Brian Cullinan

10/10/2010

# Project #2

## Readme

Running the program is simple. PHP-CLI is required. Just run php aes.php and wait a long time.

## Step-by-Step

My script runs the decrypt program from the command line. It then checks the percentage of the outputted text for English characters. If there are a lot of English characters, it will print out the key and the result. It loops through 0 to 256 for the first 3 characters. There are 3 for-loops for generating every possibility. Since most improperly decrypted messages will contains many binary characters, this should be a reasonable metric since we know the encoded message is English text. The script also prints out a percentage as a status of how many codes it has tried.

Using a comparison of 70 percent valid English characters against the decrypt output would give the string “Imagine taking a picture using your smartphone and immediately having all of your Facebook friends automatically tagged, without even visiting the website, the app, or looking at the picture itself. This is the future that Viewdle plans to make a reality.” Using a key of 112388

## Code

```
<?php

set_time_limit(0);

chdir('/var/www/');

$candidates = array();

$count = 0;
for($i = 0; $i < 100; $i++)
{
    for($j = 0; $j < 256; $j++)
    {
        for($k = 0; $k < 256; $k++)
        {
            unset($output);
            $result = exec('./decrypt ' . sprintf('%02s', dechex($i)) . sprin
tf('%02s', dechex($j)) . sprintf('%02s', dechex($k)) . ' hw.crypt 2>&1', $out
put);
            $key = array_shift($output);
            $words =  array_values(array_unique(preg_split('/[^A-
Z]/i', implode("\n", $output))));

            if(preg_match_all('/[a-
z]/i', implode("\n", $output), $matches) > strlen(implode("\n", $output)) * .
70)
            {
                print 'KEY: ' . sprintf('%02s', dechex($i)) . sprintf('%02s',
 dechex($j)) . "\n";
                print $output . "\n";
                flush();
            }

            $count++;
            if($count % 16777 == 0)
            {
                print ($count / (256*256*256)) * 100 . "\n";
                flush();
                sleep(1);
            }
        }
    }
}
```