**MALWARE ANALYSIS REPORT**

**INDEX:**

1. Objective
2. PDF Static Analysis
3. PDF Dynamic Analysis
4. Backdoor Static Analysis
5. Analysis of strings from malware
6. Backdoor dynamic analysis
7. HTTP traffic documentation

**OBJECTIVE:**

A PDF containing an attack is given to us which will deliver and execute another program onto our VM environment. This project requires us to implement all the malware analysis methods used in all the homework to document specific characteristics and behaviors of the malware.

**PDF STATIC ANALYSIS:**

To create yara rule that can be used to identify the object that contains the stream causing exploit, following should be done.

* I have used the peepdf tool present in remnux. This tool identifies the objects, streams present in the object.

**File**: gidijakj.pdf

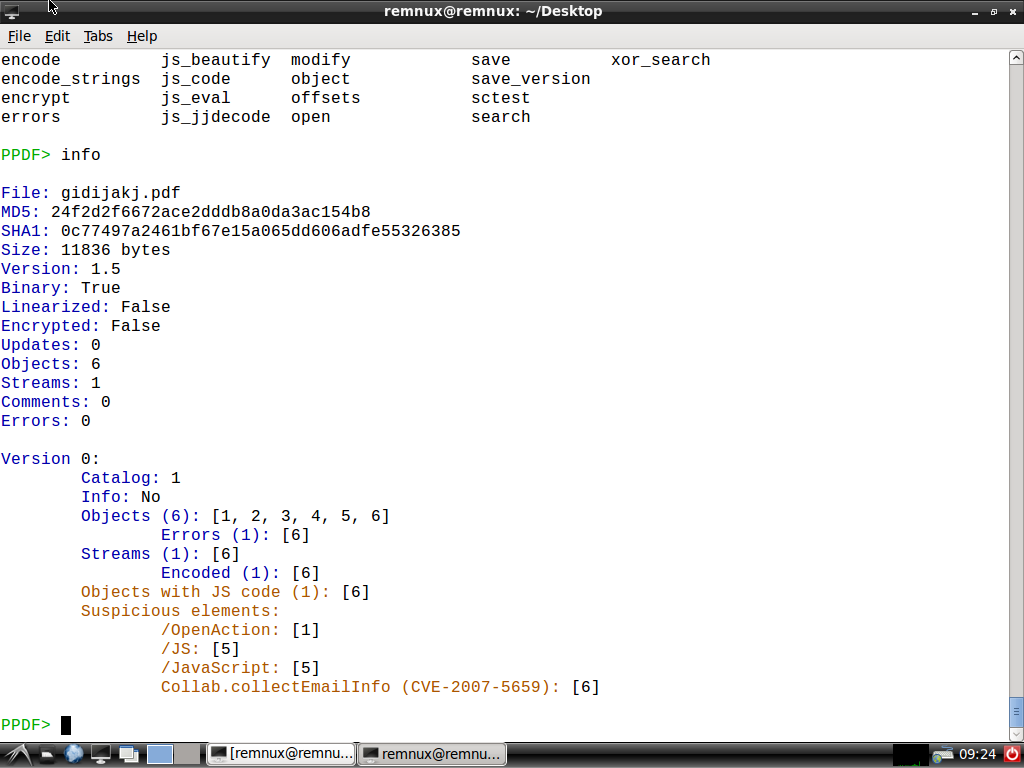
**MD5:**24f2d2f6672ace2dddb8a0da3ac154b8

**SHA-1:** 0c77497a2461bf67e15a065dd606adfe55326385

**Size:** 11836 bytes

**Version:** 1.5

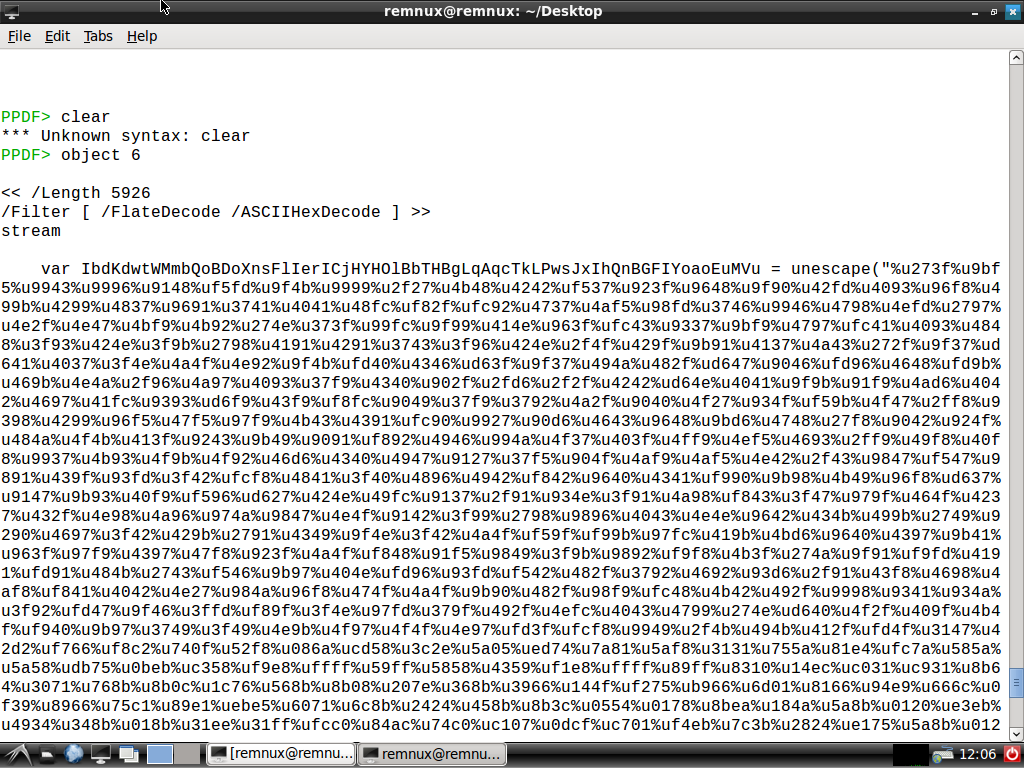
**Binary:** True

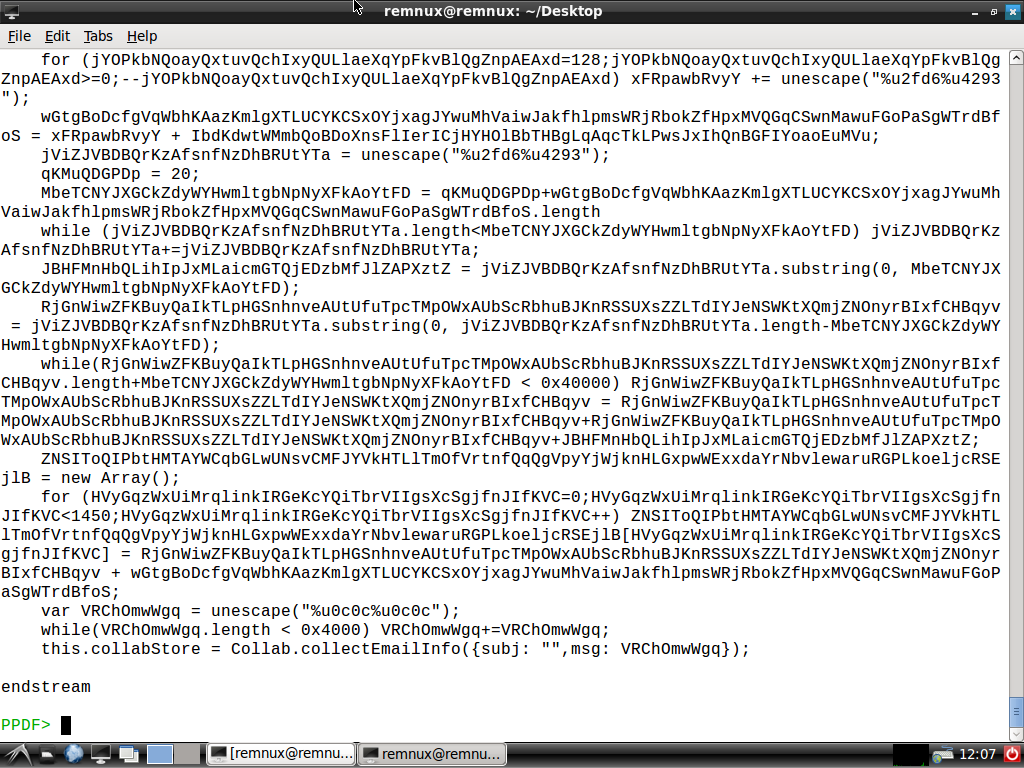


* I found out that object 6 contains the stream. To write a yara rule, the strings inside this object have to be identified. So I ran the command:

**object 6**

Using this command, the following popped up:





* I wrote the following yara rule using the strings that I found above and upon giving the following command, I saw that the result was the object for which I wrote the yara rule.

**Command:** ***pdf-parser.py -y gidijakj-pdf.yar gidijakj.pdf***

**YARA RULE:**

strings:

$a = “var VRCh0mwWgq”

$b = “unescape”

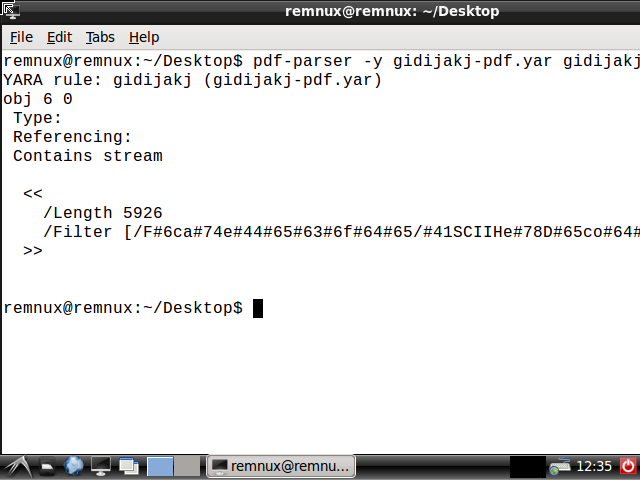
$c = “jlB = newArray() ; ”

$d = “var xFRpawbRvyY”

$e = “Collab.collectEmailInfo”

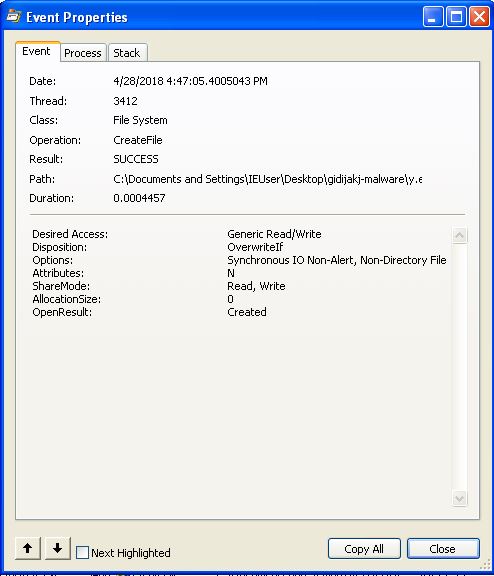
condition:

all of them

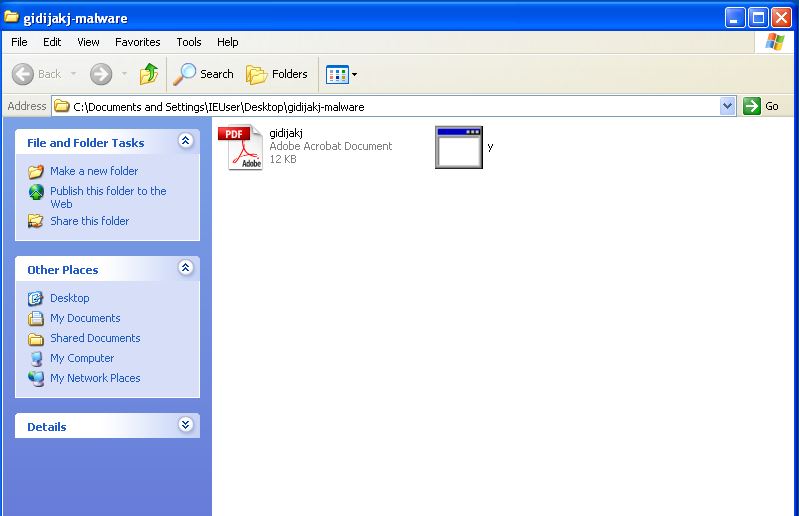
****

**PDF DYNAMIC ANALYSIS:**

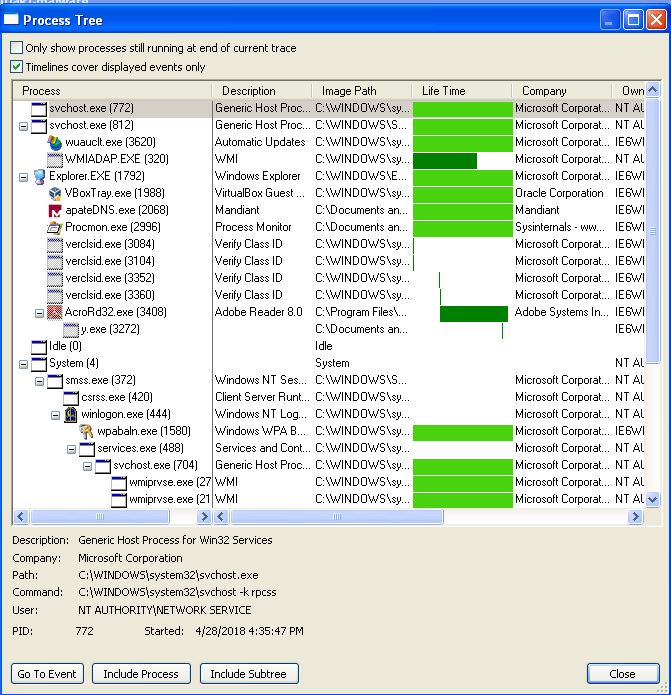
Once the pdf is opened a file named, y.exe is created.

****

y.exe is located in C:\Documents and Settings\IEUser\Desktop\gidijakj-malware

****

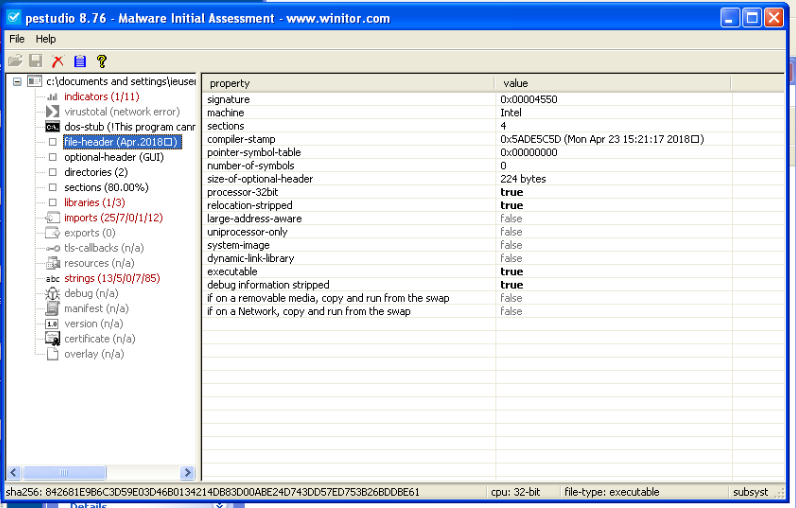
The next step I took is to run the file, y.exe.

****

**BACKDOOR STATIC ANALYSIS:**

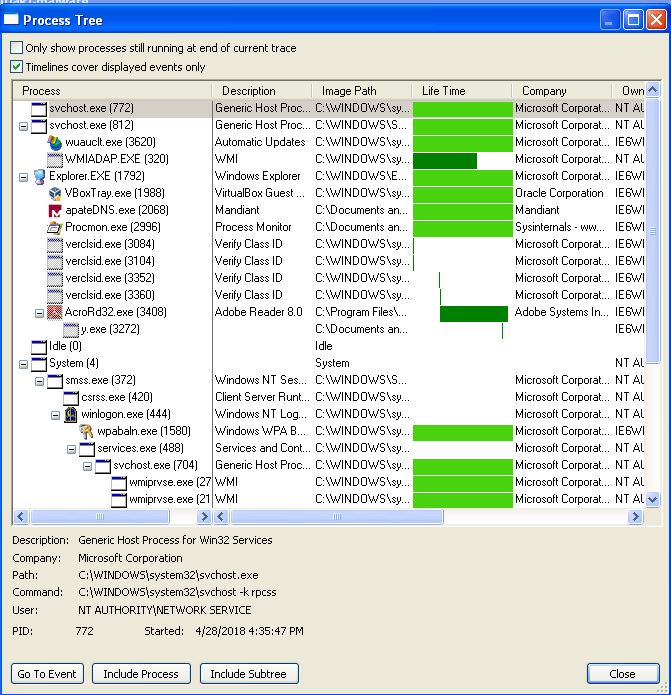
**EXE FILE NAME:** meaty.exe

**EXE COMPILE TIME:** Mon Apr 23 15:21:17 2018

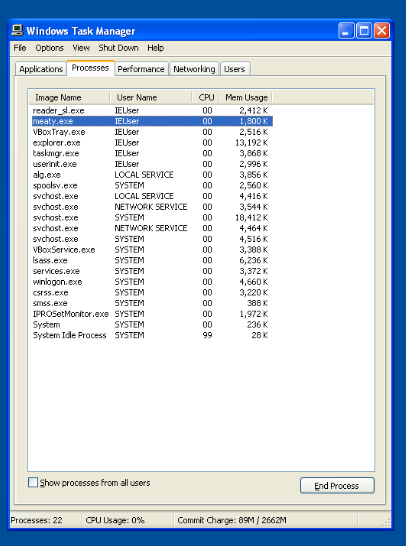


**EXE TYPE:** 32-bit windows executable

**DLL IMPORTS:**



**EXE achieves persistence by:**

****

**ANALYSIS OF STRINGS FROM MALWARE:**

**YARA RULE:**

rule meaty

{ meta:

author="gidijakj"

strings:

$g = "This program cannot be run in DOS mode"

$e = {4D 5A}

$f = {50 45}

$j = { 4B 45 52 4E 45 4C 33 32 2E 64 6C 6C }

$h = {F? FD FF FF 65}

$k = {57 53 32 5F 33 32 2E 64 6C 6C}

$i = {41 44 56 41 50 49 33 32 2E 64 6C 6C}

$a = {43 72 65 51 74 65 46 69 6C 65 41}

$b = "CreateFileA"

$c = {63 6D 64 2E 65 78 }

condition:

any of them

}

I have used the following command to run the yara rule:

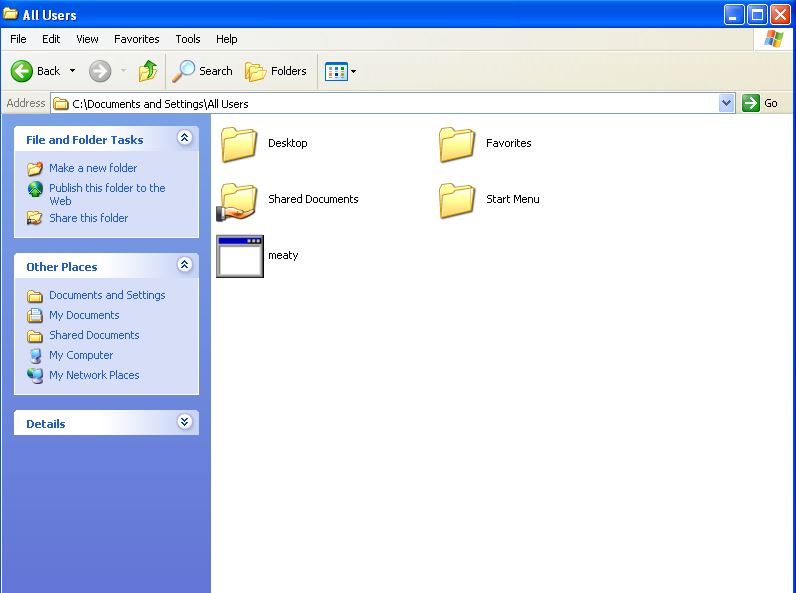
***sudo yara gidijakj-strings.yar gidijakj***

Using the command given below, I have generated a gidijakj-strings.out file which I have attached with the submission.

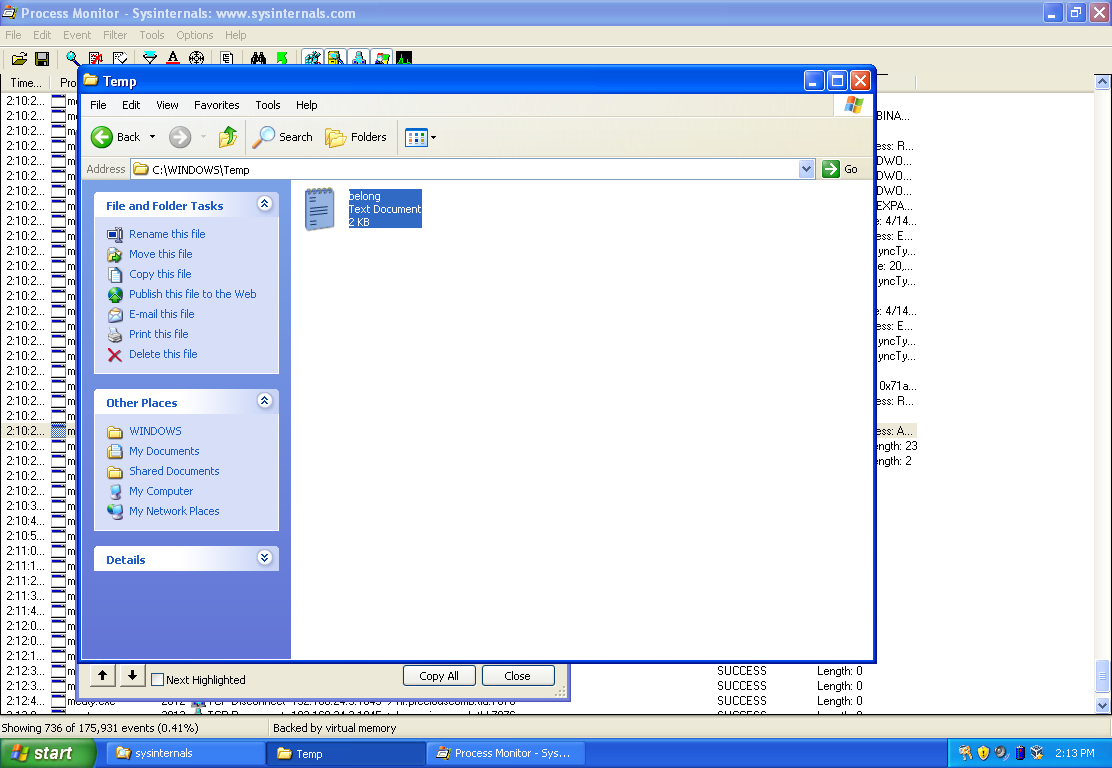
***sudo yara –s gidijakj-strings.yar gidijakj.pdf***

**BACKDOOR DYNAMIC ANALYSIS:**

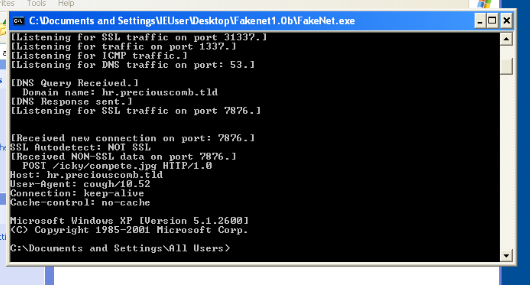
When I ran the file y.exe, a file named meaty.exe created in C:\Documents and Settings\All Users. meaty.exe is the backdoor.

****

When I opened the file, meaty.exe a file named, belong.exe was created.



When I ran the backdoor it continuously tries to connect to a domain named, hr.preciouscomb.tld through port 7876.

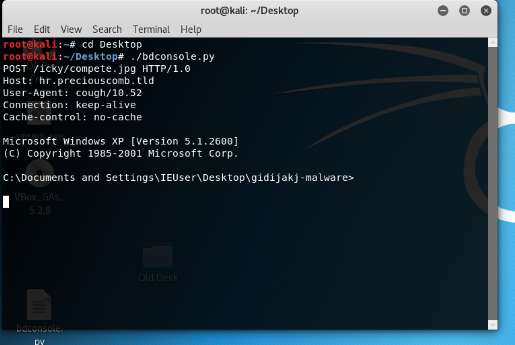
****

**HTTP TRAFFIC DOCUMENTATION:**

HTTP path requested: /icky/compete.jpg HTTP/1.0

HTTP command/verb being used: POST

User-Agent value being sent: cough/10.52

****