```
-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_

    (NNNNNN. (NN)                  (NN)     (NN)  (NN)         ___                NNN    NNNN)   JNN   .NN)
    (NNNNNNN (NN)   ___.     ___   (NN) ___ (NN)  (NN)  .___   NNN    .___    .__ NNN   JNNNN)(NNNNN NNNNN)
    (NN) NNN (NN) (NNNNNN  (NNNNN) (NN)(NN) (NN)  (NN) JNNNNN.(NNNN) NNNNNN. (NNNNNNN  .NN4NN)`"4NNN ""NNN)
    (NNL_NNN (NN) NNN (NN) NNN (NN (NN)NNN  (NN)  (NN) NNN (N)`NNN"`(NN) NNN (NN) NNN  JNF(NN)   NNN   (NN)
    (NNNNNN) (NN) NNN.JNN) NNN (NN (NNNNN`  (NN)  (NN) NNNL/"` NNN  (NNNNNNN (NN) NNN (NN`(NN)   NNN   (NN)
    (NN) NNN (NN) .NNNNNN) NNN     (NNNNN)  (NN)  (NN) `NNNNN  NNN  (NNNNNNN (NN) NNN NNN_JNNL.  NNN   (NN)
    (NN) NNN (NN) NNN`(NN) NNN (NN (NN)NNL  (NN)  (NN) __`NNN) NNN  (NN) NNN (NN) NNN NNNNNNNN)  NNN   (NN)
    (NNLJNNN (NN) NNN (NN) NNN (NN (NN)(NN. (NNNNN(NN) NN (NN) NNN  (NN) NNN (NN) NNN NNNNNNNN)  NNN   (NN)
    (NNNNNNF (NN) NNNNNNN) (NNNNN` (NN)`NNL (NNNNN(NN) 4NNNNN` NNNN) NNNNNN) (NNNNNNN     (NN)   NNN   (NN)
    `""""""  `""`  ""``""`   """   `""` """ `"""""`""`   """    """`  `"""    `"" """     `""`   """   `""`
-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
```

“BLACKLISTED 411 .NET”

Edition 5

02.15.2006

-=-

Table of Contents

-=-

[1] ==Edition Quote==

“Security is a process, not an end state.”
Dr. Mitch Kabay - 1998

[2] ==Articles==

[a]

Hacking the Web
Automatic Obfuscated Submission

By Israel Torres <israel@israeltorres.org>

The following techniques described herein are often used by phishers, spammers and scammers to have your system do something you maybe aren't expecting. The purpose of this article is to demonstrate these techniques so that you may become aware as well as also use them as tools to suit your needs. **Be responsible for your actions.**

Surely by now you have visited questionable links that have taken you for a ride the second you click on them. It used to be very obvious when pop-ups were "the thing". Thanks to browser technology I haven't seen a pop-up in years. However this hasn't stopped unscrupulous people from trying to take things away from you such as your money, identity, information, etc.

Some of the population has become savvy to not click on emails from people they don't know, but most of the population still is click happy.

Personally when I see a link to somewhere either in an email or a web page I take a quick look at my status bar to see where it leads to – this method has proven to be broken in the past so if I am hanging out in a gray area I also right click the link, copy the destination location and paste it into notepad. If things don't look right I'll past it into word pad to shift the fonts around and see if someone is trying to use an Alphabetic-L for a Numerical-One, or a Zero for an O. If I don't feel comfortable with what I see I don't bother using my browser to click on it. Instead I use something like wget to pull the file for further examination. I also make sure I don't have anything installed or running such as Google Desktop which had been known in the past to trigger system exploits upon dynamic indexing the second the file is on the drive. The next thing I do is rename the file to something without a known extension such as ".dump__" just in case it is written to exploit an expected associated application. (*What me paranoid?*) Then I take a deep breath and use a hex editor to open the file (It could be a binary file used to trigger something). After it proves to be a flat file I examine the code and look for obfuscation of any type, including calls to external scripts, css… ANYTHING out of place. Once I feel comfortable I record the hash of the file and the link in case I return to (what I suspect) a static page and move on. This long process has saved my system more times than I can count. I have also learned a lot of the techniques from this process that keep these guys in business.

This particular technique can be found in many implementations and is popular for hiding what is actually happening. However with just a little prodding it is easy to find out as well as modify it to add to your toolbox. I begin by explain how to create this scenario after which you will know how to “crack” these scenarios in the future.

Source: automatic_obfuscated_submission.html
This is what your typical web surfer will see if they click view source on their browser if they happen to get a chance to stop it before it executes.

```
<html>
<title>Hacking The Web : Automatic Obfuscated Submission - blacklisted411.israeltorres.org</title>
<!--
::::::::::::::::::::::
:: Blacklisted!411 .NET Edition 5
:: Hacking The Web : Automatic Obfuscated Submission
:: By Israel Torres <israel@israeltorres.org>
:: Source available at: http://blacklisted411.israeltorres.org
:::::::::::::::::::::::::::::::::::
:: Read automatic_obfuscated_submission_worksheet.txt for more information
:::::::::::::::::::::::::::::::::::
// -->
<body onLoad="document.forms[0].submit()">
<div id="bl411">
<script>
 document.getElementById("bl411").innerHTML = unescape('%3c%66%6f%72%6d%20%61%63%74%69%6f
%6e%3d%22%68%74%74%70%3a%2f%2f%77%77%77%2e%67%6f%6f%67%6c%65%2e%63%6f%6d%2f%73%65%61%72%63%68
%22%20%6e%61%6d%65%3d%22%66%22%20%6d%65%74%68%6f%64%3d%22%70%75%74%22%3e%3c%69%6e%70%75%7
4%20%74%79%70%65%3d%68%69%64%64%65%6e%20%6e%61%6d%65%3d%68%6c%20%76%61%6c%75%65%3d%65%6e
%3e%3c%69%6e%70%75%74%20%6d%61%78%6c%65%6e%67%74%68%3d%32%30%34%38%20%73%69%7a%65%3d%35
%35%20%6e%61%6d%65%3d%71%20%76%61%6c%75%65%3d%22%42%6c%61%63%6b%6c%69%73%74%65%64%20
%34%31%31%2c%49%73%72%61%65%6c%20%54%6f%72%72%65%73%22%20%74%69%74%6c%65%3d%22%47%6f%6f
%67%6c%65%20%53%65%61%72%63%68%22%3e%3c%69%6e%70%75%74%20%74%79%70%65%3d%22%73%75%62%6d
%69%74%22%20%76%61%6c%75%65%3d%22%47%6f%6f%67%6c%65%20%53%65%61%72%63%68%22%20%6e%61%6d
%65%3d%62%74%6e%47%3e%3c%2f%66%6f%72%6d%3e')
</script>

</body>
</html>
```

Step 1: Create Form
Using the template below you can modify the form at will to suit your needs. This is a simple example that just takes you to the Google website with results on the query string.

```
<form action="http://www.google.com/search" name="f" method="put">
 <input type=hidden name=hl value=en>
 <input maxlength=2048 size=55 name=q value="Blacklisted 411,Israel Torres" title="Google Search">
 <input type="submit" value="Google Search" name=btnG>
</form>
```

Step 2: Compress Form
To compress the form, remove unnecessary whitespace (carriage return, new line, spaces in between brackets) and modify into one line.

```
<form action="http://www.google.com/search" name="f" method="put"><input type=hidden name=hl
value=en><input maxlength=2048 size=55 name=q value="Blacklisted 411,Israel Torres" title="Google
Search"><input type="submit" value="Google Search" name=btnG></form>
```

Step 3: Encode Form

This further obfuscates what is supposed to happen when the page loads to a common web surfer.

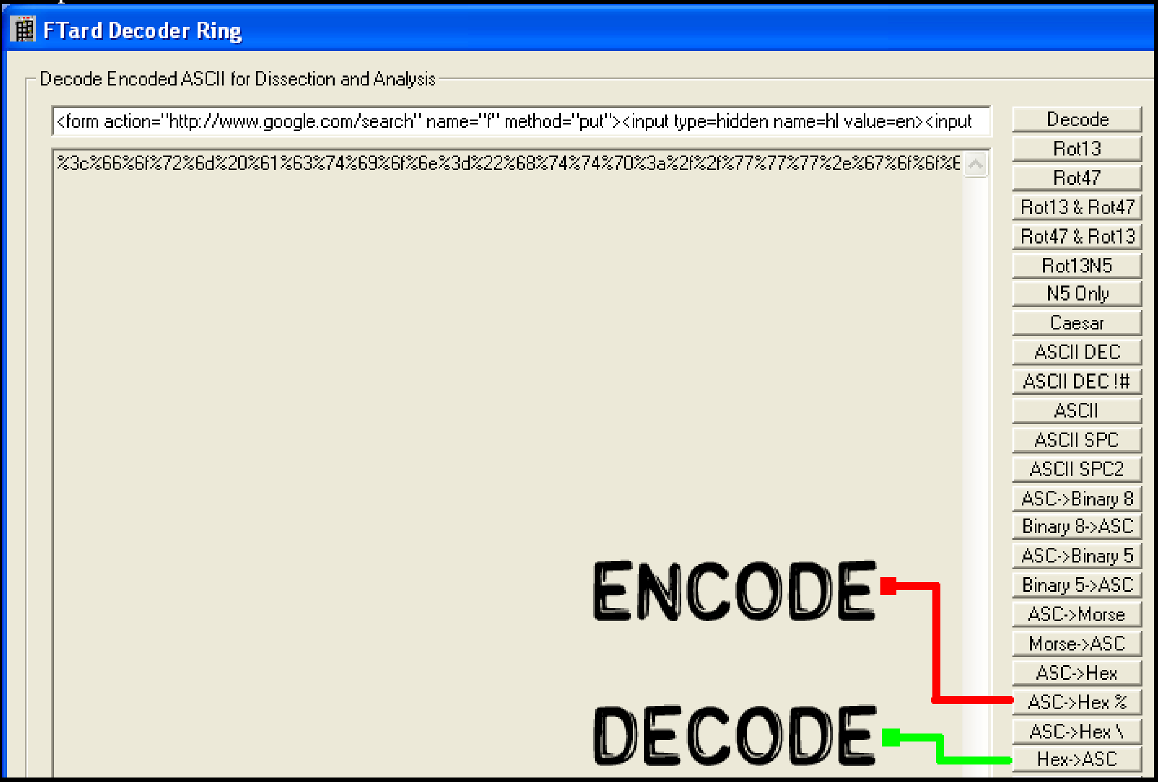
```
%3c%66%6f%72%6d%20%61%63%74%69%6f%6e%3d%22%68%74%74%70%3a%2f%2f%77%77%77
%2e%67%6f%6f%67%6c%65%2e%63%6f%6d%2f%73%65%61%72%63%68%22%20%6e%61%6d
%65%3d%22%66%22%20%6d%65%74%68%6f%64%3d%22%70%75%74%22%3e%3c%69%6e%70
%75%74%20%74%79%70%65%3d%68%69%64%64%65%6e%20%6e%61%6d%65%3d%68%6c%20
%76%61%6c%75%65%3d%65%6e%3e%3c%69%6e%70%75%74%20%6d%61%78%6c%65%6e%67
%74%68%3d%32%30%34%38%20%73%69%7a%65%3d%35%35%20%6e%61%6d%65%3d%71%20
%76%61%6c%75%65%3d%22%42%6c%61%63%6b%6c%69%73%74%65%64%20%34%31%31%2c
%49%73%72%61%65%6c%20%54%6f%72%72%65%73%22%20%74%69%74%6c%65%3d%22%47
%6f%6f%67%6c%65%20%53%65%61%72%63%68%22%3e%3c%69%6e%70%75%74%20%74%79
%70%65%3d%22%73%75%62%6d%69%74%22%20%76%61%6c%75%65%3d%22%47%6f%6f%67
%6c%65%20%53%65%61%72%63%68%22%20%6e%61%6d%65%3d%62%74%6e%47%3e%3c%2f
%66%6f%72%6d%3e
```

Once your form is encoded you will paste it into your html file between the single quotes in the **unescape(‘’)** function. You can also add more such as scripting as long as you make sure you add, and compress BEFORE you encode it.

After you have completed your crafted html file you want to be sure to test it locally first to get the results you are looking for. This works and has been tested on the latest versions of Internet Explorer as well as Mozilla Firefox.

Encoding Example

Ftard Decoder Ring is a Windows tool created for encoding and decoding text. You can paste your compressed form into the edit control and click ASC->Hex % to return the encoded form.



FTard Decoder Ring

Decode Encoded ASCII for Dissection and Analysis

```
<form action="http://www.google.com/search" name="f" method="put"><input type=hidden name=hl value=en><input
```

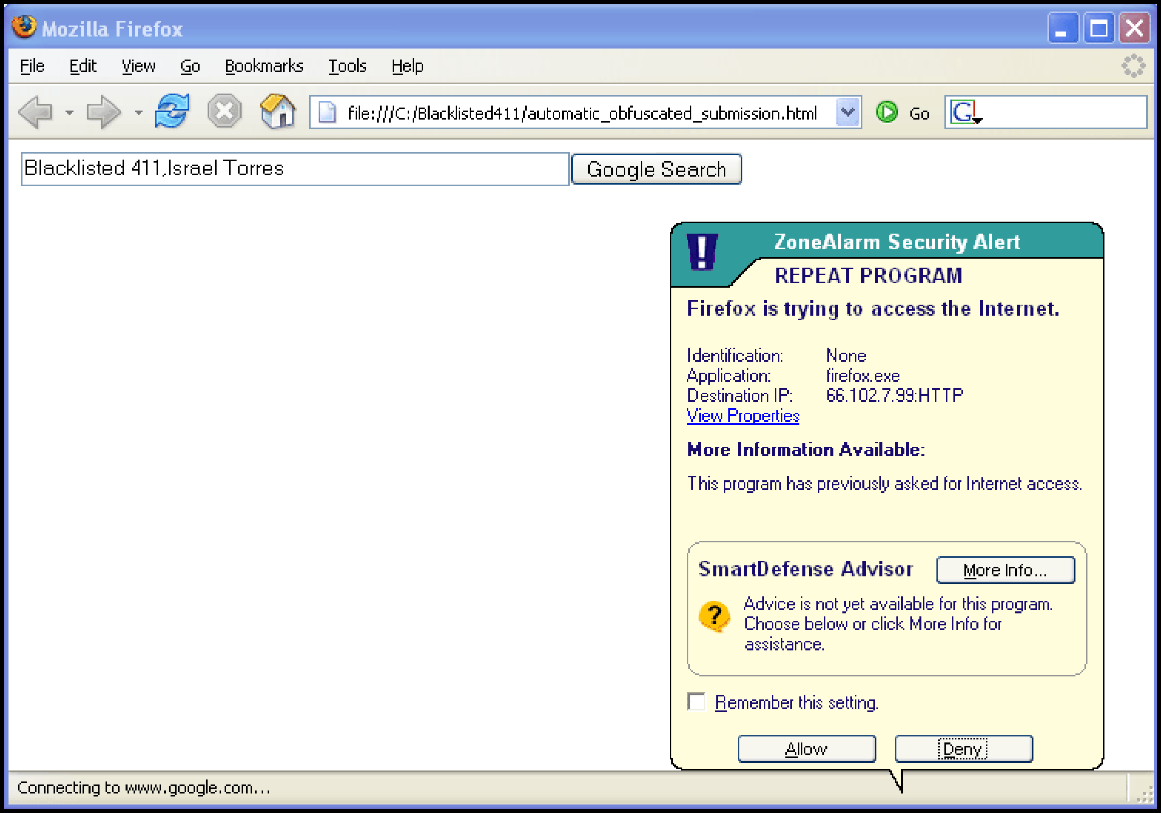
```
%3c%66%6f%72%6d%20%61%63%74%69%6f%6e%3d%22%68%74%74%70%3a%2f%2f%77%77%77%2e%67%6f%6f%6
```

Online Experience

After loading the html from a local system it attempts to execute the obfuscated string. If you have a software firewall running on your client system (*always a good idea*) you may encounter the following due to the first attempt to access the web. This case usually only will occur if this is the only instance of your browser that hasn't previously accessed the Internet within the same session.

Otherwise the user will quickly see the non-hidden submission field (you can use type="password so they see *** instead if so desired):

```
<input type="password" maxlength=2048 size=55 name=q value="Blacklisted 411,Israel Torres" title="Google Search">
```



file:///C:/Blacklisted411/automatic_obfuscated_submission.html

Blacklisted 411,Israel Torres

ZoneAlarm Security Alert

REPEAT PROGRAM

Firefox is trying to access the Internet.

Identification: None
Application: firefox.exe
Destination IP: 66.102.7.99:HTTP
View Properties

More Information Available:

This program has previously asked for Internet access.

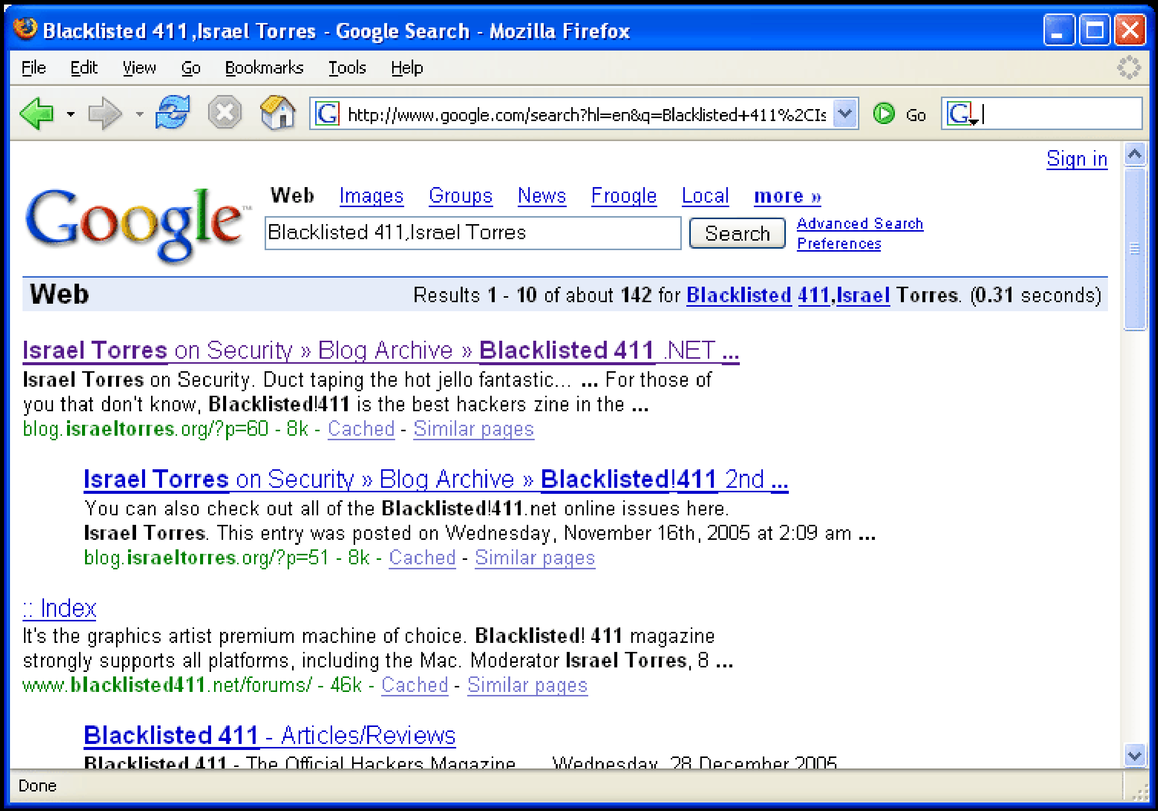
SmartDefense Advisor

Advice is not yet available for this program. Choose below or click More Info for assistance.

Connecting to www.google.com...

<u>All Results</u>

Regardless of how you load the html (local or web) it takes you to the same place.


Blacklisted 411,Israel Torres - Google Search - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com/search?hl=en&q=Blacklisted+411%2CIs
Go
Sign in
Web Images Groups News Froogle Local more »
Blacklisted 411,Israel Torres
Search
Advanced Search
Preferences
Done

Web Results 1 - 10 of about 142 for Blacklisted 411,Israel Torres. (0.31 seconds)

Israel Torres on Security » Blog Archive » Blacklisted 411 .NET ...
Israel Torres on Security. Duct taping the hot jello fantastic... ... For those of you that don't know, Blacklisted!411 is the best hackers zine in the ...
blog.israeltorres.org/?p=60 - 8k - Cached - Similar pages

Israel Torres on Security » Blog Archive » Blacklisted!411 2nd ...
You can also check out all of the Blacklisted!411.net online issues here. Israel Torres. This entry was posted on Wednesday, November 16th, 2005 at 2:09 am ...
blog.israeltorres.org/?p=51 - 8k - Cached - Similar pages

:: Index
It's the graphics artist premium machine of choice. Blacklisted! 411 magazine strongly supports all platforms, including the Mac. Moderator Israel Torres, 8 ...
www.blacklisted411.net/forums/ - 46k - Cached - Similar pages

Blacklisted 411 - Articles/Reviews
Blacklisted 411 - The Official Hackers Magazine ... Wednesday, 28 December 2005

Uses

There are many uses this can be modified for such as automating your personal logins, filling out forms; however its primary use is to automatically submit without really having a chance to see what is being submitted.

As users, filters and other forms of protection become more aware of these techniques they will be evolved by those that need to use them. It is a never ending cycle. I am always interested in learning about new techniques if you know one send it my way!

Keeping it ‘rael,
Israel Torres

Source Code

Source code bundle can be downloaded here:
http://blacklisted411.israeltorres.org/automatic_obfuscated_submission.zip

Individual Source code can be downloaded here:
http://blacklisted411.israeltorres.org/automatic_obfuscated_submission.html
http://blacklisted411.israeltorres.org/automatic_obfuscated_submission_worksheet.txt

MD5 hash of the source code bundle can be validated at:
http://blacklisted411.israeltorres.org/automatic_obfuscated_submission.md5.txt

note: you may have to copy and paste the addresses above if they wrap around!

Demonstration available at:
http://blacklisted411.israeltorres.org/automatic_obfuscated_submission.html

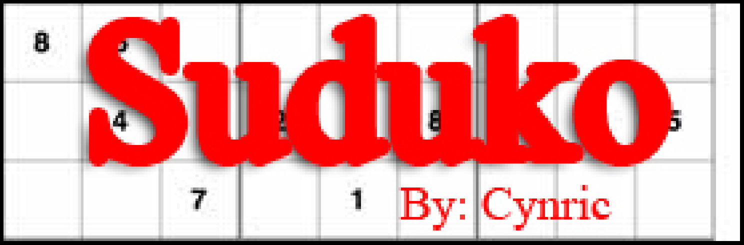
FTard Decoder Ring:

http://tools.israeltorres.org/#FTardDecoderRing

Google Desktop Issue (*relating to WMF*)
http://www.f-secure.com/weblog/archives/archive-122005.html

Spam and Phishing Obfuscation Part 1: Forum Attacks
http://www.blacklisted411.net/index.php?option=com_content&task=view&id=187&Itemid=66

[b] Suduko by Cynric



We've all seen this odd grid of numbers in newspapers, online, and books at our favorite store. Some of you may be utterly confused while others trod along at a slow pace. Hopefully, after reading this, both types will be whizzing through the more difficult problems with greater ease. The name Sudoku is the Japanese abbreviation of a longer phrase, "suji wa dokushin ni
kagiru," meaning "the digits must remain single".
[1] By the way, the [#]'s reference links mentioned at the end of the article.

Before we can get our feet wet, we need to lay down some basic terms that will be used throughout the article and in many online tutorials. If you have absolutely no idea what sudoku is, take a look at http://sudoku.com.au (an easy to use system with many features and an active community). "Cell" refers to a single space that a number would fall in. A "block" is a 3x3 group of Cells typically outlined darker than the Cells. "Rows" and "Columns" go in horizontal (that's left and right) and vertical (up and down), respectively.
There is also a general notation used to represent specific Cells. If you've ever played chess or Battleship, then you'll feel right at home. Each Cell in a Row is labeled with a letter of the alphabet from A-I. Each Cell in a Column is labeled with a number from 1-9; these numbers are NOT the same digits you are trying to fill in, but merely part of a coordinate system. So, when we want to address the bottom left Cell, we would say A1. Similarly for the top right, we address it as I9. Blocks are numbered from left to right, top to bottom.

The object here is to fill in the puzzle with missing numbers 1-9.
Each Row and Column must contain the digits 1-9. On top of that, each block must contain the digits 1-9. That's it, only two rules. Let's start with some basic techniques and work our way up to a walkthrough. If you have 8 of the 9 Cells filled in a block, the last cell must be the only digit you have yet to use. Similarly, if a Row or Column is missing a single Cell, then that missing Cell is the left over digit.
Example of a single Row:

1 2 3 | 4 5 6 | 7 _ 9

We know that the missing Cell (represented by the underscore "_") is an 8 since it's the only digit we've yet to use. Now, most of the time, to keep from juggling all these numbers in your head, you'll probably want to print out your puzzle and proceed to mark it up with possibilities. By using the fact that only the digits 1-9 can only appear once in each block, Row, and Column, we can create a list of possibilities for

each. When all but one possibility for a Cell has been removed (meaning that they appear somewhere else in the block, Row, or Column), then the remaining digit must be the correct entry.
This is generally referred to as the "Sole Candidate" since no other possibility legally exists. When the same scenario is extended to Rows and
Columns, it is known as the "Unique Candidate" for the same reasons. Let's take a look at another example of Rows:

```
1  2  3 | 7  8  9 | _  _  _
4  5  6 | 1  2  3 | _  _  _
_  _  _ | _  _  _ | _  _  _
```

In the above example, we can see that Cells A3, B3, and C3 contain 1, 2, and 3.
Also, D2, E2, and F2 contain 1, 2, and 3. This means that 1, 2, and 3 must go in G1, H1, and I1. Of course, we do not know the order in which they go, but we do know that they occupy some position of Row 1 in Block 3. What does this mean?
It means that the digits 1, 2, and 3 can NOT be in G2, H2, I2, G3, H3, or I3.
This type of rational is known as Block, Row, or Column (depending on which you're dealing with) Interactions.

The last basic technique that needs to be covered is called the "Naked
Subset"; no, you do not need to strip for this one. This technique is based on pairs. If two Cells in a block, Row, or Column have the same two candidates, then we know that those two candidates can be safely removed from the Cells in the block, Row, or Column depending on which you are using. In other words, if you have two Cells that exist in the same block, Row, or Column with the possible candidates of 1 and 9 each, then the 1 and 9 can not be anywhere else in that block, Row, or Column. Let's look at a short example:

```
1/9  _  _  |  _  _  _|  _  _  1/9
```

Let's say A1 is the 1. This means that I1 has to be the 9 since the same digit can not exist in the same Block, Row, or Column. This same logic applies if we had chosen A1 to be a 9 instead. Although we can't know from this alone which values A1 and I1 are, we do know that the digits 1 and 9 can not be anywhere else in the Row. If we were dealing with Blocks or Columns, the same idea applies. Pretty neat, eh?

Ok. I know this is difficult to follow in an article without an example. So, let's step through a tough puzzle [2] from my favorite online site [3].
Hopefully, in seeing my thought process, you'll gain a better understanding on how these techniques are used. You can also find proofs [4] on this site under "MORE" at the top. There are 1.76 million hits for "technique sudoku" on
Google so there should be plenty of examples to learn from. I found this [5] particular site to be fairly straightforward with decent examples.

Sudoku for 13/February/2006 from http://sudoku.com.au:

Figure 1:

```
_  _  6 | 3  _  _ | _  _  _
_  5  _ | 4  _  _ | _  9  _
7  _  _ | _  _  _ | _  _  2
```

```
---------------------------------
6 _ _ | _ 5 _ | 8 _ _
_ _ 3 | _ _ _ | 2 _ _
_ _ 1 | _ 8 _ | _ _ 6
---------------------------------
2 _ _ | _ _ _ | _ _ 8
_ 9 _ | _ _ 7 | _ 5 _
_ _ _ | _ _ 1 | 6 _ _
```

I suggest copying this to paper or just visiting the puzzle online so you can follow along. First off, I like to look for pairs that will eliminate some possibilities by using the Interaction rule. But, if any good ideas come before that, I won't hesitate to jump.

I'm feeling lucky with 8's. In Block 3, 8 is only possible at H9 and H7 due to an 8 being in Columns G (G6) and I (I3) already. In Block 4, an 8 can only be at A5 and B5 because of Rows 4 (E4) and 6 (G6). And in Block 8, only at D1 and D2 because of Row 3 (I3) and Column E (E4). In Block 1, only at A2 and C3. No luck so far. Ahah, in Block 9, we have a Sole Candidate for 2 at H1. Another Sole Candidate exists at B3 for a 6. Notice how Rows 1 and 3 have both 2 and 6.
This means that 2 and 6 go somewhere at D2 and E2. This good news means that since D2 contains either a 2 or 6, our 8 from before can not go there; thus,
D1 is 8. Because of the 1's on Row 1 (F1) and Column C (C4), we get a Sole Candidate for 1 at A2. This blocks our previously mentioned 8 from going there and therefore must go at C2.

Figure 2:

```
_ _ 6 | 3 _ _ | _ _ _
_ 5 _ | 4 _ _ | _ 9 _
7 _ _ | _ _ _ | _ _ 2

6 _ _ | _ 5 _ | 8 _ _
_ _ 3 | _ _ _ | 2 _ _
_ _ 1 | _ 8 _ | _ _ 6

2 6 _ | _ _ _ | _ _ 8
1 9 8 | _ _ 7 | _ 5 _
_ _ _ | _ _ 1 | 6 2 _
```

Let's do something a little more complex. Take a look at the intersection of E6 and G2. The only positions left for a 5 in Block 8 are at D3 and F3. Or, better yet, along Row 3. Now, take a look at the intersection of B8 and E6.
Possible positions for 5 in Block 4 are at A4 and A5 -- or, better yet, Column A. This tells us that Column A and B contain a 5. We also know that Row 2 and 3 contain a 5. This madness of Naked Subsets gives us a Sole Candidate for 5 at C1.

Looking at Column C, we see that digits 2, 4, 7, and 9 are left. Digit 2 is the only candidate at C8. Remember that 2 and 6 have taken up residence somewhere at D2 and E2? Good. That means that for Block 8, 3 can only be at
E1, E3, or F3. But, because of the 3 at C5, we know that 3 must go at A1 or B1 effectively eliminating E1 as a valid possibility. The intersection of D9 and C5 yield positions F4 and F6 as the only valid choices for 3 in Block 4. This just so happens to eliminate Column F giving us our Sole Candidate for 3 at E3 which we fought hard for. An easy Sole Candidate exists at H7 due to the 6's at C3, G1, and H4. This gives us another at H9 for an 8.

Figure 3:

```
_ _ 6 | 3 _ _ | _ 8 _
_ 5 2 | 4 _ _ | _ 9 _
7 _ _ | _ _ _ | _ 6 2

6 _ _ | _ 5 _ | 8 _ _
_ _ 3 | _ _ _ | 2 _ _
_ _ 1 | _ 8 _ | _ _ 6

2 6 _ | _ 3 _ | _ _ 8
1 9 8 | _ _ 7 | _ 5 _
_ _ 5 | 8 _ 1 | 6 2 _
```

Let's look at the combined intersection of 3 and 8 using C2, C5, D9, and H9. The only two positions for our 3 and 8 in Block 1 are somewhere at A8 and B7. Perfect. We now have a Sole Candidate for 1 at B9 thanks to the eliminations at A2 and C4. In Block 1, the only unclaimed values are 4 and 9 which will reside somewhere on A9 and C7. On Row 3, a 4 is only possible at C3 and F3. If C3 is 4 then C7 must be 9. Or, if F3 is 4, then E1 must be 9 because the only possible values for it were 4 and 9. Whichever way it is, the 9 being at C7 or E1 forbids a 9 from showing up at E7. If you look at all the possible values for E7, you'll see that only one digit remains -- a 1. Since a 1 can no longer show up in Block 2, possible solutions for D7 get reduced to a 5 or 9. We also have a 5 and 9 as possible values for D3. Alright, let's make an assumption about C3. It's going to take us 9 more steps, but afterwards, we will have a solution for C3 that reduces the entire puzzle to Sole and Unique Candidates. Pick up your pencils, here we go.

Figure 4:

```
      _ 1 6 | 3 _ _ | _ 8 _
      _ 5 2 | 4 _ _ | _ 9 _
      7 _ _ | _ 1 _ | _ 6 2
  ______________________________
      6 _ _ | _ 5 _ | 8 _ _
      _ _ 3 | _ _ _ | 2 _ _
      _ _ 1 | _ 8 _ | _ _ 6
  ______________________________
      2 6 _ | _ 3 _ | _ _ 8
      1 9 8 | _ _ 7 | _ 5 _
      _ _ 5 | 8 _ 1 | 6 2 _
```

Let's assume C3 is 7. We now get C7 as being 4. With C5 and C7 eliminating 3 and 4 on Row 3, it must mean the 3 and 4 go somewhere at A1 and B1. Coupled with our 3 and 4 pair at G2 and I2, we can eliminate Row 2 as well. This leaves us with only one place in Block 8 for the 4 to go; F3. Because of the 5 at H2, we get a Unique Candidate for 5 at D3 because D2 and E2 are still reserved for digits 2 and 6. This means D7 is 9 -- remember that pair we mentioned earlier? Due to H8, we get a Unique Candidate on Row 3 for a 9 at G3. H3 becomes a 1 and I1 becomes 7. Using the intersection of H8 and G3, the only place a 9 can fit in Block 6 is I5 or I6. Using the intersection of C4 and H3, the only place a 1 can fit in Block 6 is I5 or I6. On Row 9, the only position open for a 5 is at I9. The only position for 4 is now at I2. This gives us a 3 at G2 and I8, a 4 at G9, a 7 at G8 and ... oh ... wait a minute. A 1 at G7? Uh uh. This would conflict with our 1 at E7 which we know is correct. Therefore, we have finally deduced that C3 is not 7. You might be thinking I'm crazy at this point, but we have actually just broken the puzzle down to simple Sole and Unique Candidates; well, if you were able to keep up with this that is.

Now that we know C3 is 4, it follows that B1 is 7 and A is 3. Look at Column C, C6 must be 7 which leaves C7 as 9. Getting C7 nails down A9 as 4 -- remember that they were paired? The 4 at A9 and D8 tells us that G7 is the only valid position for a 4 in Block 3. Remember that 3 and 4 pair at G2 and I2? Well, we finally get to solve it with G2 being 3 leaving I2 as 4. That 4 and the one at C3 gives us the value of E1 as 4. The only digit left on Row 1 is a 9 at I1. It, with the 9 at H8, yields a 9 at G4. The intersection of G4 and B2 leave only A5 available for the 9 in Block 4. A 5 is already in Block 1, so

A4 must be 5 on Column A with A8 being 8.

Figure 5:

```
            4 1 6 | 3 _ _ | _ 8 _
            8 5 2 | 4 _ _ | _ 9 _
            7 3 9 | _ 1 _ | 4 6 2
      ------------------------------------
            6 _ 7 | _ 5 _ | 8 _ _
            9 _ 3 | _ _ _ | 2 _ _
            5 _ 1 | _ 8 _ | 9 _ 6
      ------------------------------------
            2 6 4 | _ 3 _ | _ _ 8
            1 9 8 | _ _ 7 | 3 5 4
            3 7 5 | 8 4 1 | 6 2 9
```

The last Cell in Block 1 is a 3. E4 and G6 leave only B5 open for an 8. D1 and E4 only leave F7 open for an 8 on Row 7. That means D7 is 5 and D3 is 9 -- back to using those pairs again. The only Cell in Block 8 not reserved is F3 which becomes 5. Looking at Block 2, we see that the 2 at C8 and the 9 at H8 eliminate those values along Row 8, forcing them to be somewhere on Row 9. With that Row reserved, the 7 at F1 conveniently positions a 7 at E8. This leaves a 6 on F8. The possibilities along Row 8 get reduced by the 3 at G2. Now that we have our 3 at I8, we can safely fill in a 1 at G8. The intersection of E6 and H2 put a 5 at I5. This pairs up with Block 3 and gives us a 5 at G9 as well. Block 3 is finished off with a 7 at I9 leading to polishing off Column I with a 1 at I6. To finish off Column G and Block 9, we get a 7 and 1, respectively.

Figure 6:

```
4 1 6 | 3 _ _ | 5 8 7
8 5 2 | 4 7 6 | 1 9 3
7 3 9 | 5 1 8 | 4 6 2

6 _ 7 | _ 5 _ | 8 _ 1
9 8 3 | _ _ _ | 2 _ 5
5 _ 1 | _ 8 _ | 9 _ 6

2 6 4 | _ 3 _ | 7 1 8
1 9 8 | _ _ 7 | 3 5 4
3 7 5 | 8 4 1 | 6 2 9
```

Friendly 9's at A5, D3, and G4 solidify a 9 at F6 which forbids a 9 at F9. We get 9 at E9 and 2 at F9. The intersection of F9 and G5 puts the 2 for Block 5 on Column D. This gives us a 2 at E2 and therefore a 6 at D2. Column E is completed with a 6 at E5. Column F is completed with F4 as 3 and F5 as 4 after noticing C5. H6 is 3 thanks to C5 and F4. F5 forbids a 4 at H5 placing it at H4. Block 6 is finished with a 7 that couples with C6 to place a 7 at D4. I6 gives a 1 for D5 with a 2 finishing off Block 5. Block 4 is completed with a 2 and 4 at B4 and B6, respectively. That's it! Whew, what a job. The final solution is below followed by some links.

Figure 7:

```
4 1 6 | 3 9 2 | 5 8 7
8 5 2 | 4 7 6 | 1 9 3
7 3 9 | 5 1 8 | 4 6 2

6 4 7 | 2 5 9 | 8 3 1
9 8 3 | 1 6 4 | 2 7 5
5 2 1 | 7 8 3 | 9 4 6

2 6 4 | 9 3 5 | 7 1 8
1 9 8 | 6 2 7 | 3 5 4
3 7 5 | 8 4 1 | 6 2 9
```

==================

[1] http://en.wikipedia.org/wiki/Sudoku
[2] http://sudoku.com.au/1V13-2-2006-sudoku.aspx
[3] http://sudoku.com.au
[4] http://sudoku.com.au/MoreSudoku.aspx
[5] http://www.simes.clara.co.uk/programs/sudokutechniques.htm

[c]

Hacking Cryptograms 101
In Digestion

By Israel Torres <israel@israeltorres.org>

Welcome to the **fourth** lesson in hacking cryptograms. This .NET column engages in forms of hacking cryptograms (*secret messages)* found in everyday transactions. Commonly known cryptograms are seen in the daily newspaper and various websites that focus on mono-alphabetic substitution. However any text that contains a secret or hidden message can be a cryptogram.

In Edition 4 of .NET we had two visible cryptograms and one hidden cryptogram to help you solve the visible cryptograms. The best way to hide something is to put it in plain sight.

Cryptogram A (*beginner*)
5EBE2294
ECD0E0F0
8EAB7690
D2A6EE69

Cryptogram B (*advanced*)
1814E0A5
AF374437
03D7EE4C
87EFA6BA

Hint: There is no need to GET the SID to solve this cryptogram. You also shouldn't need Bravo Foxtrot.

I've been informed by the .NET editor that no one solved this particular set (shame * shame) – If they did they didn't submit the answer to us for recognition. In lieu of the last two concluding cryptograms that are posted at the end of the article not being solved by readers the .NET editor is planning to come up with something else (last I heard a type of puzzle). Worry not dear readers I will simply involve a cryptogram in this column for entertainment purposes. As part of the format change you can email me directly the solutions you have and I will talk about them the next time around.

Solution: The hidden hints in the Hint!

Allow me to demonstrate the hidden cryptogram first. This cryptogram was hidden in the hint (*of all places*):

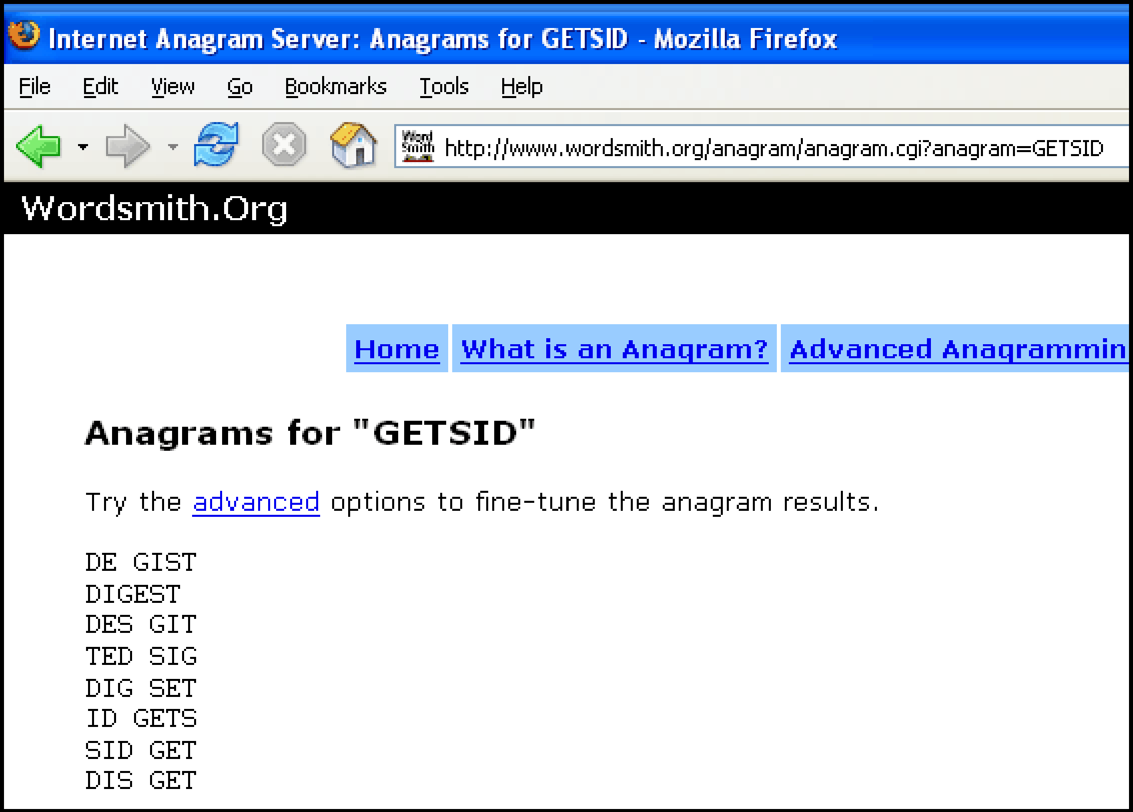
Hint: There is no need to GET the SID to solve this cryptogram. You also shouldn't need Bravo Foxtrot.

I embedded the hint with a pseudo-anagram (rearranged letters).

GET the SID

GET SID is extracted turned into one word **GETSID**. If this still isn't obvious you can put this anagram into an anagram analyzer. One that has been around for a long time is the Internet Anagram Server available on wordsmith.org.

http://www.wordsmith.org/anagram/anagram.cgi?anagram=GETSID


Internet Anagram Server: Anagrams for GETSID - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.wordsmith.org/anagram/anagram.cgi?anagram=GETSID
Wordsmith.Org
Home
What is an Anagram?
Advanced Anagrammin

Anagrams for "GETSID"

Try the advanced options to fine-tune the anagram results.

```
DE GIST
DIGEST
DES GIT
TED SIG
DIG SET
ID GETS
SID GET
DIS GET
```

In which the following results are returned:

DE GIST
DIGEST
DES GIT
TED SIG

DIG SET
ID GETS
SID GET
DIS GET

Using the results you can quickly figure out there is only one word returned (DIGEST):

DE GIST
DIGEST
DES GIT
TED SIG
DIG SET
ID GETS
SID GET
DIS GET

This tells you that the each cryptogram is a digests of the message commonly referred to as message digest (MD) hash.

The second part of this hidden message is the following:

You also shouldn't need Bravo Foxtrot.

This is a message telling you that you shouldn't need Bravo Foxtrot. What the heck is Bravo Foxtrot? If you type it in Google the results will involve sites using alpha-phonetic coding.

Bravo **F**oxtrot.

The phonetic code Bravo Foxtrot represents two letters – B and F (BF). This one is more difficult to grasp if crypto isn't your thing (or maybe even if it is). **BF** is another way to declare **Brute Force** attacks. Knowing this you can rewrite the statement as:

You also shouldn't need Brute Force

This tells you right away that this message digest can be cracked using a dictionary attack – even better a common word or set of words in a normal dictionary!

Solution: Cryptogram A (*beginner*)

If you didn't consider the hint in part of considering the solution you may have ended up trying to apply various attacks on the block of text which would have led you astray. You must remember not to take for granted what is being presented to you. At the same time do not wholly trust what you are being allowed to see as what is to be seen. I'd throw in a *matrixism* here but I won't ;)

Now that we know we are dealing with a message digest this part is a lot easier. Since this is the beginner version it is pretty straightforward.

Cryptogram A (*beginner*)
5EBE2294

```
ECD0E0F0
8EAB7690
D2A6EE69
```

By simply removing the line break and carriage return we turn this into its original format.

Cryptogram A (*beginner*)
5EBE2294ECD0E0F08EAB7690D2A6EE69

The idea of a message digest algorithm is that you can apply the mechanism to result in a unique hash generated by the computation of the original data. There have been recent studies to show that this may not be unique as originally thought but that is neither here nor there for this focus. In any case this is still a popular method to use data values.

At this stage you may not be sure what type of message digest this is. Could it be MD4, MD5, MD* ? You will assume that it is the more popular MD5 and see if this proves true.

You have a few options once you have a hash you determine to be MD5. Listed in the reverse order of difficulty they are but not limited to:

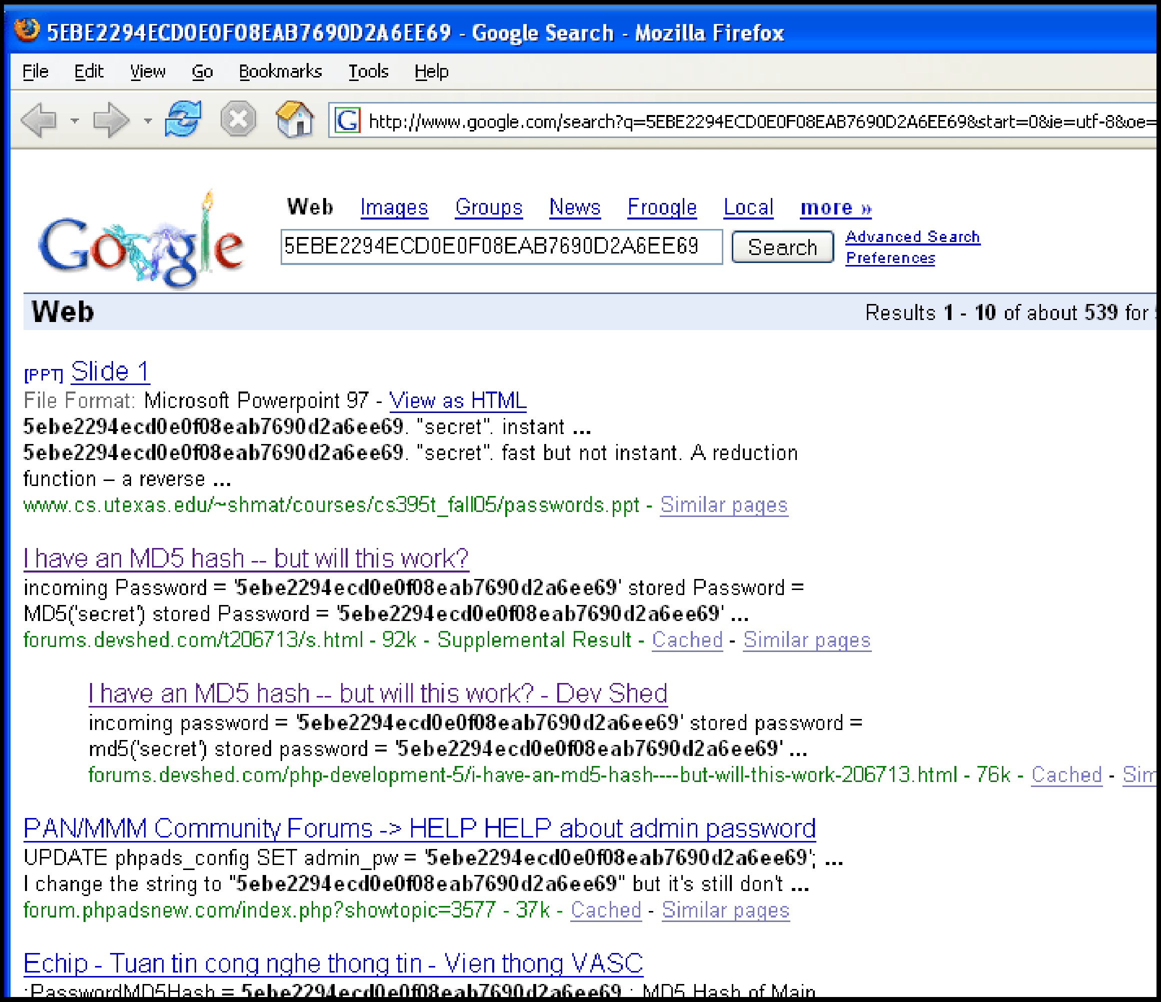
1. Find the original text by searching the hash in Google.
2. Use online hash crackers.
3. Use your own rainbow tables.
4. Use cracking tools locally.
5. Write your own tools.

You can look into each option quickly to see which gets you the results you need.

1. Finding the original text by searching the hash in Google

Copy and paste the hash to your Google toolbar or browser and begin your search. With this particular hash I found a goldmine of results:

Results 1 - 10 of about 539 for 5EBE2294ECD0E0F08EAB7690D2A6EE69. (0.45 seconds)


5EBE2294ECD0E0F08EAB7690D2A6EE69 - Google Search - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com/search?q=5EBE2294ECD0E0F08EAB7690D2A6EE69&start=0&ie=utf-8&oe=
Web Images Groups News Froogle Local more »
5EBE2294ECD0E0F08EAB7690D2A6EE69
Search
Advanced Search
Preferences

Web Results **1** - **10** of about **539** for

[PPT] Slide 1
File Format: Microsoft Powerpoint 97 - View as HTML
5ebe2294ecd0e0f08eab7690d2a6ee69. "secret". instant **...**
5ebe2294ecd0e0f08eab7690d2a6ee69. "secret". fast but not instant. A reduction
function – a reverse **...**
www.cs.utexas.edu/~shmat/courses/cs395t_fall05/passwords.ppt - Similar pages

I have an MD5 hash -- but will this work?
incoming Password = '**5ebe2294ecd0e0f08eab7690d2a6ee69**' stored Password =
MD5('secret') stored Password = '**5ebe2294ecd0e0f08eab7690d2a6ee69**' **...**
forums.devshed.com/t206713/s.html - 92k - Supplemental Result - Cached - Similar pages

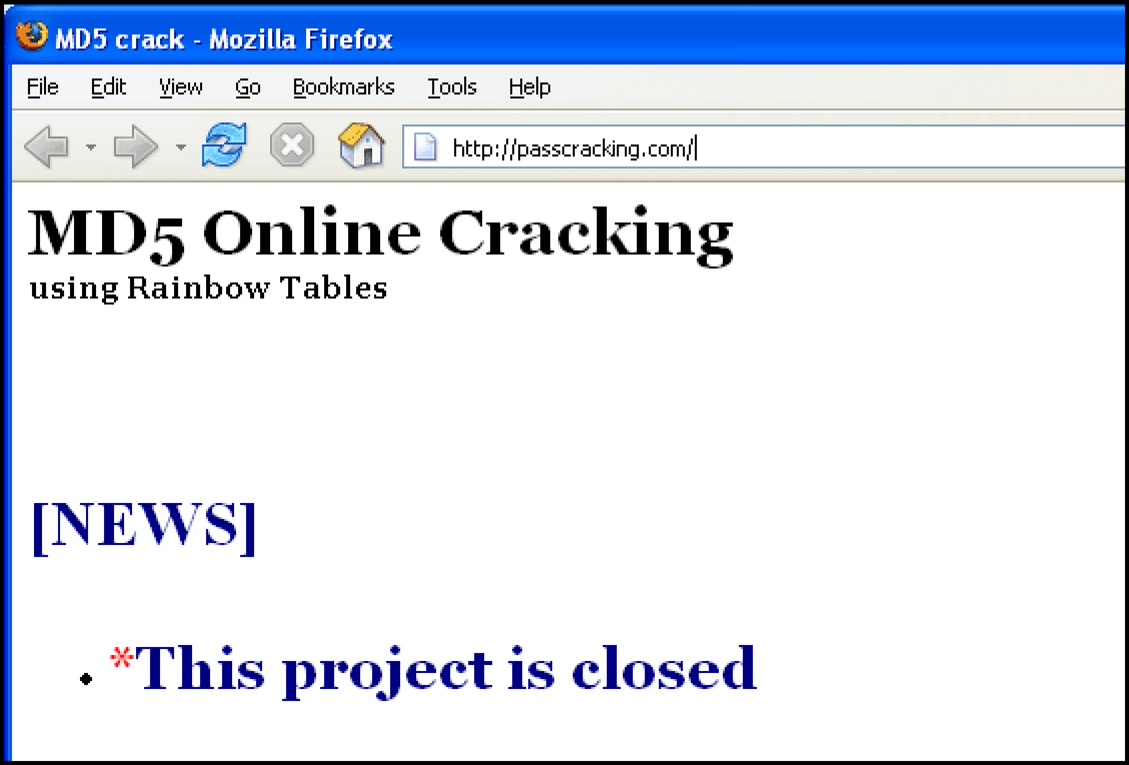
I have an MD5 hash -- but will this work? - Dev Shed
incoming password = '**5ebe2294ecd0e0f08eab7690d2a6ee69**' stored password =
md5('secret') stored password = '**5ebe2294ecd0e0f08eab7690d2a6ee69**' **...**
forums.devshed.com/php-development-5/i-have-an-md5-hash----but-will-this-work-206713.html - 76k - Cached - Sim

PAN/MMM Community Forums -> HELP HELP about admin password
UPDATE phpads_config SET admin_pw = '**5ebe2294ecd0e0f08eab7690d2a6ee69**'; **...**
I change the string to "**5ebe2294ecd0e0f08eab7690d2a6ee69**" but it's still don't **...**
forum.phpadsnew.com/index.php?showtopic=3577 - 37k - Cached - Similar pages

Echip - Tuan tin cong nghe thong tin - Vien thong VASC
;PasswordMD5Hash = **5ebe2294ecd0e0f08eab7690d2a6ee69** ; MD5 Hash of Main

2. Using an online hash cracker

There are a handful of free public cracking sites available to do all the work for you. The first one that comes to mind is passcracking.com:


MD5 crack - Mozilla Firefox

http://passcracking.com/

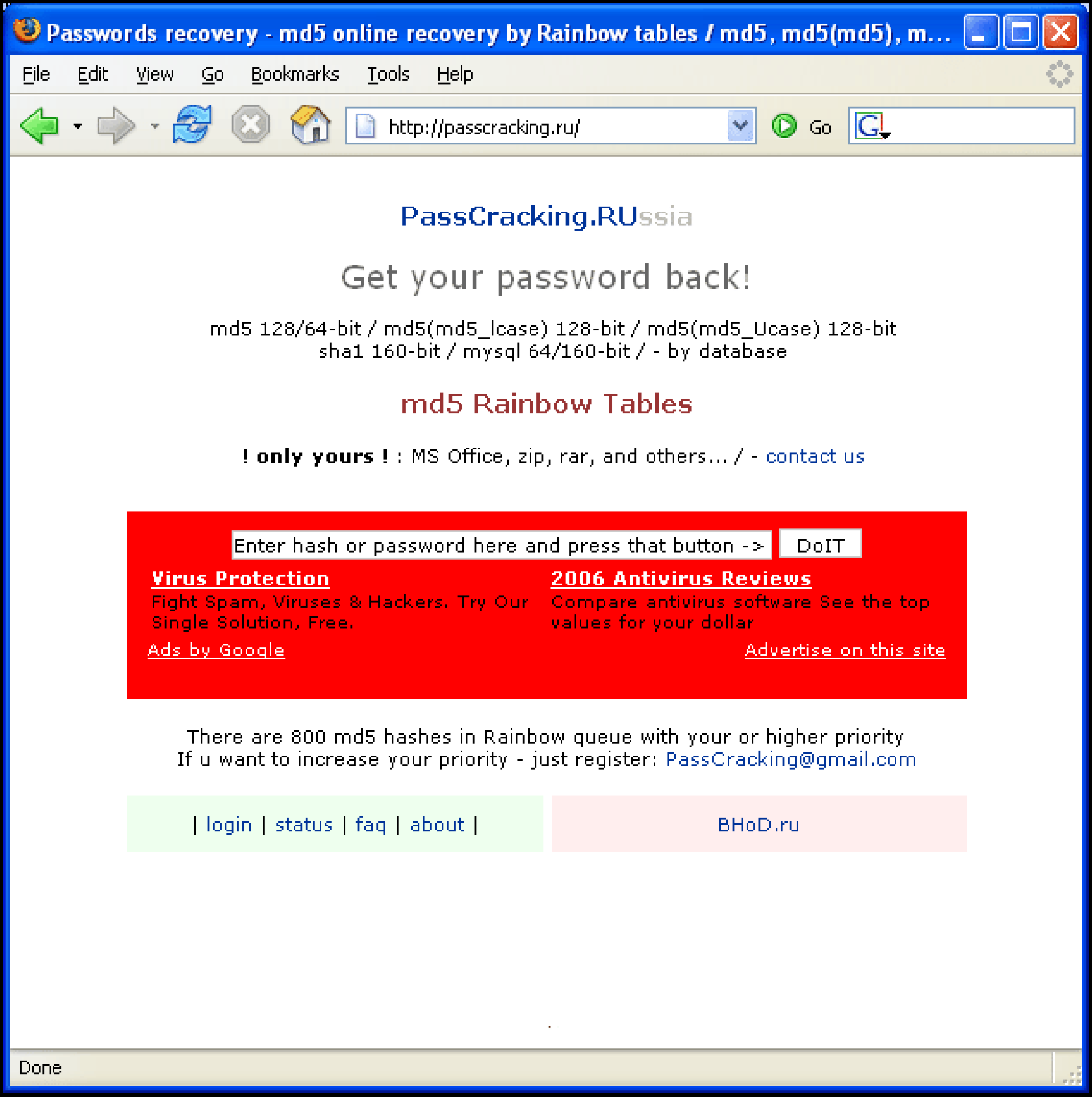
MD5 Online Cracking
using Rainbow Tables

[NEWS]

- *This project is closed

… However this once wildly heralded site has been closed. There is little mention in most links that refer this site so chances are you are bound to get your hopes up that they are still operational.

Though there is still hope (for now) with passcracking.ru.


Passwords recovery - md5 online recovery by Rainbow tables / md5, md5(md5), m...
File Edit View Go Bookmarks Tools Help
http://passcracking.ru/
Go
Done

PassCracking.RUssia

Get your password back!

md5 128/64-bit / md5(md5_lcase) 128-bit / md5(md5_Ucase) 128-bit
sha1 160-bit / mysql 64/160-bit / - by database

md5 Rainbow Tables

! only yours ! : MS Office, zip, rar, and others... / - contact us

Enter hash or password here and press that button -> DoIT

Virus Protection
Fight Spam, Viruses & Hackers. Try Our Single Solution. Free.

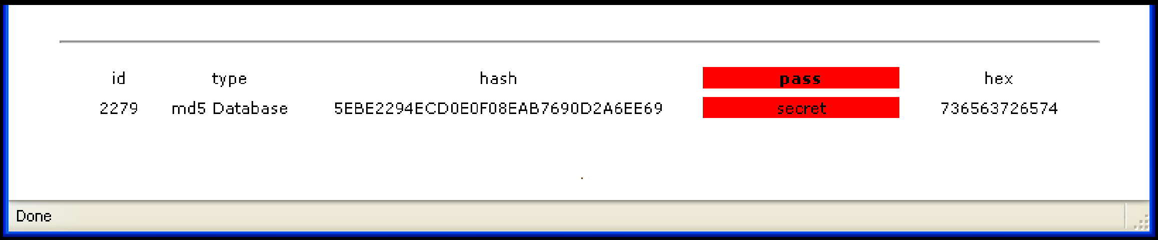
2006 Antivirus Reviews
Compare antivirus software See the top values for your dollar

Ads by Google Advertise on this site

There are 800 md5 hashes in Rainbow queue with your or higher priority
If u want to increase your priority - just register: PassCracking@gmail.com

| login | status | faq | about | BHoD.ru

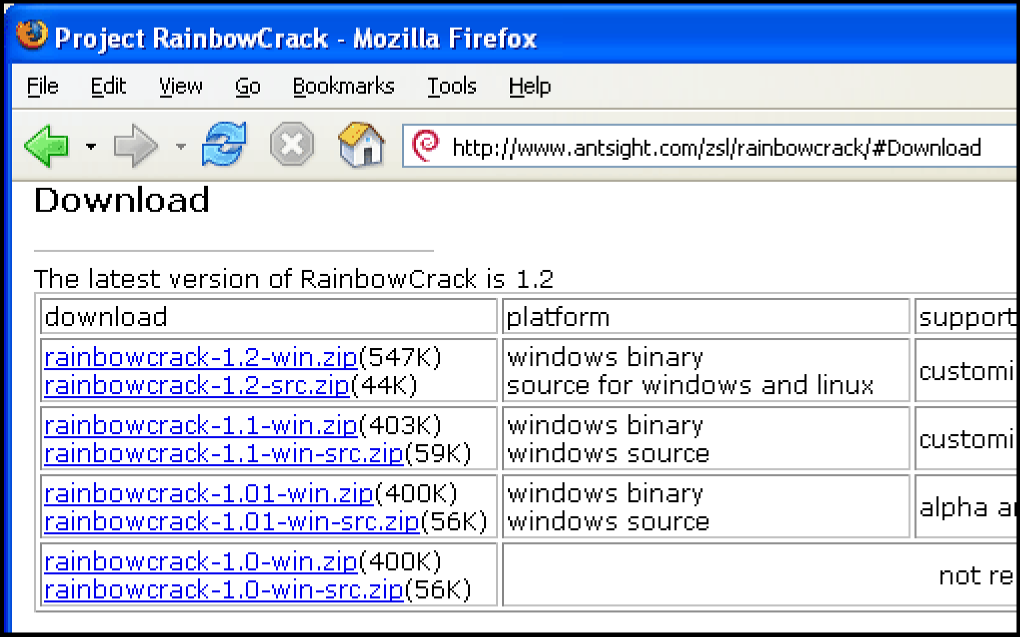
Once the site renders go ahead and copy and paste the hash and click DoIT. Almost instantly the results come back positive!



id	type	hash	pass	hex
2279	md5 Database	5EBE2294ECD0E0F08EAB7690D2A6EE69	secret	736563726574

5EBE2294ECD0E0F08EAB7690D2A6EE69 secret

3. Using your own rainbow tables.
Project RainbowCrack


Project RainbowCrack - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.antsight.com/zsl/rainbowcrack/#Download

Download

The latest version of RainbowCrack is 1.2

download	platform	support

rainbowcrack-1.2-win.zip(547K)
rainbowcrack-1.2-src.zip(44K)
	windows binary
	source for windows and linux
		customi

rainbowcrack-1.1-win.zip(403K)
rainbowcrack-1.1-win-src.zip(59K)
	windows binary
	windows source
		customi

rainbowcrack-1.01-win.zip(400K)
rainbowcrack-1.01-win-src.zip(56K)
	windows binary
	windows source
		alpha a

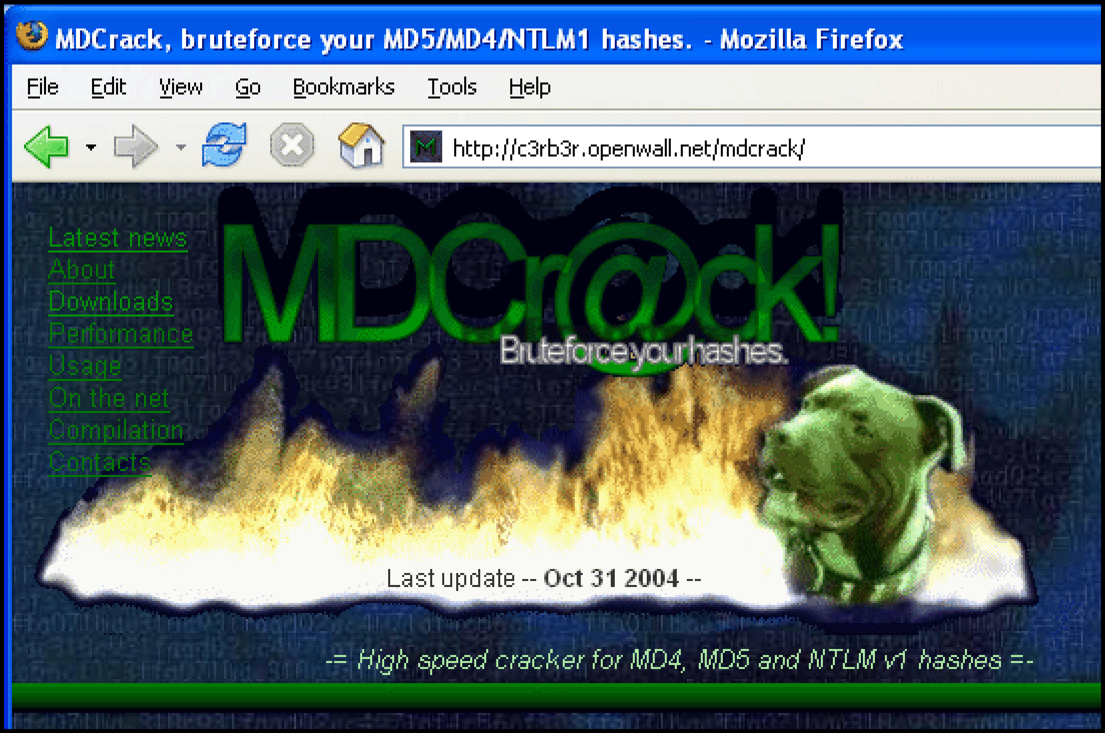
rainbowcrack-1.0-win.zip(400K)
rainbowcrack-1.0-win-src.zip(56K)
	not re	

You may have noticed that both of the sites listed in 2 above get their results using rainbow tables. The Russian site even offers priority accounts; there are even more private sites that offer pay for priority services to use their servers – as you can expect it can get expensive. You can check out Project RainbowCrack and generate your own set of tables to whistler through cracking MD5 hashes (and a handful of others). Though after you read the information as to the disk space required to generate a decent table signing up for a remote subscription doesn't sound too bad. Being self-sufficient I can understand spending a few bucks and carry around a USB drive filled only with a set of tables and maybe even a front end. If you end up going cheap and not thoroughly covering enough characters you have a higher chance to miss the solution. It is well worth it to wait a few months of 24/7 table generation and sorting to get a clean thorough table.

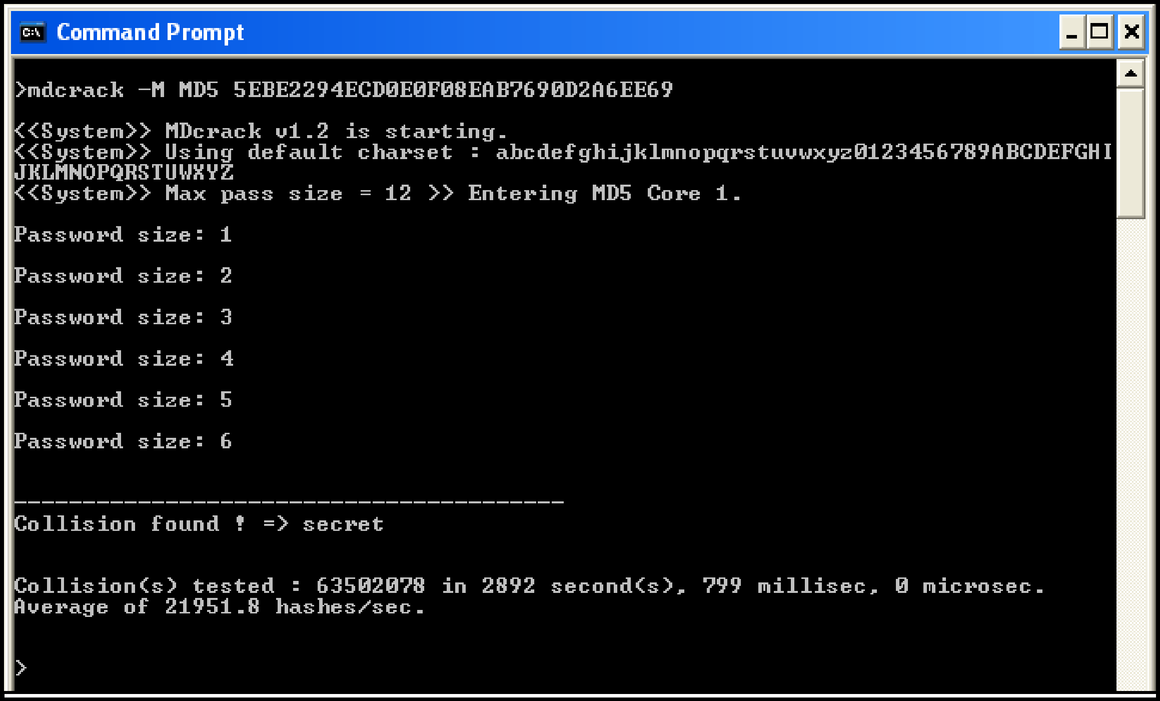
4. Using cracking tools locally.

MDrack v1.2

MD Crack is one of my favorite all time message digest cracking tools. This particular version was last updated back in 2004 but it still keeps on cracking!


MDCrack, bruteforce your MD5/MD4/NTLM1 hashes. - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://c3rb3r.openwall.net/mdcrack/
Latest news
About
Downloads
Performance
Usage
On the net
Compilation
Contacts
MDCr@ck!
Bruteforce your hashes.
Last update -- Oct 31 2004 --
-= High speed cracker for MD4, MD5 and NTLM v1 hashes =-

The worst downside of this local application is that it seriously hogs your resources until you are done. The second I began the process for the demo below my MP3s stopped playing and turned into chunk.



```
>mdcrack -M MD5 5EBE2294ECD0E0F08EAB7690D2A6EE69

<<System>> MDcrack v1.2 is starting.
<<System>> Using default charset : abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHI
JKLMNOPQRSTUWXYZ
<<System>> Max pass size = 12 >> Entering MD5 Core 1.

Password size: 1

Password size: 2

Password size: 3

Password size: 4

Password size: 5

Password size: 6


----------------------------------------
Collision found ! => secret


Collision(s) tested : 63502078 in 2892 second(s), 799 millisec, 0 microsec.
Average of 21951.8 hashes/sec.


>
```

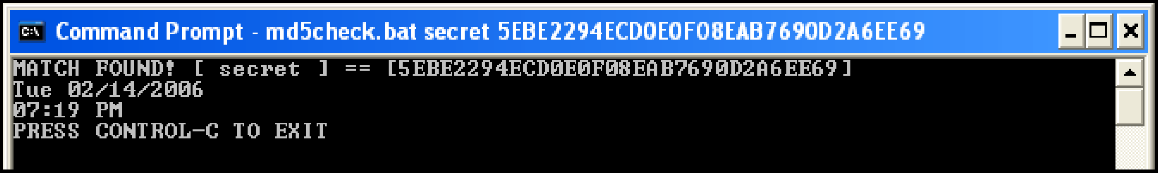
This hash took a little under an hour to find using brute force (**BF**):

Collision found ! => secret

It would have taken a lot less time to use a dictionary file, but this version doesn't support it at this time *(I demonstrate how to make your own in the next section).*

4. Writing your own cracking tools.

I wrote the following tool (batch files) in a matter of seconds with a few more seconds of testing to get it hammered into place. I call it a poor man's way to crack MD5 hashes. In essence it isn't the hash being cracked but the method applied is cracking the solution by hashing each word found in our custom dictionary and then comparing it to the hash sought to be cracked. (*The source code is available in the coding section of this article as well as online as linked at the end of this article – I have also created a BF generator but that is not within scope of this document.)*



Command Prompt - md5check.bat secret 5EBE2294ECD0E0F08EAB7690D2A6EE69

```
MATCH FOUND! [ secret ] == [5EBE2294ECD0E0F08EAB7690D2A6EE69]
Tue 02/14/2006
07:19 PM
PRESS CONTROL-C TO EXIT
```

After checking each hashed entry against the target hash we find the match for Cryptogram A. This all really depends on your dictionary file so in theory the more complete dictionary file you have the more of a chance you have of encountering a match at the same time the longer it will take to search through.

After a few seconds of hashing the dictionary file we get match – a lot shorter than the brute force approach with MDCrack above.

MATCH FOUND! [secret] == [5EBE2294ECD0E0F08EAB7690D2A6EE69]

Now let's get to the fun part where I demonstrate how you can do this programmatically. This way if you run into this type of cryptogram again you can use this tool to quickly find the solution. If you aren't into coding, I suggest you at least look it over to get a feel of the logic involved (*this source is available online via the link below*). If you simply refuse to even look at it skip the following example and **GOTO NO_CODING();**

md5demo.txt

```
::::::::::::::::::::::
:: Blacklisted!411 .NET Edition 5
:: Hacking Cryptograms 101 : In Digestion
:: By Israel Torres <israel@israeltorres.org>
:: Source available at: http://blacklisted411.israeltorres.org
:::::::::::::::::::::::::::::::::

This simple demonstration requires the four files listed below:
-the first three are available at http://blacklisted411.israeltorres.org

dictionary1.txt (included)      CC722A145A3565C50C8218EEDF6DED68
md5crack.bat  (included)        3CA9862E7FECF952F4BF800DA29CA868
md5check.bat  (included)        05ECE55F6C13A8827ECE14C9BA461D32
md5.exe (available from http://www.fourmilab.ch/md5/)
609F46A341FEDEAEEC18ABF9FB7C9647

Control Flow:
md5crack.bat calls md5check.bat after opening dictionary1.txt
```

md5check.bat calls md5.exe to generate hash against dictionary1.txt and using find and errorlevel to identify match

Release Notes: To keep this simple there is minimal error control please make sure required files are available in same directory, or path.

MD5Crack.bat

```
@ECHO OFF
::::::::::::::::::::::
:: Blacklisted!411 .NET Edition 5
:: Hacking Cryptograms 101 : In Digestion
:: By Israel Torres <israel@israeltorres.org>
:: Source available at: http://blacklisted411.israeltorres.org
::::::::::::::::::::::::::::::::::
:: Please read md5demo.txt for instructions and usage
:::::::::::::::::::::
:: 1/2 MD5Crack.bat
:::::::::::::::::::::
prompt $g

:: Display when this process began
DATE /T
TIME /T

::For each string in our dictionary file we want to check it against this hash
::
FOR /F %%A IN (dictionary1.txt) DO md5check.bat %%A
5EBE2294ECD0E0F08EAB7690D2A6EE69

::
:: END
```

MD5Check.bat

```
@ECHO OFF
::::::::::::::::::::::
:: Blacklisted!411 .NET Edition 5
:: Hacking Cryptograms 101 : In Digestion
:: By Israel Torres <israel@israeltorres.org>
:: Source available at: http://blacklisted411.israeltorres.org
::::::::::::::::::::::::::::::::::
:: Please read md5demo.txt for instructions and usage
:::::::::::::::::::::
:: 2/2 MD5Check.bat
```

```
::::::::::::::::::::::

md5 -d%1 | find /i "%2"

IF ERRORLEVEL 1 GOTO FOUND_NOT
IF ERRORLEVEL 0 GOTO FOUND_MATCH
GOTO FOUND_NOT

:FOUND_MATCH
CLS
ECHO MATCH FOUND! [ %1 ] == [%2]
:: Display when this process ended successfully
DATE /T
TIME /T
ECHO PRESS CONTROL-C TO EXIT
PAUSE > NUL
GOTO END_BATCH

:FOUND_NOT
ECHO MATCH _NOT_ FOUND
GOTO END_BATCH

:END_BATCH
:: Display when this process ended
DATE /T
TIME /T
```

These batch files can be refined for greater granularity and control but it gets the job done effectively.

NO_CODING()

That was easy enough… but what about Cryptogram B? It is solved in the same fashion with the only exception being that you need a bigger dictionary or more time for brute forcing it since it is two words (literally - without a space in between them). Here is a summary of steps:

Step 0.

Cryptogram B (*advanced*)

```
1814E0A5
AF374437
03D7EE4C
87EFA6BA
```

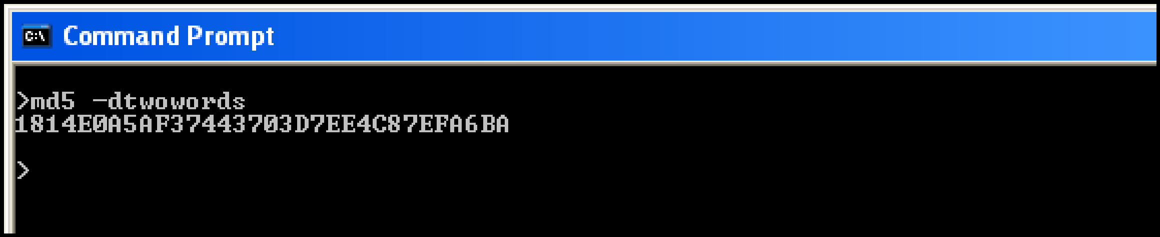
Step 1.

Cryptogram B (*advanced*)

```
1814E0A5AF37443703D7EE4C87EFA6BA
```

Step 2.

```
>md5 -dtwowords
1814E0A5AF37443703D7EE4C87EFA6BA
```

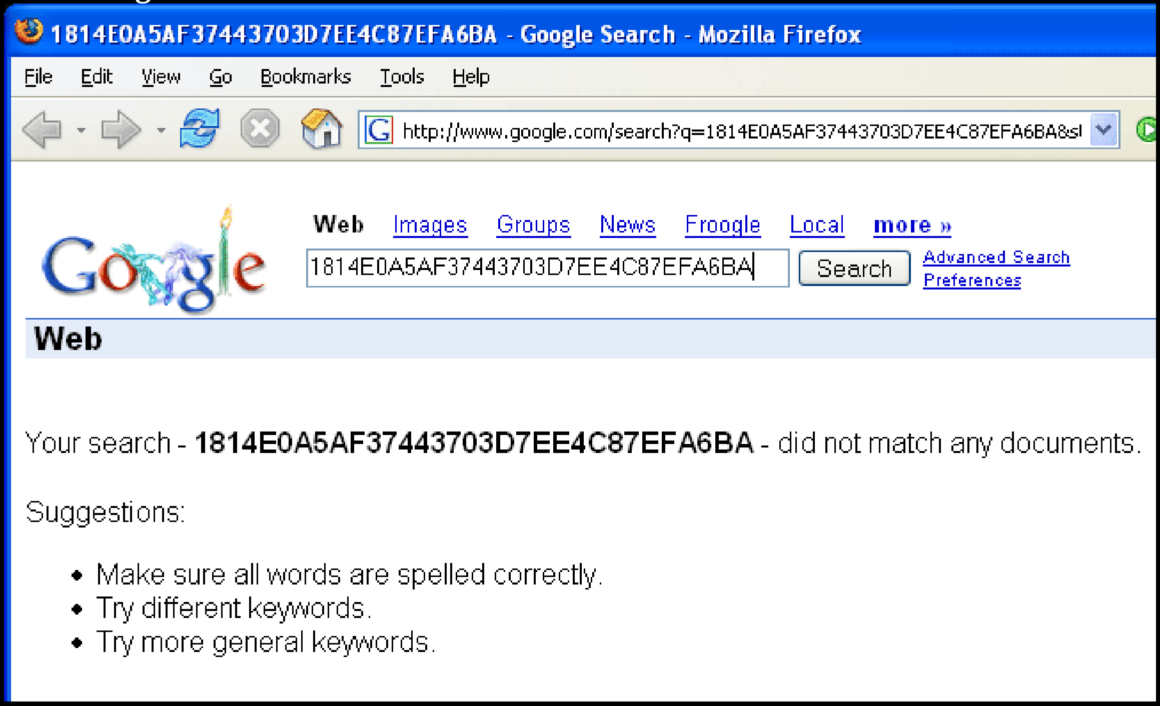


```
>md5 -dtwowords
1814E0A5AF37443703D7EE4C87EFA6BA

>
```

For this article I went to check the sites I mentioned in helping crack this online.

First Google…



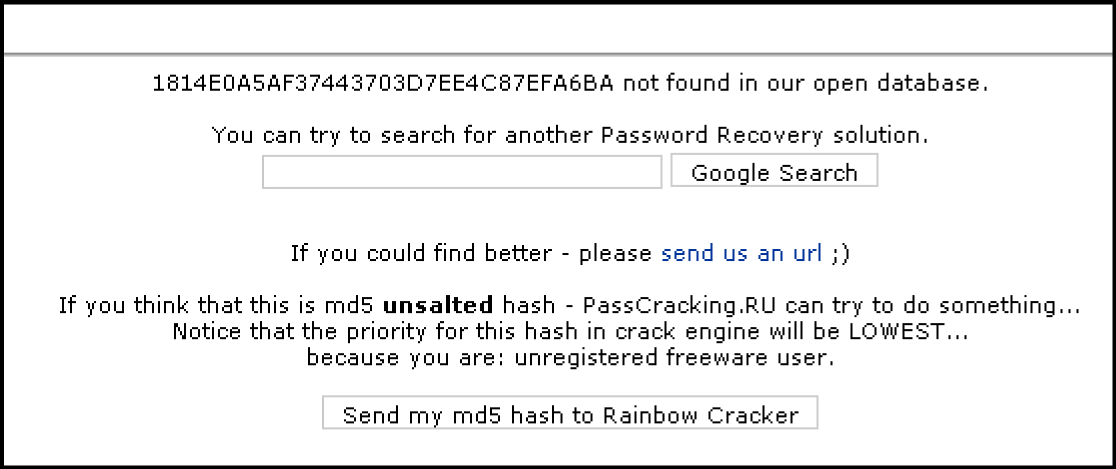
1814E0A5AF37443703D7EE4C87EFA6BA - Google Search - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com/search?q=1814E0A5AF37443703D7EE4C87EFA6BA&s
Web Images Groups News Froogle Local more »
1814E0A5AF37443703D7EE4C87EFA6BA
Search
Advanced Search
Preferences
Web

Your search - **1814E0A5AF37443703D7EE4C87EFA6BA** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.

… then the Russian site:



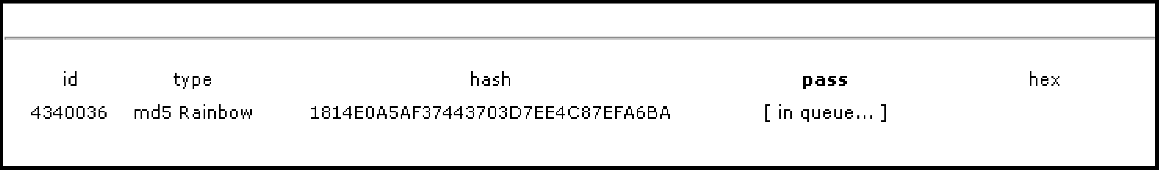
1814E0A5AF37443703D7EE4C87EFA6BA not found in our open database.

You can try to search for another Password Recovery solution.

If you could find better - please send us an url ;)

If you think that this is md5 **unsalted** hash - PassCracking.RU can try to do something...
Notice that the priority for this hash in crack engine will be LOWEST...
because you are: unregistered freeware user.

… of course I submitted it and it has been put into the queue – it is very possible that if you check this the time you have read this there will be a solution in the database …



id	type	hash	pass	hex
4340036	md5 Rainbow	1814E0A5AF37443703D7EE4C87EFA6BA	[in queue...]	

In the research for hacking message digests I found that a lot of sites out there use MD5 for many things (some things that it shouldn't). The next time you see a hash type it in Google and see if you surprised ;)

This concludes Lesson 4 of ***Hacking Cryptograms 101***. If you encounter a cryptogram or what you suspect to be a secret message that interests you either because you are stuck solving it or for any other reason please send me an email. See you next edition!

Keeping it 'rael,
Israel Torres

Cryptogram

The test of a first-rate intelligence is the ability to hold two opposed ideas in mind at the same time and still retain the ability to function. - F. Scott Fitzgerald

```
04 00 00 00 00 00
16 18 21 22 24 30
01 10 17 21 24 40
60 64 89 00 00 00
10 17 33 00 00 00
```

Email your solution to israel@israeltorres.org

Source Code

Source code bundle can be downloaded here:
http://blacklisted411.israeltorres.org/Blacklisted411.net_HackingCryptograms101_example_4_all.zip

Individual Source code can be downloaded here:
http://blacklisted411.israeltorres.org/md5demo.txt
http://blacklisted411.israeltorres.org/md5crack.bat
http://blacklisted411.israeltorres.org/md5check.bat
http://blacklisted411.israeltorres.org/dictionary1.txt

MD5 hash of the source code bundle can be validated at:
http://blacklisted411.israeltorres.org/Blacklisted411.net_HackingCryptograms101_example_4_all.md5.txt

note: you may have to copy and paste the addresses above if they wrap around!

References and Related Links

Internet Anagram Server
http://www.wordsmith.org/anagram/

NATO Phonetic Alphabet
http://en.wikipedia.org/wiki/NATO_phonetic_alphabet

Brute Force Attack
http://en.wikipedia.org/wiki/Brute_force_attack

MD5

http://en.wikipedia.org/wiki/MD5

MD5 Online Cracking (*Project Closed*)
http://passcracking.com/

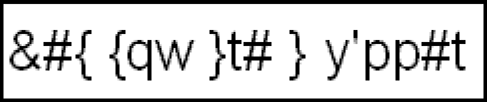
Project RainbowCrack
http://www.antsight.com/zsl/rainbowcrack/

MD Crack
http://c3rb3r.openwall.net/mdcrack/

MD5: Command Line Message Digest Utility
http://www.fourmilab.ch/md5/

[3] ==Cryptogram==

This should be rather simple for those that spend some time on it. Submissions need to be sent to crypto@blacklisted411.net by the 14th of March at the latest.



&#{ {qw }t# } y'pp#t

[4] ==Favorite Photo==

RCA MSN TV2 Linux Cluster
http://mirror.toc2rta.com/index.php/Main_Page

[5] ==Credits==

Chief Editor: Zachary Blackstone

Online Edition Editor: Ustler

Authors: Ustler, Israel Torres. Cynric

Blacklisted411

iHACKED.COM