GreHack 2013 CTF Write-up

Deva - Guillaume J.

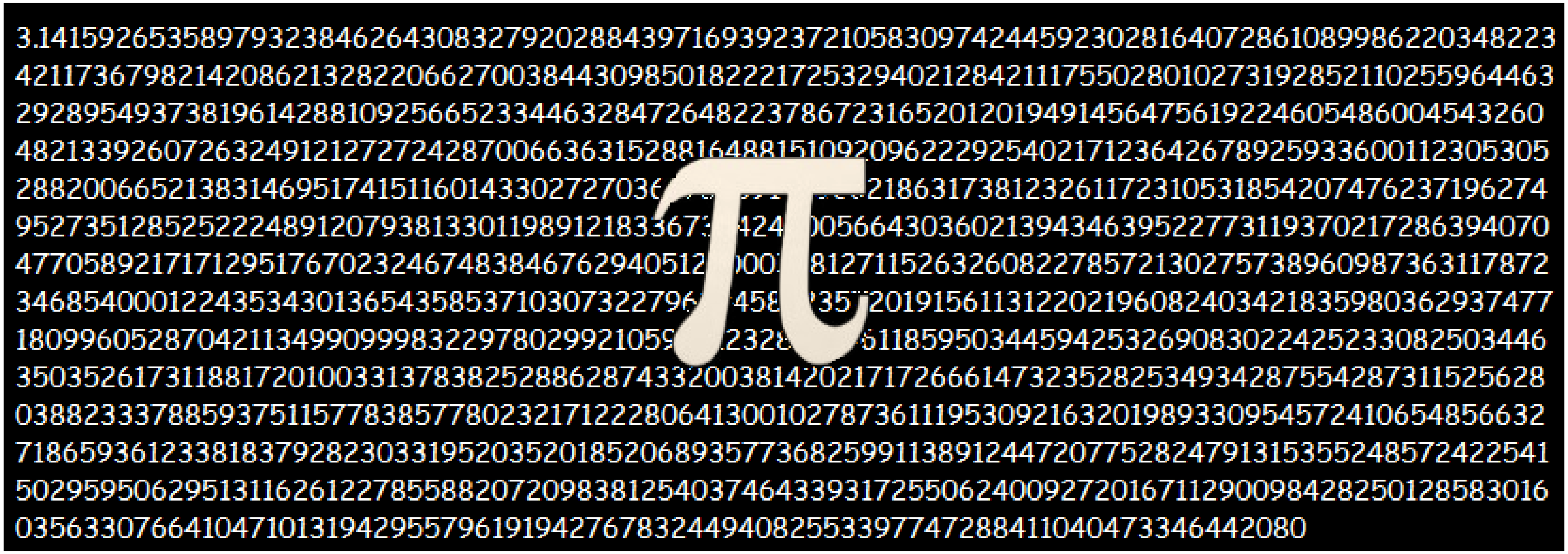
$prenom@grehack.org

Misc 200 – To PI or not to PI

Misc 200 – To PI or not to PI

Misc 200 – To PI or not to PI

- Hint 1 : This is not a mathematical operation, use your fucking brain!
- Hint 2 : This is not steganography!
- Hint 3 : Run it!

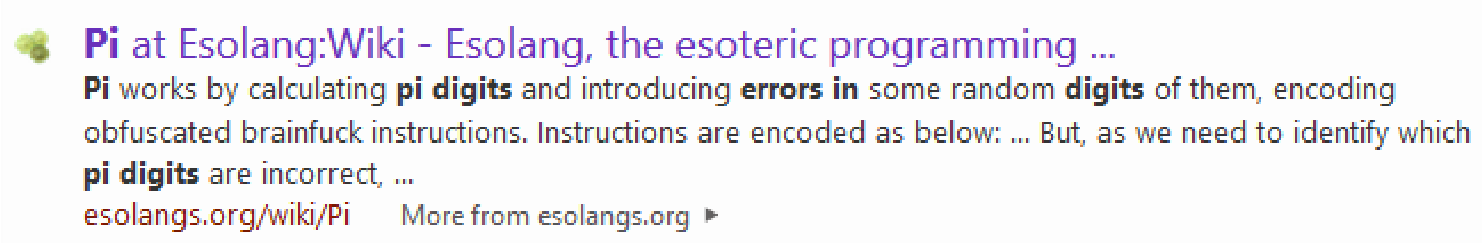


```
3.1415926535897932384626430832792028843971693923721058309742445923028164072861089986220348223
42117367982142086213282206627003844309850182222172532940212842111755028010273192852110255964463
292895493738196142881092566652334463284726482237867231652012019491456475619224605486004543260
482133926072632491212727242870066363152881648815109209622292540217123642678925933600112305305
2882006652138314695174151160143302727036[illegible]218631738123261172310531854207476237196274
952735128525222489120793813301198912183367[illegible]424[illegible]005664303602139434639522773119370217286394070
477058921717129517670232467483846762940512[illegible]00[illegible]812711526326082278572130275738960987363117872
34685400012243534301365435853710307322796[illegible]458[illegible]35[illegible]201915611312202196082403421835980362937477
1809960528704211349909998322978029921059[illegible]2328[illegible]61185950344594253269083022425233082503446
350352617311881720100331378382528862874332003814202171726661473235282534934287554287311525628
038823337885937511577838577802321712228064130010278736111953092163201989330954572410654856632
718659361233818379282303319520352018520689357736825991138912447207752824791315355248572422541
502959506295131162612278558820720983812540374643393172550624009272016711290098428250128583016
035633076641047101319429557961919427678324494082553397747288411040473346442080
```

Misc 200 – To PI or not to PI

- Hint 1 : This is not a mathematical operation, use your **fucking brain**!
  - http://esolangs.org/wiki/**Brainfuck**
    - « Pi obfuscates Brainfuck instructions in random errors in pi digits. “

- Or “random errors in pi digits” on google…



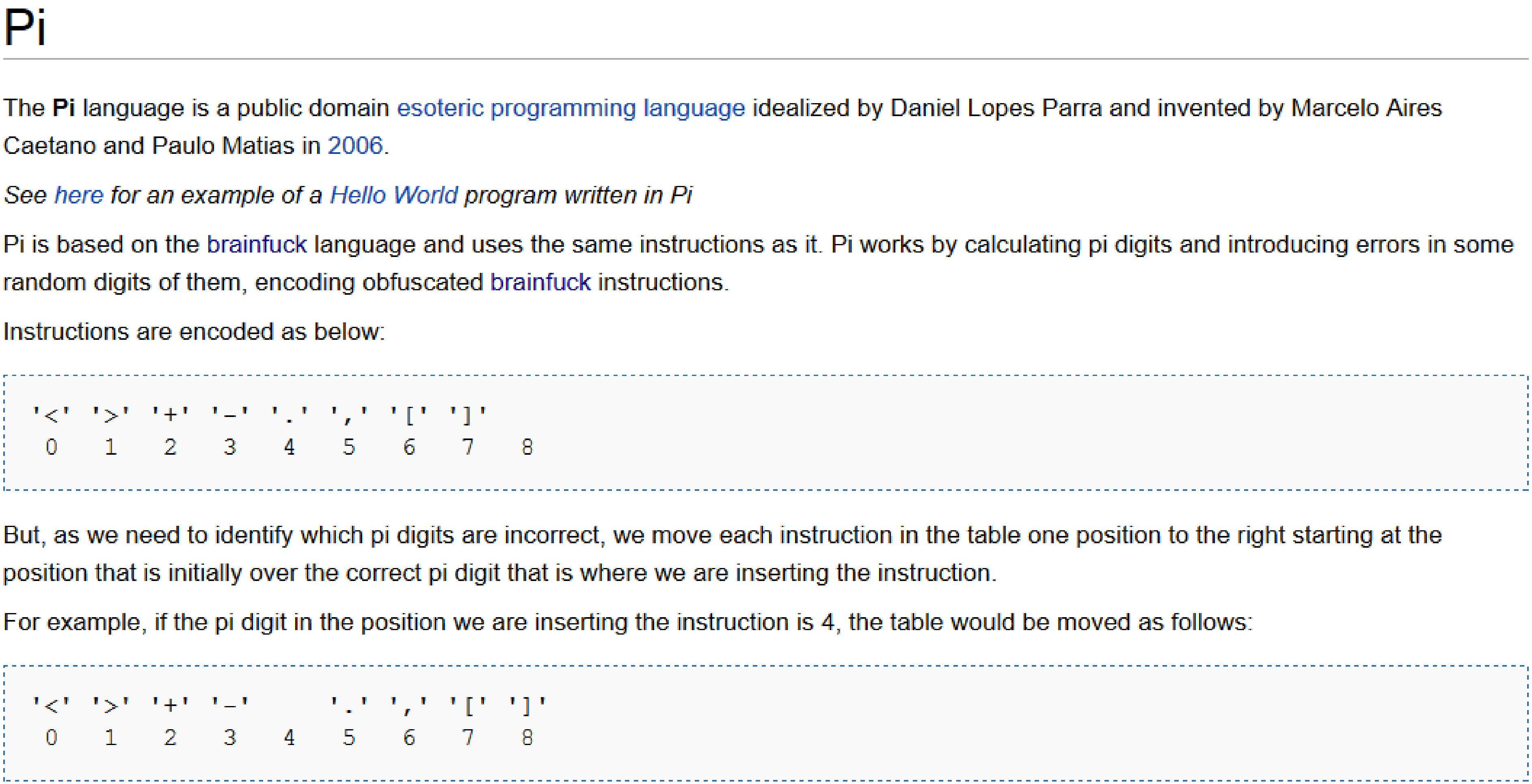
Pi at Esolang:Wiki - Esolang, the esoteric programming ...

Pi works by calculating **pi digits** and introducing **errors in** some random **digits** of them, encoding obfuscated brainfuck instructions. Instructions are encoded as below: ... But, as we need to identify which **pi digits** are incorrect, ...

esolangs.org/wiki/Pi More from esolangs.org ▸

- PI digits were at the end of the png file

Misc 200 – To PI or not to PI



Pi

The **Pi** language is a public domain esoteric programming language idealized by Daniel Lopes Parra and invented by Marcelo Aires Caetano and Paulo Matias in 2006.

See here for an example of a Hello World program written in Pi

Pi is based on the brainfuck language and uses the same instructions as it. Pi works by calculating pi digits and introducing errors in some random digits of them, encoding obfuscated brainfuck instructions.

Instructions are encoded as below:

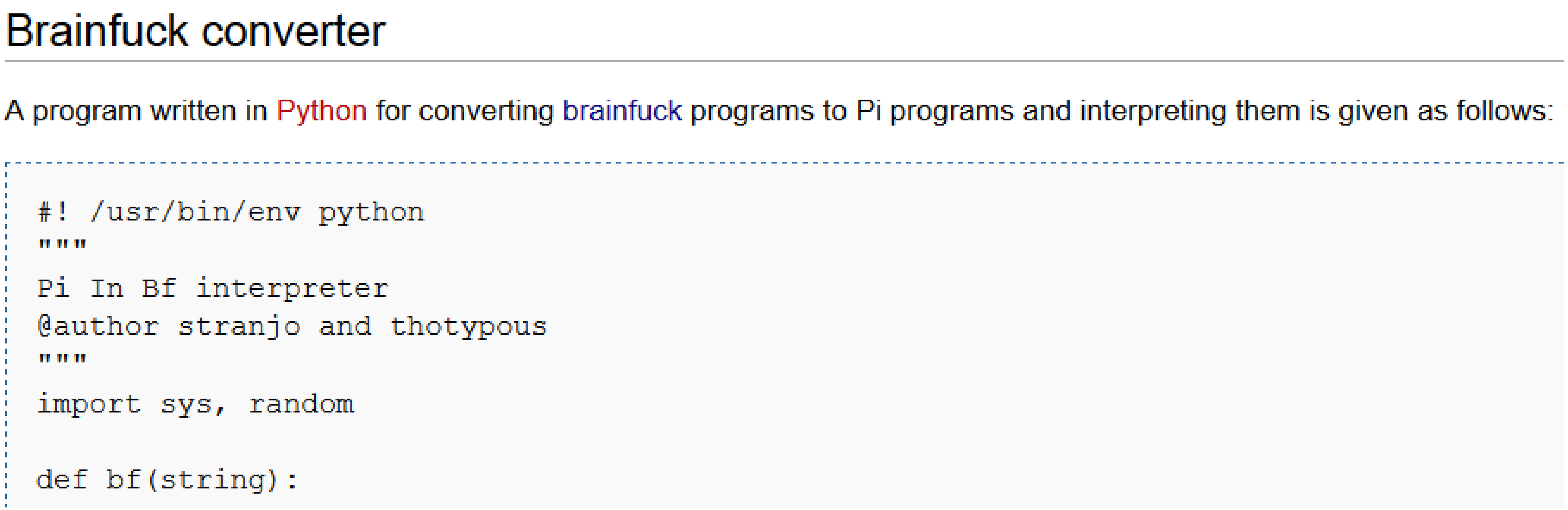
```
 '<'  '>'  '+'  '-'  '.'  ','  '['  ']'
  0    1    2    3    4    5    6    7    8
```

But, as we need to identify which pi digits are incorrect, we move each instruction in the table one position to the right starting at the position that is initially over the correct pi digit that is where we are inserting the instruction.

For example, if the pi digit in the position we are inserting the instruction is 4, the table would be moved as follows:

```
 '<'  '>'  '+'  '-'       '.'  ','  '['  ']'
  0    1    2    3    4    5    6    7    8
```

Misc 200 – To PI or not to PI



Brainfuck converter

A program written in Python for converting brainfuck programs to Pi programs and interpreting them is given as follows:

```
#! /usr/bin/env python
"""
Pi In Bf interpreter
@author stranjo and thotypous
"""
import sys, random

def bf(string):
```

- Result in Brainfuck:

>+++++++[<+++++++>-]<++++.>+++++[<++++++>-]<+.>>+++++++[<+++++++>-
]<++.>+++++[<++++>-]<.>--[<+++>-]<.>++++[<+++>-]<+.>------[<+++++>-]<.>++++[<+++++>-
]<+++.>++[<+++++>-]<+++.>------[<+++++>-]<.>+++++[<+++++>-]<+++.>---[<++>-]<-.

- Run it and get the flag

Misc 300 – It will be fun !

Misc 300 – It will be fun !

Misc 300 – It will be fun !

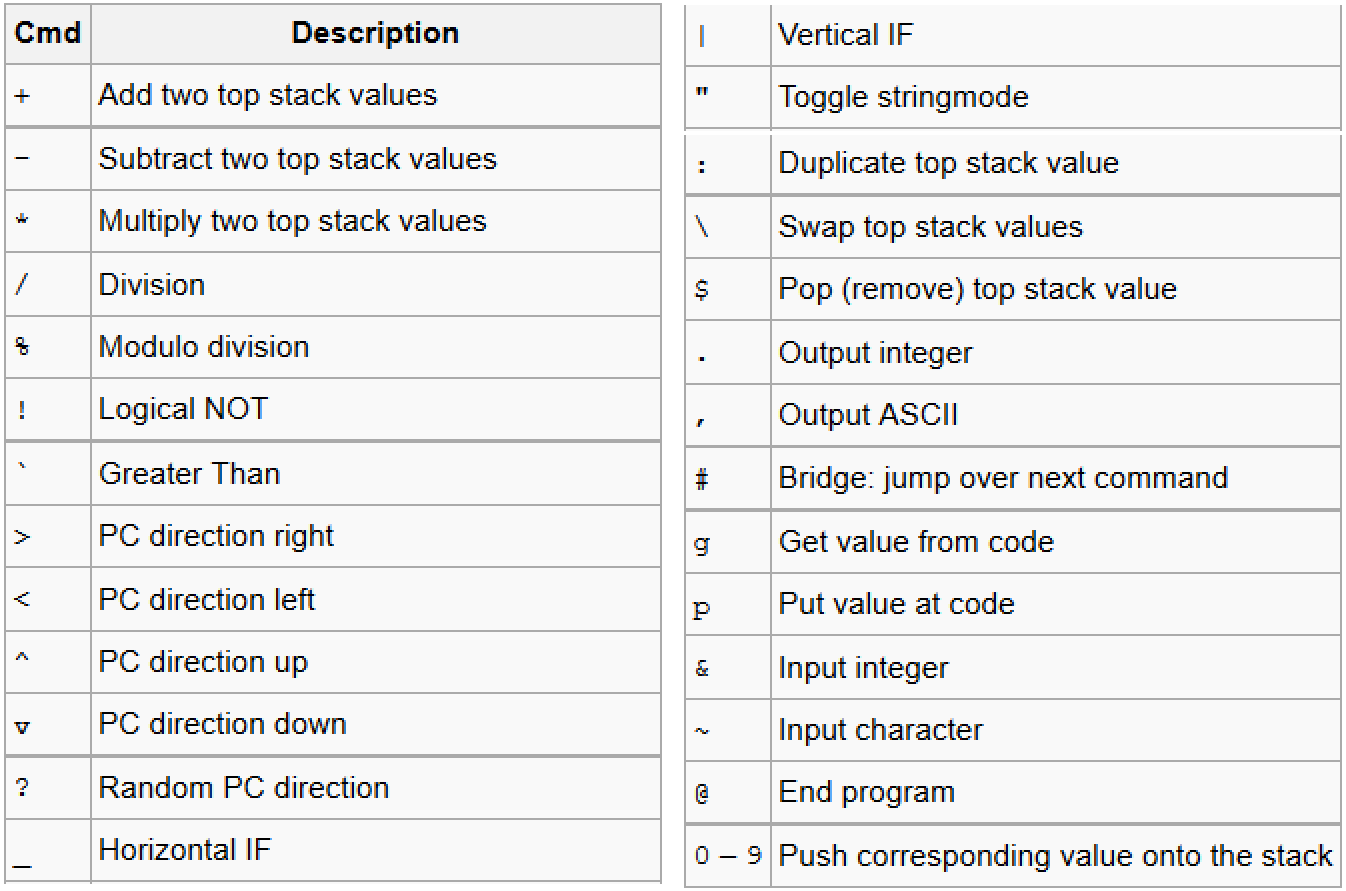
```
" GREHACK"#   4^
^vv"HACK iZ"<<
| 4#-7/+19<#p
>#- # 22+1p#+^
>#,88$$~\ :#0  |
  7|-3%+19::*#$<
^<->91+/91+#3-|
vp,#1+43 \0 6#<
_#2::91+%2-|7  ^
|-*#5/+@#19<"
```

```
>#2 0_#, 51#^+v
v1-#9::,# p1"#<
>#:+%8#"-|##,
v1:#9$1g#<   ,
>#,+/9#e1+-|,
v 7#1+3"#2 <,
>#8p::91+%| ,
10+DEVA,YO# ^"wit"<
01>+,8+,$3*1-:,"h"^
<v $$$$"Deva"-1<v
|   -7/+19<<
>#11$0$1$:v
>  >"non", ,,@> >
```

- 2 teams solved it
- Many thought it was brainfuck

Misc 300 – It will be fun !

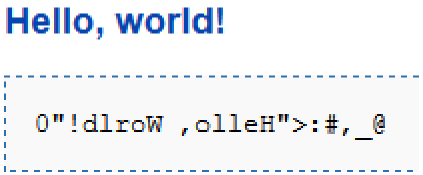
- Befunge, https://fr.wikipedia.org/wiki/Befunge



Cmd	Description
+	Add two top stack values
-	Subtract two top stack values
*	Multiply two top stack values
/	Division
%	Modulo division
!	Logical NOT
`	Greater Than
>	PC direction right
<	PC direction left
^	PC direction up
v	PC direction down
?	Random PC direction
_	Horizontal IF
|	Vertical IF
"	Toggle stringmode
:	Duplicate top stack value
\	Swap top stack values
$	Pop (remove) top stack value
.	Output integer
,	Output ASCII
#	Bridge: jump over next command
g	Get value from code
p	Put value at code
&	Input integer
~	Input character
@	End program
0 – 9	Push corresponding value onto the stack

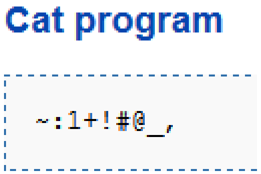
Misc 300 – It will be fun !

- Some examples from http://esolangs.org/wiki/Befunge
- Can be extended in multi-dimensional space (dimension > 2)



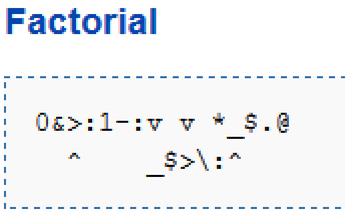
Hello, world!

```
0"!dlroW ,olleH">:#,_@
```



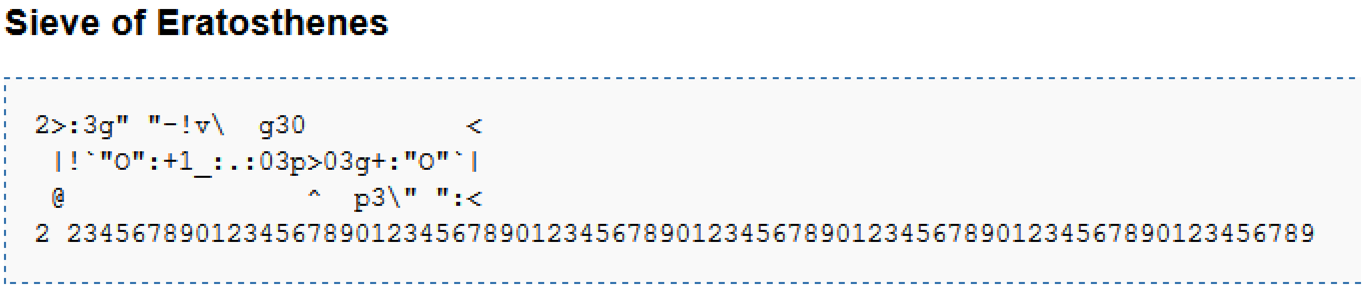
Cat program

```
~:1+!#@_,
```



Factorial

```
0&>:1-:v v *_$.@
  ^    _$>\:^
```



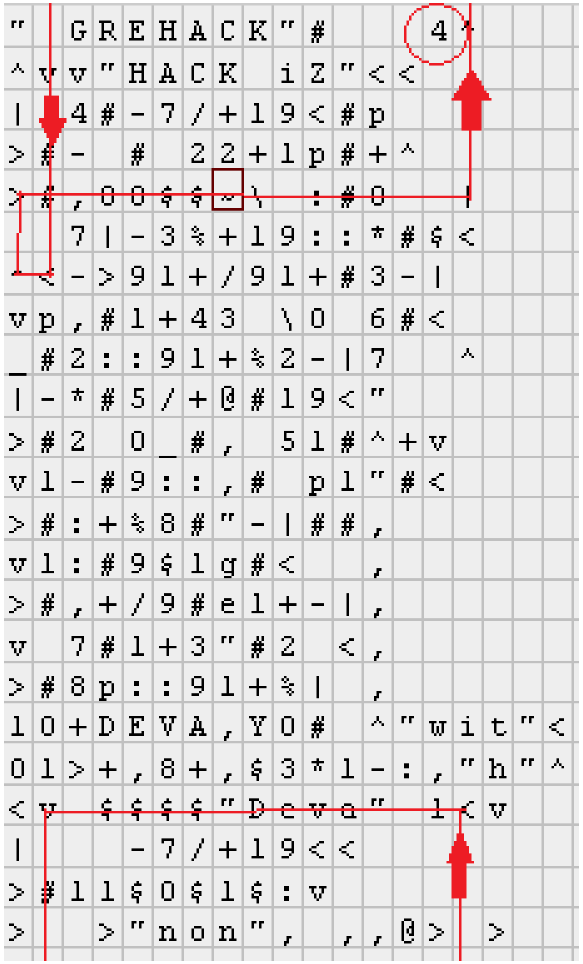
Sieve of Eratosthenes

```
2>:3g" "-!v\  g30          <
 |!`"O":+1_:.:03p>03g+:"O"`|
 @               ^  p3\" ":<
2 234567890123456789012345678901234567890123456789012345678901234567890123456789
```

Misc 300 – It will be fun !

- Use a befunge interpreter, eg Wasabi, jsFunge or online!
- Flag is computed from a 4 chars input, with a self-modifying key
- If input key is wrong : “non” in answer
  - Bruteforce
  - Reverse
  - Guess?
- Input key : Fl4g

Misc 300 – It will be fun !



```
" GREHACK"#   4^
^vv"HACK iZ"<<
| 4#-7/+19<#p
>#- # 22+1p#+^
>#,88$$~\ :#0  |
  7|-3%+19::*#$<
^<->91+/91+#3-|
vp,#1+43 \0 6#<
_#2::91+%2-|7  ^
|-*#5/+@#19<"
>#2 0_#, 51#^+v
v1-#9::,# p1"#<
>#:+%8#"-|##,
v1:#9$1g#<  ,
>#,+/9#e1+-|,
v 7#1+3"#2 <,
>#8p::91+%| ,
10+DEVA,Y0# ^"wit"<
01>+,8+,$3*1-:,"h"^
<v $$$$"Deva"-1<v
|   -7/+19<<
>#11$0$1$:v
>  >"non",  ,,@> >
```

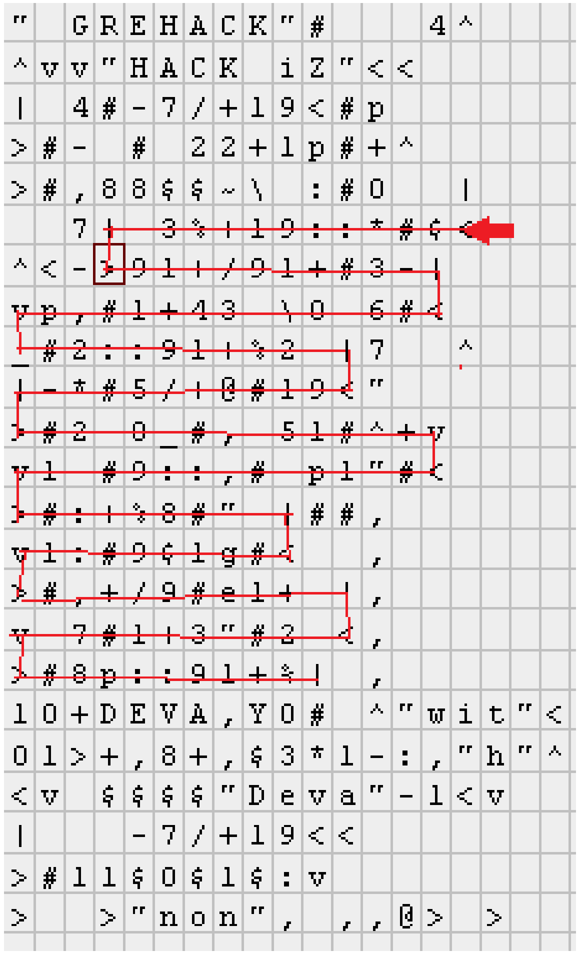
Input loop,

```
for(i=4, i>0, i--) {
        x = get(char);
        push x;
}
```

Verification routine

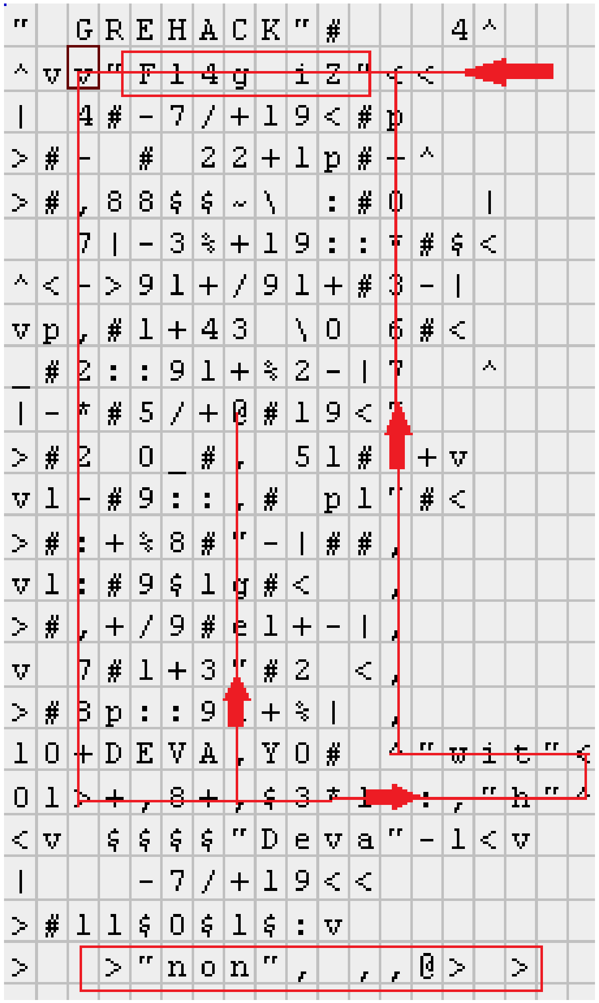
(first line)

```
Pop x;
If(x/10-7==0)
        continue;
else
        abort;
```



```
" GREHACK"#   4^
^vv"HACK iZ"<<
| 4#-7/+19<#p
>#- # 22+1p#+^
>#,88$$~\ :#0  |
  7|-3%+19::*#$<
^<->91+/91+#3-|
vp,#1+43 \0 6#<
_#2::91+%2-|7  ^
|-*#5/+@#19<"
>#2 0_#, 51#^+v
v1-#9::,# p1"#<
>#:+%8#"-|##,
v1:#9$1g#<  ,
>#,+/9#e1+-|,
v 7#1+3"#2 <,
>#8p::91+%| ,
10+DEVA,Y0# ^"wit"<
01>+,8+,$3*1-:,"h"^
<v $$$$"Deva"-1<v
|   -7/+19<<
>#11$0$1$:v
>  >"non",  ,,@> >
```

Misc 300 – It will be fun !



```
" GREHACK"#   4^
^vv"Fl4g iZ"<<
| 4#-7/+19<#p
>#- # 22+1p#-^
>#,88$$~\ :#0  |
  7|-3%+19::*#$<
^<->91+/91+#3-|
vp,#1+43 \0 6#<
_#2::91+%2-|7  ^
|-*#5/+@#19<7
>#2 0_#, 51# +v
v1-#9::,# p1"#<
>#:+%8#"-|##,
v1:#9$1g#<  ,
>#,+/9#e1+-|,
v 7#1+3"#2 <,
>#3p::91+%| ,
10+DEVA,Y0# ^"wit"<
01>+,8+,$3*1 :,"h"^
<v $$$$"Deva"-1<v
|   -7/+19<<
>#11$0$1$:v
>  >"non", ,,@> >
```

Abort function

Flag computation

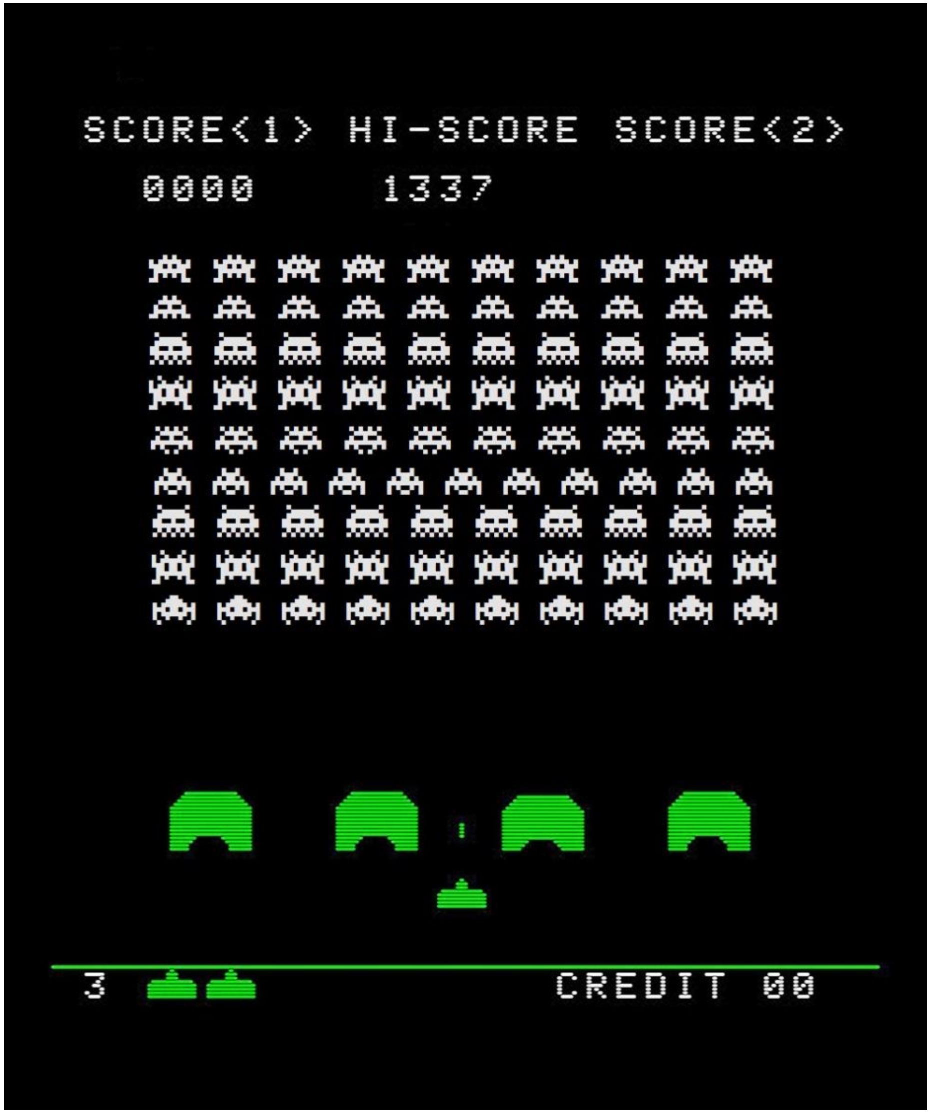
- Flag is « Befun_wih_Befunge »
- Same path is used to compute both « Befun »
- « HACK iZ » -> « Fl4g iZ » (self-modifying code)

Stega 100 - Invaders

Stega 100 - Invaders

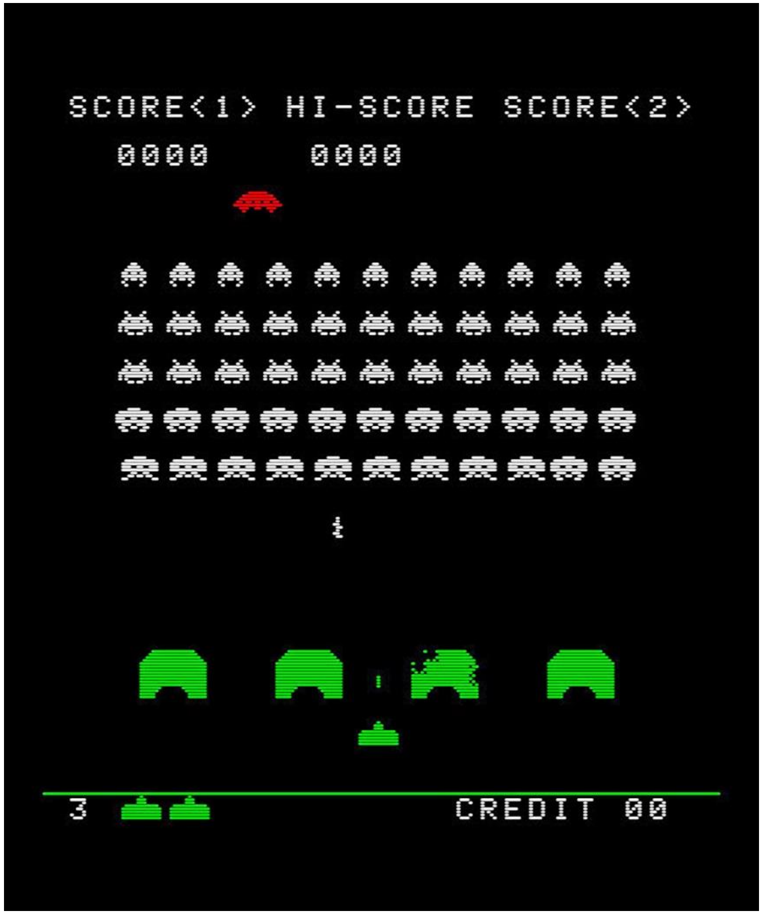
Stega 100 - Invaders

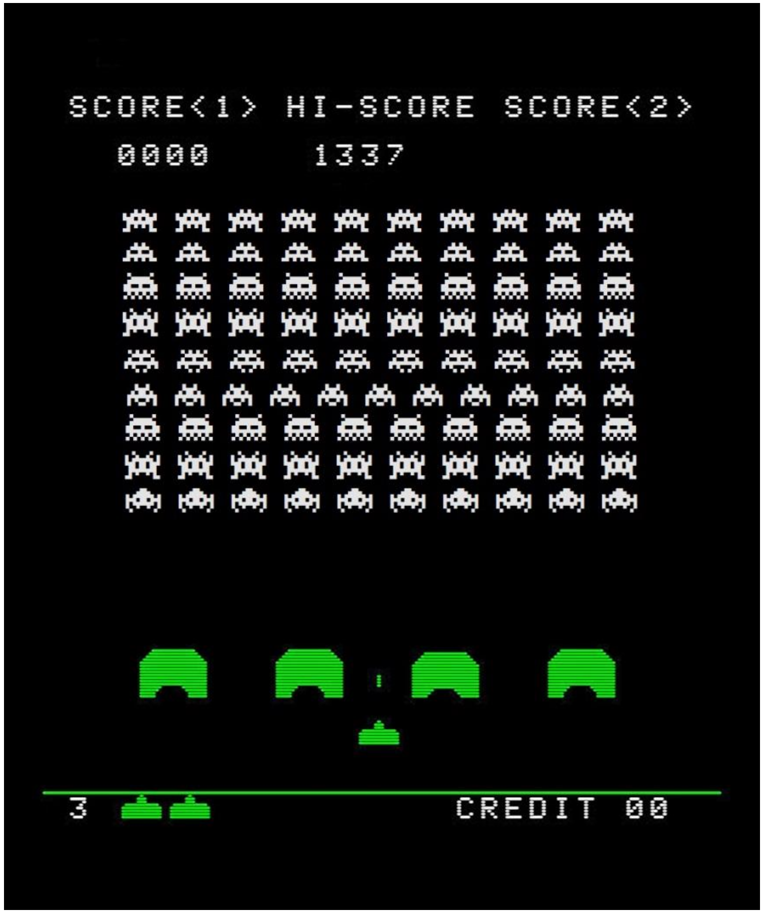
- unsolved



Stega 100 - Invaders

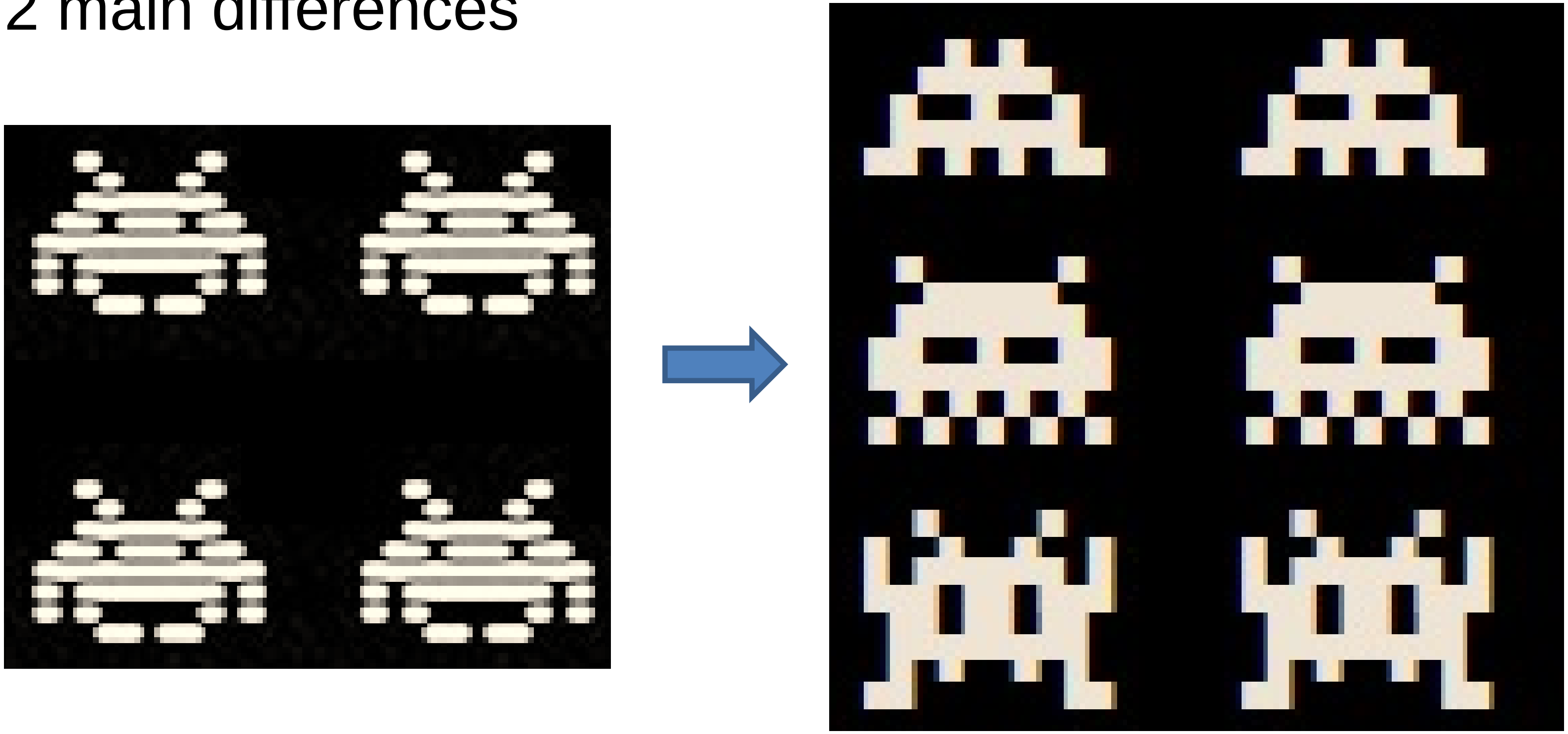
- « Invaders » on google :

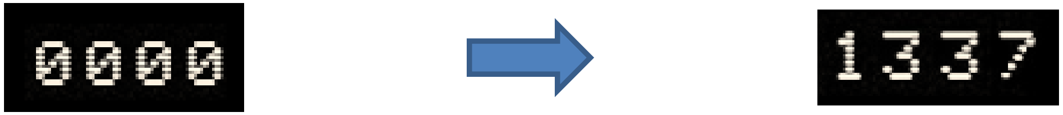




Stega 100 - Invaders

- 2 main differences



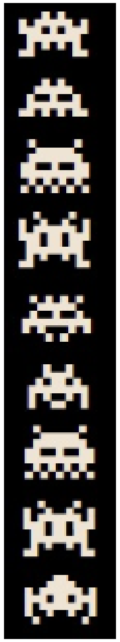


Stega 100 - Invaders

- Invaders…leet...

- => http://www.dafont.com/fr/invaders.font

  (ok it was not obvious)

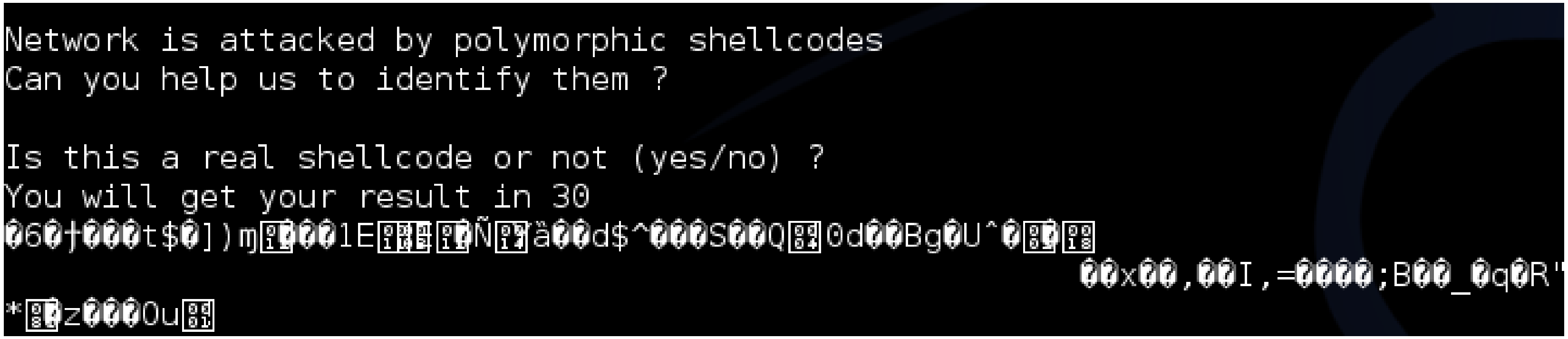
- Translate  with the invaders font

Network 300 - NIDS

Network 300 - NIDS

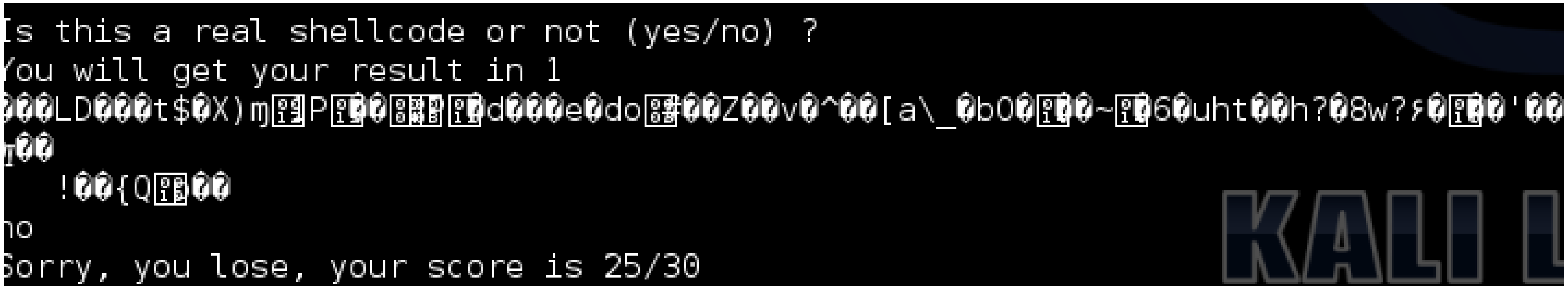
Network 300 - NIDS

- 2 teams solved it
- At least 4 different ways to solve it!



```
Network is attacked by polymorphic shellcodes
Can you help us to identify them ?

Is this a real shellcode or not (yes/no) ?
You will get your result in 30
�6�ϯ���t$�])ɱ���1E��Ñ�ä��d$^���S��Q�0d��Bg�U^���
                                                  ��x��,��I,=����;B��_�q�R"
*��z���0u�
```



```
Is this a real shellcode or not (yes/no) ?
You will get your result in 1
���LD���t$�X)ɱ�P������d���e�do���Z��v�^��[a\_�b0���~��6�uht��h?�8w?ϟ���'��
ȋ��
  !��{Q���
no
Sorry, you lose, your score is 25/30
```

Network 300 - NIDS

½ : random shellcode

½ : shikata_ga_nai from linux/reverse_tcp in metasploit

- The goal is to find if it is a polymorphic version of reverse_tcp or just random data.

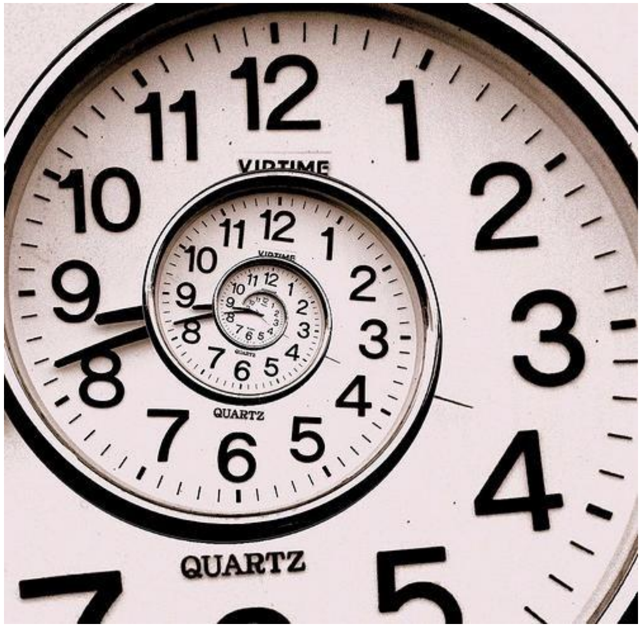
- Score /30 is only displayed at the end
  - => Impossible to know which one is a shellcode

Network 300 - NIDS

Manual way

Long and boring

- No time limit
  - Reverse 30 potential shellcodes



Network 300 - NIDS

Theoretical way

some possible solutions…

- Run it in a sandbox, check if it acts like the original reverse_tcp
- Disass and do abstract interpretation
- …

Network 300 - NIDS

Systematic way

(1 team used it)

- Get shellcodes, hash them and store them in a database
  - 500 real shellcodes
  - 256^98 random shellcodes possible (length = 98)
  - => If the shellcode is already in the database, it is very likely a real one.

Network 300 - NIDS

Hacker way

(1 team used it)

- Exploit a bug (voluntary or not :D ?) in the verification of answers.
- Answer “yes/no” to all questions