RSA

Blog Archive

- 2017 (1)
- ▼ 2015 (8)
 - September (1)
 - July (2)
 - June (1)
 - May (1)

```

Chosen-prefix file 1: plane.jpg
Chosen-prefix file 2: ship.jpg
birthday searching\n
Birthday search for MD5 chosen-prefix collisions
Copyright (C) 2009 Marc Stevens
http://homepages.cwi.nl/~stevens/

IHW1 = {3192106934,525660334,4202843188,3698886027}
IHW1 = b6af43beaef0541f344c82fa8b8578dc

IHW2 = {3579950299,2053399571,2824271360,2870659336}
IHW2 = dbb461d51364647a00f656a808c91aab

Maximum amount of memory in MB for trails: 100 (local: 100)
Estimated number of trails that will be stored: 1872457 (local: 1872457)
Estimated number of trails that will be generated: 1061342
Estimated complexity per trail: 2^(17)
Estimated complexity on trails: 2^(37.0175)
Estimated complexity on collisions: 2^(27.7053)

Thread 1 created.
Thread 3 created.
Thread 5 created.
Thread 4 created.
Thread 2 created.
Thread 6 created.
Thread 7 created.
Thread 8 created.
Work: 2^(28.4503), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(29.7058), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(30.4344), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(30.8627), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(31.2008), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(31.485), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(31.7212), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(31.9226), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(32.1012), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64
Work: 2^(32.2685), Coll.: 0(uf=0,nuf=0,7=0,q=0,rh=0), Blocks: 64

```

6) Leave the birthday search to do it's thing for an hour or so. Hopefully when you come back the attack should have moved on to the next stage, creating the near collision blocks to gradually reduce the hash differences. The best way to check this is to look at files created. The workdir0 contains all the data for the current collision search for the first near collision block. More of these will be created as more near collision blocks are created.

```

[ec2-user@ip-10-150-76-230 ~]$ cd hashclash/src/scripts/
[ec2-user@ip-10-150-76-230 scripts]$ ls -l
total 2980
-rwxr-xr-x 1 ec2-user ec2-user 3437 Feb  2 16:11 cpc.sh
-rwxr-xr-x 2 ec2-user ec2-user 4096 Feb  4 10:32 data
-rw-rw-r-- 1 ec2-user ec2-user 78987 Feb  4 12:17 demo.output
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file1_0.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file2_0.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file2.bin
-rw-rw-r-- 1 ec2-user ec2-user 1012887 Feb  4 12:11 hashutil5_scn.bin.gz
-rw-rw-r-- 1 ec2-user ec2-user 150 Feb  4 12:11 md5diffpathbackward.cfg
-rw-rw-r-- 1 ec2-user ec2-user 97 Aug 19 14:39 md5diffpathbackward.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 69 Feb  4 12:11 md5diffpathconnect.cfg
-rw-rw-r-- 1 ec2-user ec2-user 16 Aug 19 14:39 md5diffpathconnect.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 343 Feb  4 12:11 md5diffpathforward.cfg
-rw-rw-r-- 1 ec2-user ec2-user 123 Aug 19 14:39 md5diffpathforward.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 39 Feb  4 12:11 md5diffpathhelper.cfg
-rw-rw-r-- 1 ec2-user ec2-user 227041 Feb  3 00:07 plane.jpg
-rw-rw-r-- 1 ec2-user ec2-user 334497 Feb  2 06:33 ship.jpg
-rwxr-xr-x 2 ec2-user ec2-user 4096 Feb  4 12:11 workdir0
[ec2-user@ip-10-150-76-230 scripts]$

```

7) Go away again, a watched collision pretty much never happens. Check back in ~5 hours that it is still going on. Tailing demo.output and listing the directory should let you know roughly what stage the attack is at.


```

Amazon Linux version 2014.09 is available.
[ec2-user@ip-10-150-76-230 ~]$ date
Wed Feb  4 14:13:56 UTC 2015
[ec2-user@ip-10-150-76-230 ~]$ cd hashclash/src/scripts/
[ec2-user@ip-10-150-76-230 scripts]$ ls -l
total 5544
-rwxr-xr-x 1 ec2-user ec2-user 3437 Feb  2 16:11 cpc.sh
drwxrwxr-x 2 ec2-user ec2-user 4096 Feb  4 10:32 data
-rw-rw-r-- 1 ec2-user ec2-user 681635 Feb  4 14:14 demo.output
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file1_0.bin
-rw-rw-r-- 1 ec2-user ec2-user 334592 Feb  4 13:05 file1_1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334656 Feb  4 13:45 file1_2.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file2_0.bin
-rw-rw-r-- 1 ec2-user ec2-user 334592 Feb  4 13:05 file2_1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334656 Feb  4 13:45 file2_2.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file2.bin
-rw-rw-r-- 1 ec2-user ec2-user 1012887 Feb  4 12:11 hashutil5_scn.bin.gz
-rw-rw-r-- 1 ec2-user ec2-user 151 Feb  4 13:45 md5diffpathbackward.cfg
-rw-rw-r-- 1 ec2-user ec2-user 97 Aug 19 14:39 md5diffpathbackward.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 70 Feb  4 13:45 md5diffpathconnect.cfg
-rw-rw-r-- 1 ec2-user ec2-user 16 Aug 19 14:39 md5diffpathconnect.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 341 Feb  4 13:45 md5diffpathforward.cfg
-rw-rw-r-- 1 ec2-user ec2-user 123 Aug 19 14:39 md5diffpathforward.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 40 Feb  4 13:45 md5diffpathhelper.cfg
-rw-rw-r-- 1 ec2-user ec2-user 227041 Feb  3 00:07 plane.jpg
-rw-rw-r-- 1 ec2-user ec2-user 334656 Feb  4 13:45 plane.jpg.coll
-rw-rw-r-- 1 ec2-user ec2-user 334497 Feb  2 06:33 ship.jpg
-rw-rw-r-- 1 ec2-user ec2-user 334656 Feb  4 13:45 ship.jpg.coll
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 13:05 workdir0
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 13:45 workdir1
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 14:06 workdir2
[ec2-user@ip-10-150-76-230 scripts]$

```

Here we are only at block number 2 of probably 9.

8) Come back again about 10-12 hours from start and with any luck we have a collision.

```

[ec2-user@ip-10-150-76-230 scripts]$ ls -l
total 14020
-rwxr-xr-x 1 ec2-user ec2-user 3437 Feb  2 16:11 cpc.sh
drwxrwxr-x 2 ec2-user ec2-user 4096 Feb  4 10:32 data
-rw-rw-r-- 1 ec2-user ec2-user 4628535 Feb  5 02:45 demo.output
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file1_0.bin
-rw-rw-r-- 1 ec2-user ec2-user 334592 Feb  4 13:05 file1_1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334656 Feb  4 13:45 file1_2.bin
-rw-rw-r-- 1 ec2-user ec2-user 334720 Feb  4 14:23 file1_3.bin
-rw-rw-r-- 1 ec2-user ec2-user 334784 Feb  4 15:09 file1_4.bin
-rw-rw-r-- 1 ec2-user ec2-user 334848 Feb  4 16:02 file1_5.bin
-rw-rw-r-- 1 ec2-user ec2-user 334912 Feb  4 17:09 file1_6.bin
-rw-rw-r-- 1 ec2-user ec2-user 334976 Feb  4 20:54 file1_7.bin
-rw-rw-r-- 1 ec2-user ec2-user 335040 Feb  4 23:43 file1_8.bin
-rw-rw-r-- 1 ec2-user ec2-user 335104 Feb  5 02:45 file1_9.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file2_0.bin
-rw-rw-r-- 1 ec2-user ec2-user 334592 Feb  4 13:05 file2_1.bin
-rw-rw-r-- 1 ec2-user ec2-user 334656 Feb  4 13:45 file2_2.bin
-rw-rw-r-- 1 ec2-user ec2-user 334720 Feb  4 14:23 file2_3.bin
-rw-rw-r-- 1 ec2-user ec2-user 334784 Feb  4 15:09 file2_4.bin
-rw-rw-r-- 1 ec2-user ec2-user 334848 Feb  4 16:02 file2_5.bin
-rw-rw-r-- 1 ec2-user ec2-user 334912 Feb  4 17:09 file2_6.bin
-rw-rw-r-- 1 ec2-user ec2-user 334976 Feb  4 20:54 file2_7.bin
-rw-rw-r-- 1 ec2-user ec2-user 335040 Feb  4 23:43 file2_8.bin
-rw-rw-r-- 1 ec2-user ec2-user 335104 Feb  5 02:45 file2_9.bin
-rw-rw-r-- 1 ec2-user ec2-user 334528 Feb  4 12:11 file2.bin
-rw-rw-r-- 1 ec2-user ec2-user 1012887 Feb  4 12:11 hashutil5_scn.bin.gz
-rw-rw-r-- 1 ec2-user ec2-user 152 Feb  4 23:43 md5diffpathbackward.cfg
-rw-rw-r-- 1 ec2-user ec2-user 97 Aug 19 14:39 md5diffpathbackward.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 71 Feb  4 23:43 md5diffpathconnect.cfg
-rw-rw-r-- 1 ec2-user ec2-user 16 Aug 19 14:39 md5diffpathconnect.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 343 Feb  4 23:43 md5diffpathforward.cfg
-rw-rw-r-- 1 ec2-user ec2-user 123 Aug 19 14:39 md5diffpathforward.cfg.template
-rw-rw-r-- 1 ec2-user ec2-user 41 Feb  4 23:43 md5diffpathhelper.cfg
-rw-rw-r-- 1 ec2-user ec2-user 227041 Feb  3 00:07 plane.jpg
-rw-rw-r-- 1 ec2-user ec2-user 335104 Feb  5 02:45 plane.jpg.coll
-rw-rw-r-- 1 ec2-user ec2-user 334497 Feb  2 06:33 ship.jpg
-rw-rw-r-- 1 ec2-user ec2-user 335104 Feb  5 02:45 ship.jpg.coll
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 13:05 workdir0
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 13:45 workdir1
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 14:23 workdir2
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 15:09 workdir3
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 16:02 workdir4
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 17:09 workdir5
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 20:53 workdir6
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  4 23:42 workdir7
drwxrwxr-x 10 ec2-user ec2-user 4096 Feb  5 02:45 workdir8
drwxrwxr-x 2 ec2-user ec2-user 4096 Feb  5 02:45 workdir9

```

This one finished at 02:45 in the morning having been started at 10:30 the previous morning. You can tell when it finished as that was the last point the log was written to. If the log log file is still being updated the collision search is still going on. It took 9 near collision blocks to finally eliminate all the differences which is normal. 16 hours is a bit longer than average.

The collisions have been created in files named plane.jpg.coll and ship.jpg.coll. You can verify they do indeed have the same md5 hash with md5sum.

```

[ec2-user@ip-10-150-76-230 scripts]$ md5sum plane.jpg.coll ship.jpg.coll
253dd04e87492e4fc3471de5e776bc3d plane.jpg.coll
253dd04e87492e4fc3471de5e776bc3d ship.jpg.coll

```


Here are the images with collision blocks added.



I downloaded them to my local machine with scp

```
nmchugh-mac:~ nmchugh$ scp -i virgina.pem ec2-user@ec2-184-73-62-37.compute-1.amazonaws.com:/home/ec2-user/hashclash/src/scripts/ship.j  
ship.jpg.coll  
nmchugh-mac:~ nmchugh$ scp -i virgina.pem ec2-user@ec2-184-73-62-37.compute-1.amazonaws.com:/home/ec2-user/hashclash/src/scripts/plane.  
plane.jpg.coll  
nmchugh-mac:~ nmchugh$
```

Posted by Nathaniel McHugh at 2:01 PM

 +3 Recommend this on Google

Labels: Hash functions, MD5