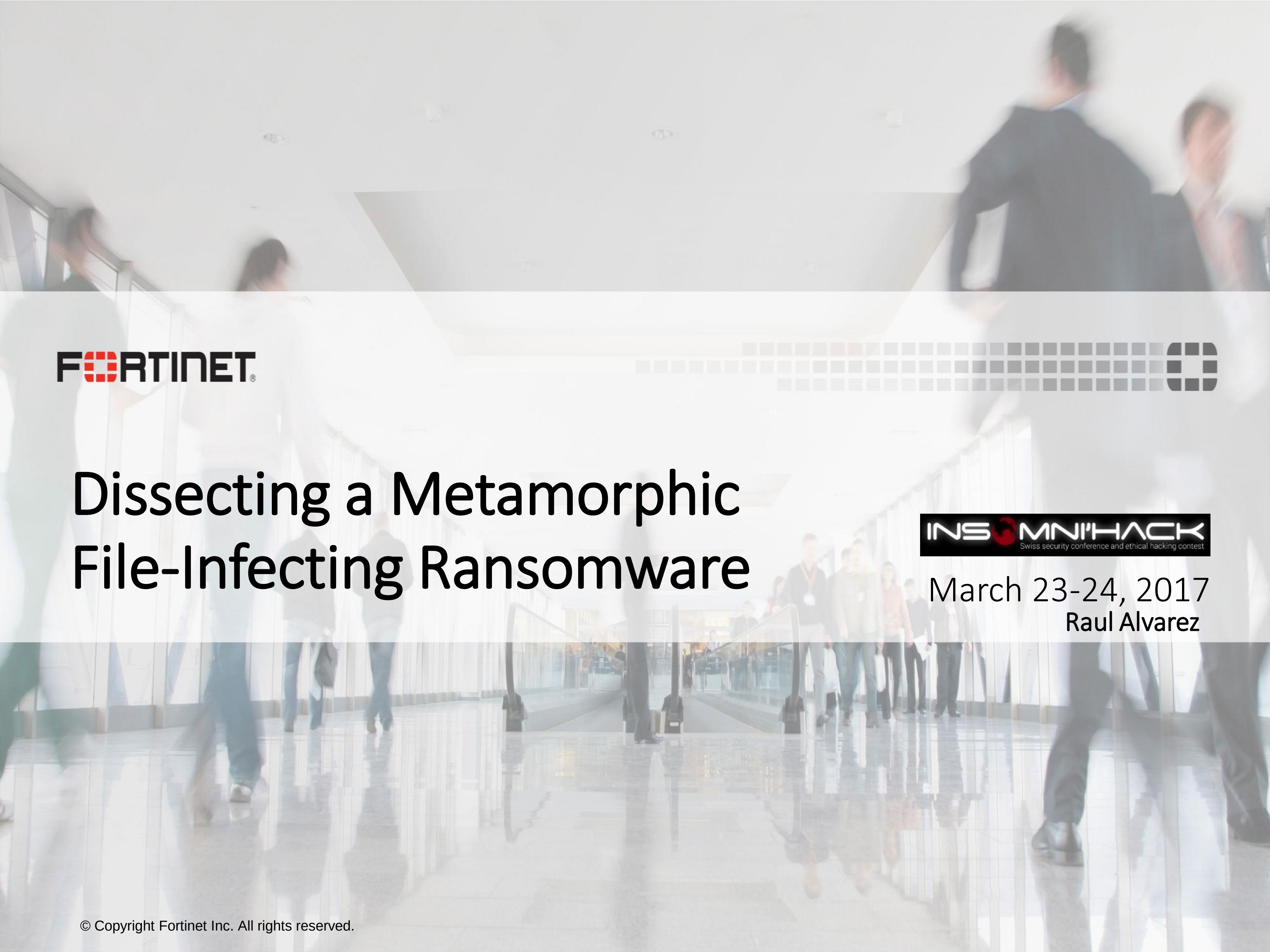


FORTINET

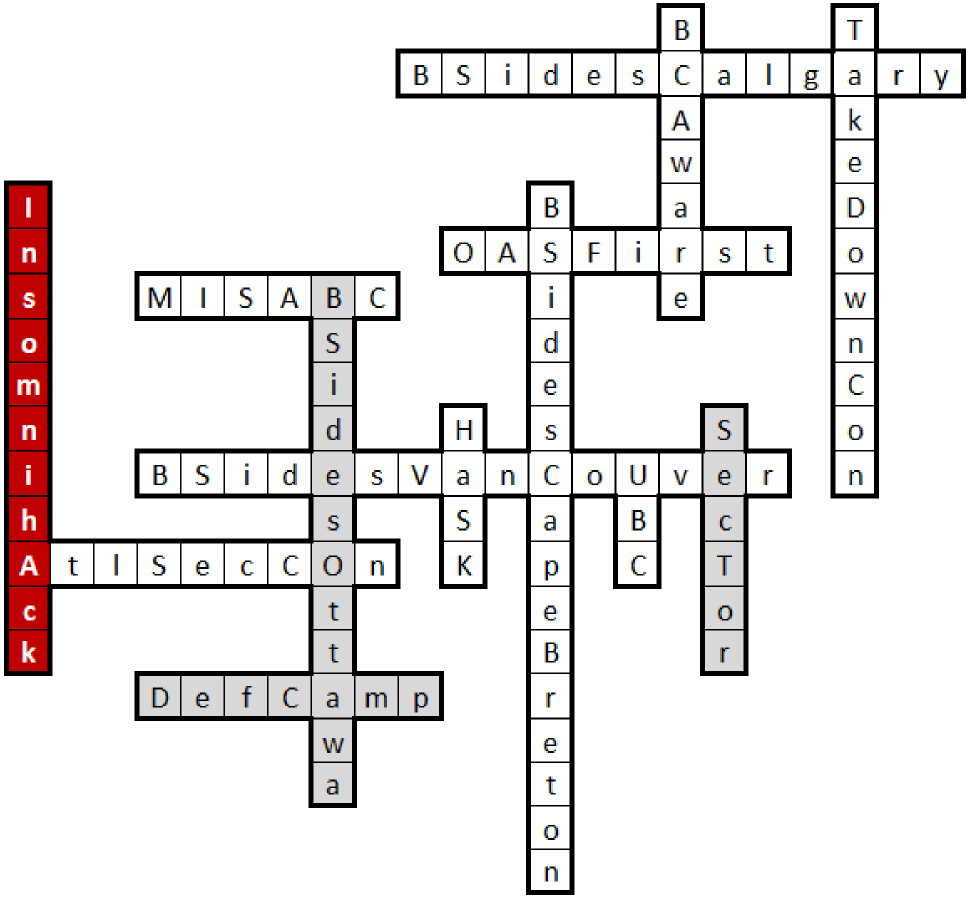
Dissecting a Metamorphic File-Infecting Ransomware

INSOMNI'HACK
Swiss security conference and ethical hacking contest

March 23-24, 2017
Raul Alvarez

About Me

- Senior Security Researcher @ Fortinet
- 22 published articles in Virus Bulletin
- Regular contributor in our company blog



BSidesCalgary

BCAware

TakeDownCon

OASFirst

BSidesCapeBreton

MISABC

BSidesOttawa

InsomniHack

HaSK

SecTor

BSidesVancouver

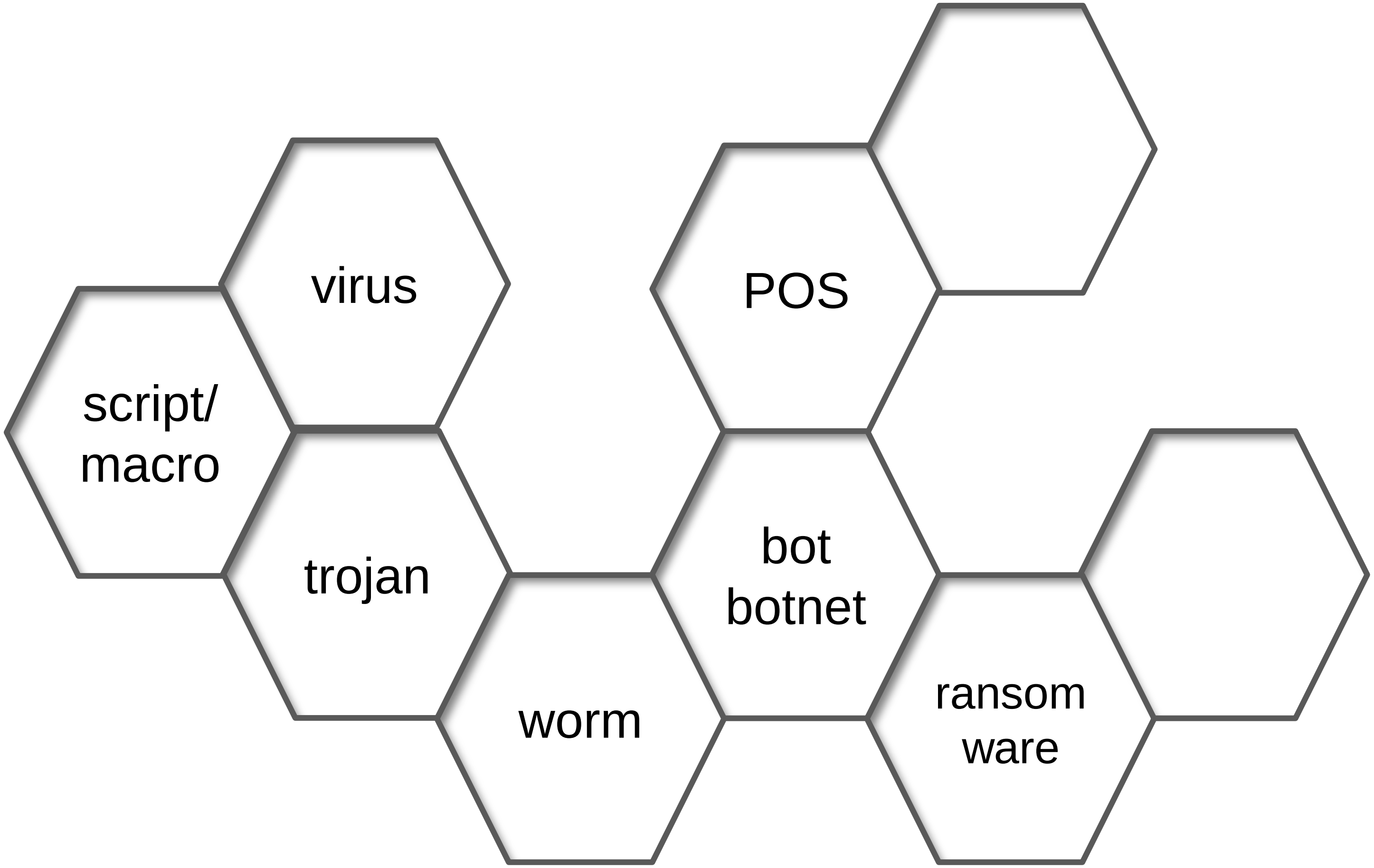
UBC

AtlSecCon

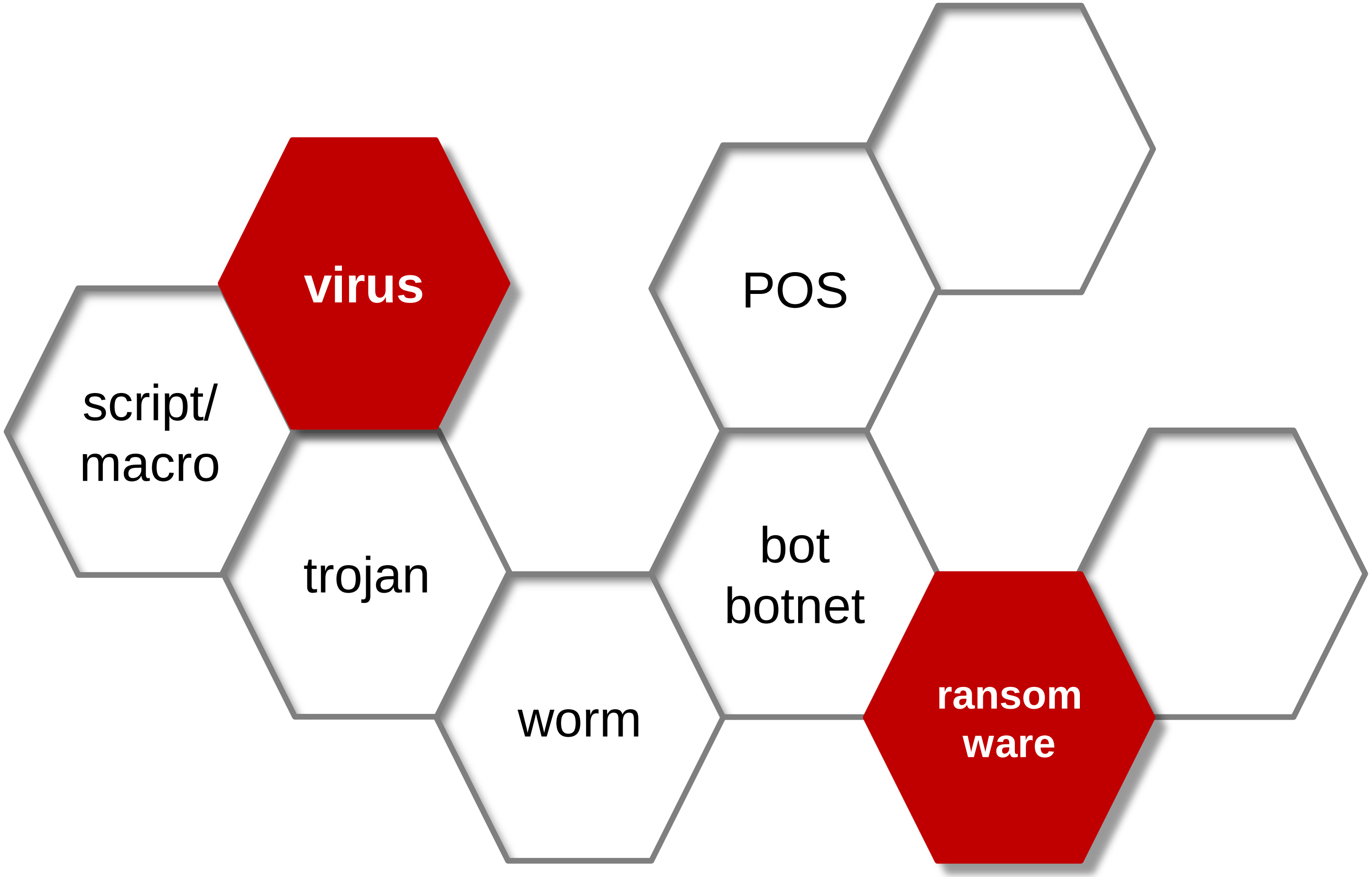
DefCamp

Malware Categories

Malware Honeycomb



Virlock



virus

script/ macro

trojan

worm

POS

bot botnet

ransom ware

Agenda

Agenda

Virlock as a ransomware

- Visible signs

Virlock as a common malware

- Reversing stages

Virlock as a file infector

- Extracting the host file

Virlock as a polymorphic malware

- On-demand polymorphic algorithm

Virlock as a metamorphic malware

- Metamorphic engine
- Generated metamorphic algorithm

Virlock

What Is A File Infector?

- Attaches the malware code into the host file.
- Appending, prepending, and cavity type
- Maintains persistency within the computer system
- Infected file is hard to restore

What Is A Ransomware?

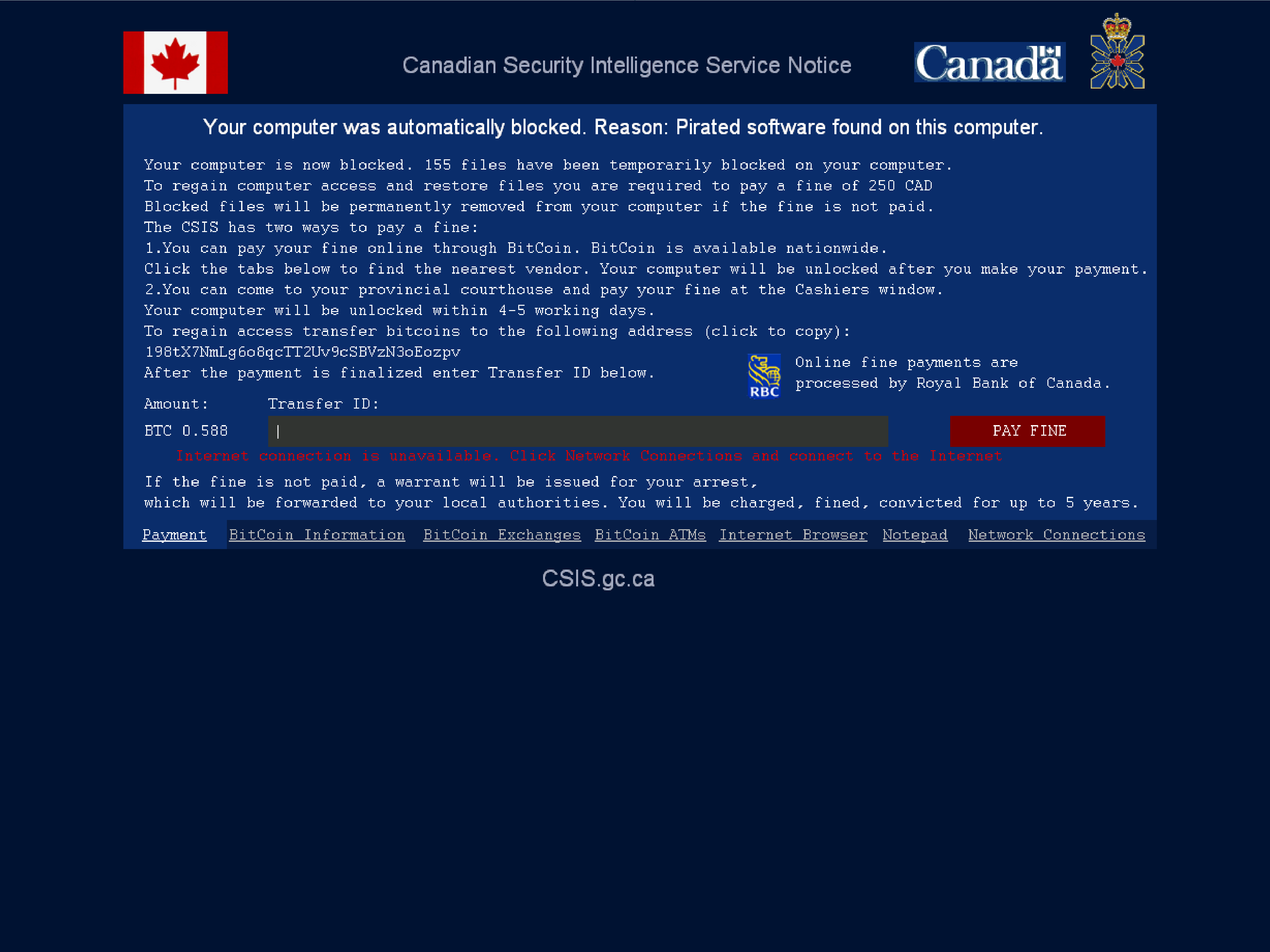
- Holds your computer for ransom
- Encrypts files
- Uses cryptocurrency, such as bitcoins, for payment

What Is Virlock?

- A ransomware
- A file infector
- Uses on-demand polymorphic algorithm
- Uses metamorphic algorithm
- Locks your screen

Virlock As A Ransomware

Visible Signs of Infection



Canadian Security Intelligence Service Notice

Your computer was automatically blocked. Reason: Pirated software found on this computer.

Your computer is now blocked. 155 files have been temporarily blocked on your computer.
To regain computer access and restore files you are required to pay a fine of 250 CAD
Blocked files will be permanently removed from your computer if the fine is not paid.
The CSIS has two ways to pay a fine:
1.You can pay your fine online through BitCoin. BitCoin is available nationwide.
Click the tabs below to find the nearest vendor. Your computer will be unlocked after you make your payment.
2.You can come to your provincial courthouse and pay your fine at the Cashiers window.
Your computer will be unlocked within 4-5 working days.
To regain access transfer bitcoins to the following address (click to copy):
198tX7NmLg6o8qcTT2Uv9cSBVzN3oEozpv
After the payment is finalized enter Transfer ID below.

Online fine payments are
processed by Royal Bank of Canada.

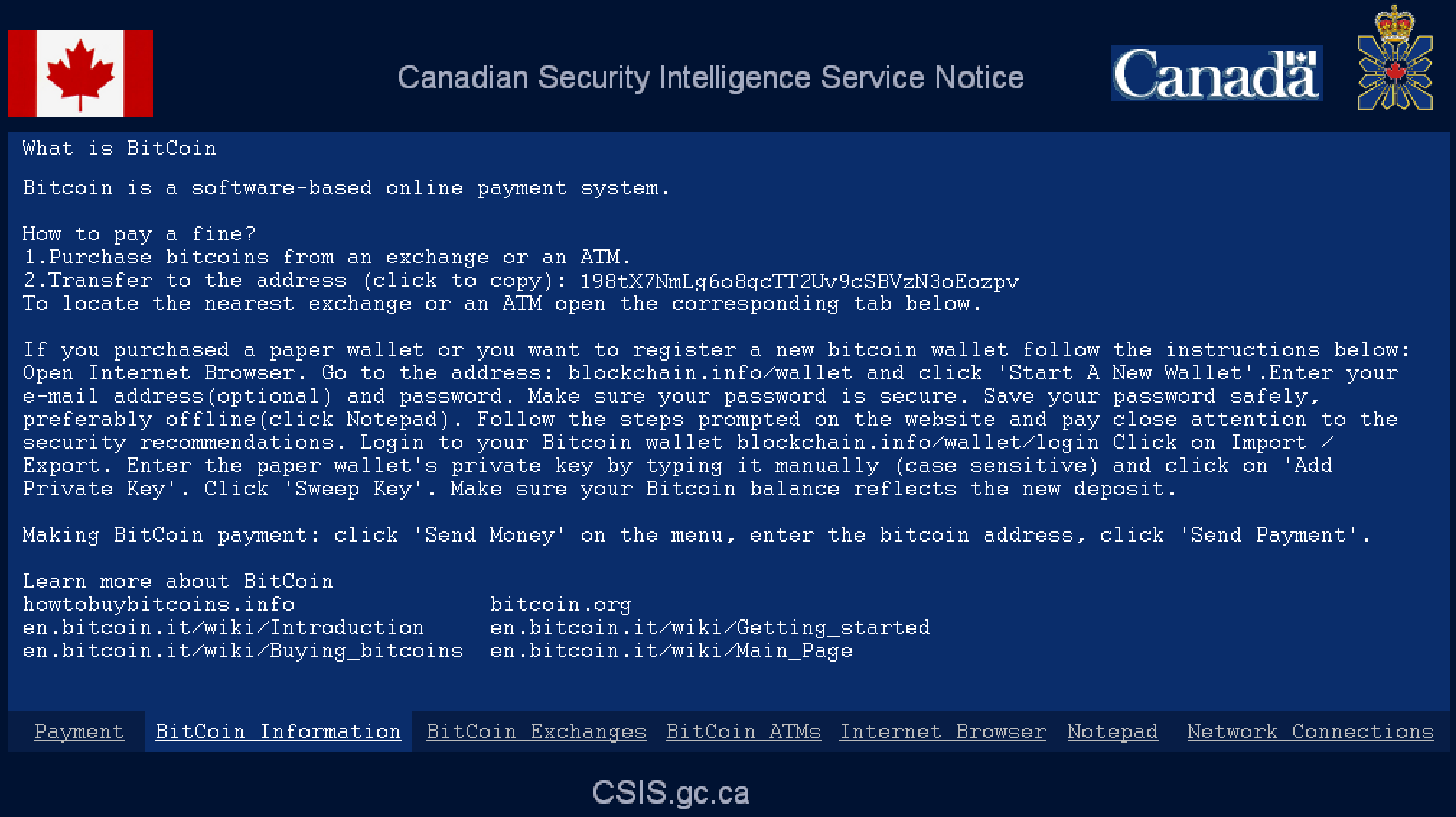
Amount: Transfer ID:

BTC 0.588

Internet connection is unavailable. Click Network Connections and connect to the Internet

If the fine is not paid, a warrant will be issued for your arrest,
which will be forwarded to your local authorities. You will be charged, fined, convicted for up to 5 years.

CSIS.gc.ca



Canadian Security Intelligence Service Notice

What is BitCoin

Bitcoin is a software-based online payment system.

How to pay a fine?
1.Purchase bitcoins from an exchange or an ATM.
2.Transfer to the address (click to copy): 198tX7NmLq6o8qcTT2Uv9cSBVzN3oEozpv
To locate the nearest exchange or an ATM open the corresponding tab below.

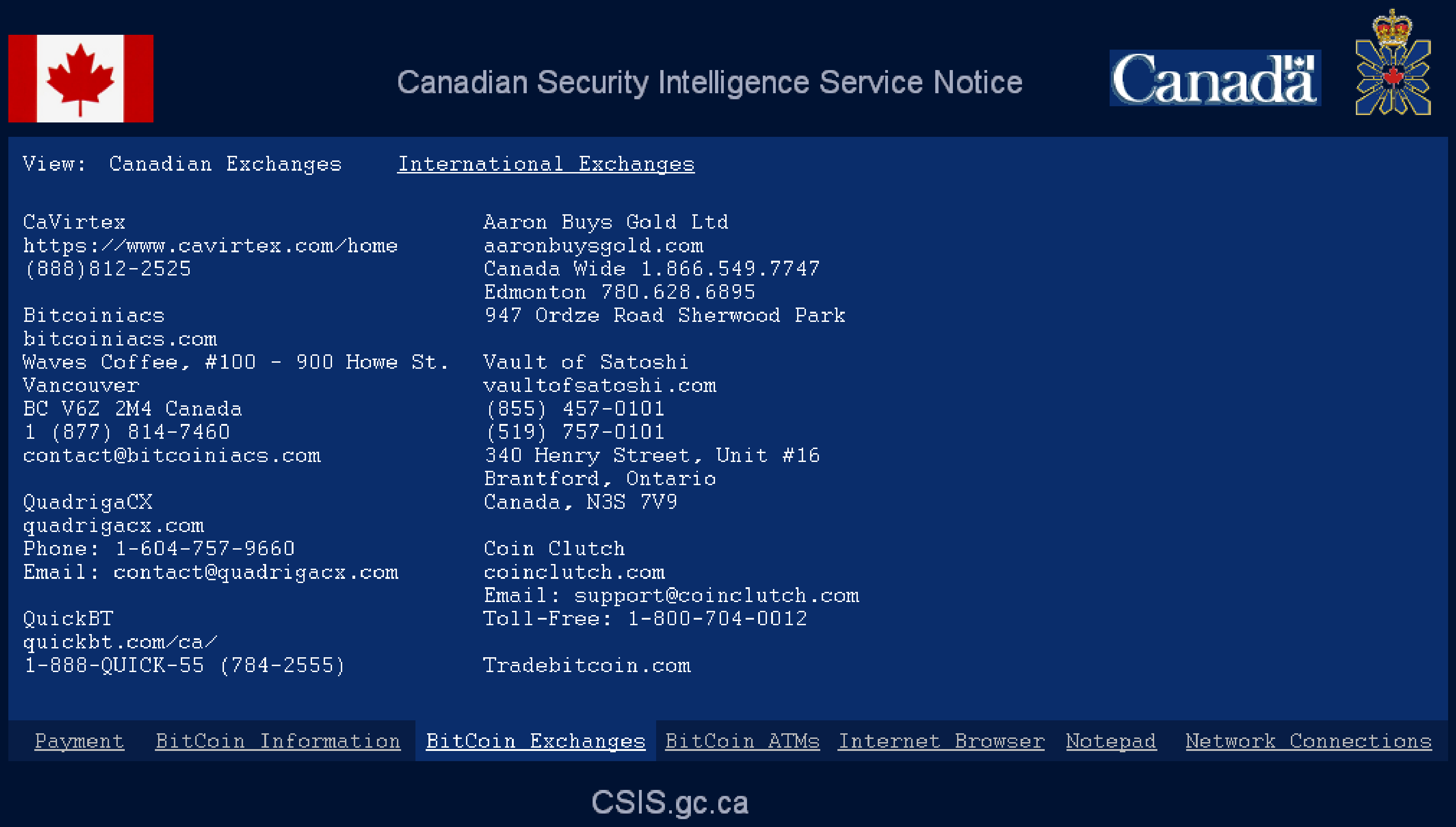
If you purchased a paper wallet or you want to register a new bitcoin wallet follow the instructions below: Open Internet Browser. Go to the address: blockchain.info/wallet and click 'Start A New Wallet'.Enter your e-mail address(optional) and password. Make sure your password is secure. Save your password safely, preferably offline(click Notepad). Follow the steps prompted on the website and pay close attention to the security recommendations. Login to your Bitcoin wallet blockchain.info/wallet/login Click on Import / Export. Enter the paper wallet's private key by typing it manually (case sensitive) and click on 'Add Private Key'. Click 'Sweep Key'. Make sure your Bitcoin balance reflects the new deposit.

Making BitCoin payment: click 'Send Money' on the menu, enter the bitcoin address, click 'Send Payment'.

Learn more about BitCoin

howtobuybitcoins.info	bitcoin.org
en.bitcoin.it/wiki/Introduction	en.bitcoin.it/wiki/Getting_started
en.bitcoin.it/wiki/Buying_bitcoins	en.bitcoin.it/wiki/Main_Page

CSIS.gc.ca



Canadian Security Intelligence Service Notice

View: Canadian Exchanges   International Exchanges

CaVirtex
https://www.cavirtex.com/home
(888)812-2525

Bitcoiniacs
bitcoiniacs.com
Waves Coffee, #100 - 900 Howe St.
Vancouver
BC V6Z 2M4 Canada
1 (877) 814-7460
contact@bitcoiniacs.com

QuadrigaCX
quadrigacx.com
Phone: 1-604-757-9660
Email: contact@quadrigacx.com

QuickBT
quickbt.com/ca/
1-888-QUICK-55 (784-2555)

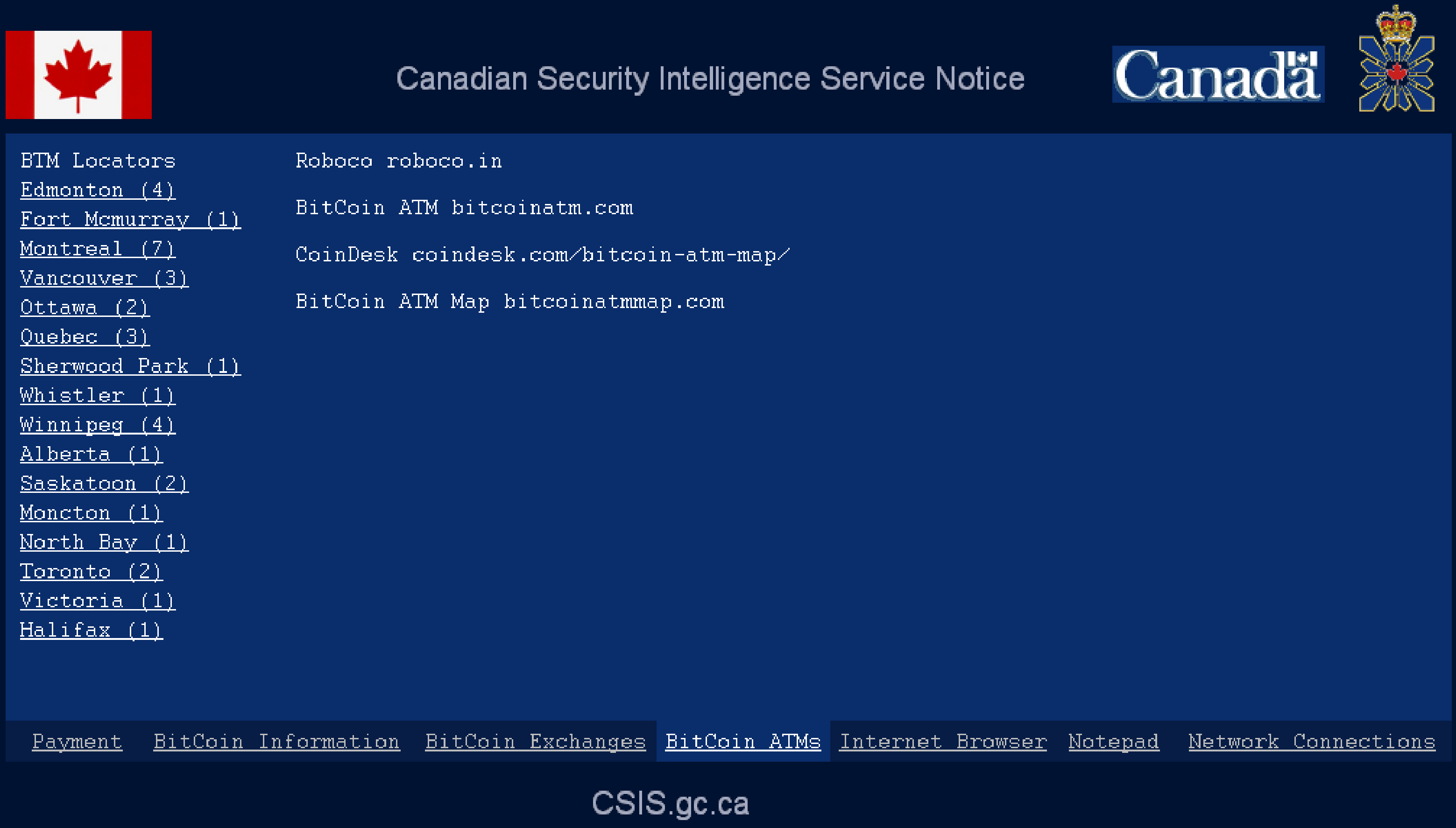
Aaron Buys Gold Ltd
aaronbuysgold.com
Canada Wide 1.866.549.7747
Edmonton 780.628.6895
947 Ordze Road Sherwood Park

Vault of Satoshi
vaultofsatoshi.com
(855) 457-0101
(519) 757-0101
340 Henry Street, Unit #16
Brantford, Ontario
Canada, N3S 7V9

Coin Clutch
coinclutch.com
Email: support@coinclutch.com
Toll-Free: 1-800-704-0012

Tradebitcoin.com

CSIS.gc.ca



Canadian Security Intelligence Service Notice

BTM Locators
Edmonton (4)
Fort Mcmurray (1)
Montreal (7)
Vancouver (3)
Ottawa (2)
Quebec (3)
Sherwood Park (1)
Whistler (1)
Winnipeg (4)
Alberta (1)
Saskatoon (2)
Moncton (1)
North Bay (1)
Toronto (2)
Victoria (1)
Halifax (1)

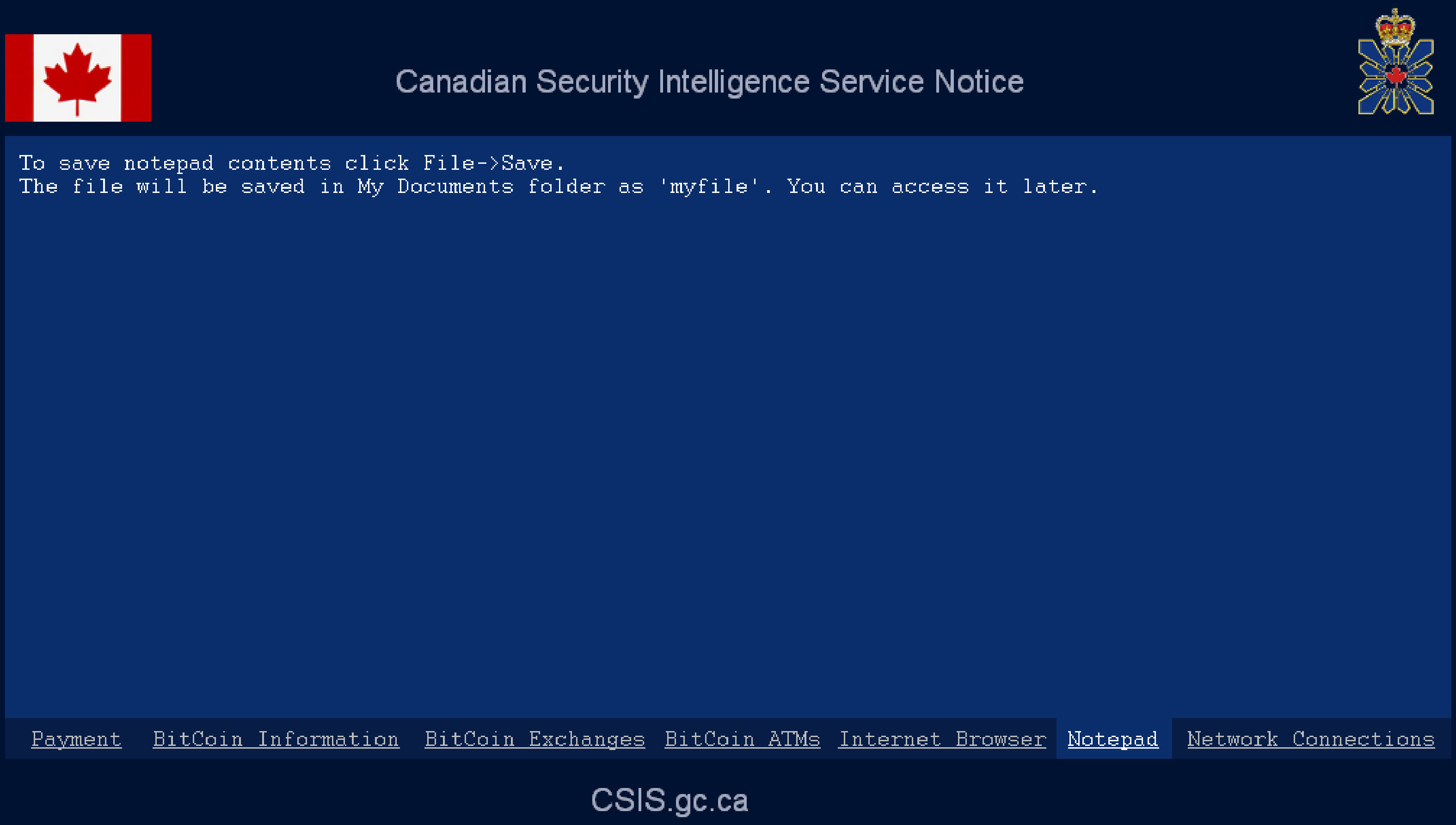
Roboco roboco.in

BitCoin ATM bitcoinatm.com

CoinDesk coindesk.com/bitcoin-atm-map/

BitCoin ATM Map bitcoinatmmap.com

CSIS.gc.ca



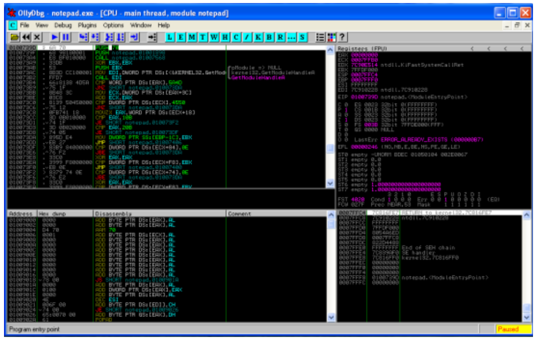
Canadian Security Intelligence Service Notice

```
To save notepad contents click File->Save.
The file will be saved in My Documents folder as 'myfile'. You can access it later.
```

CSIS.gc.ca

Virlock As A Common Malware

Debugging Tools

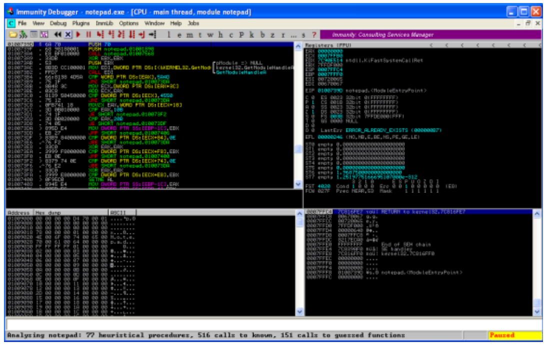


OllyDbg - notepad.exe - [CPU - main thread, module notepad]

Program entry point

Paused

ollydbg
http://www.ollydbg.de/

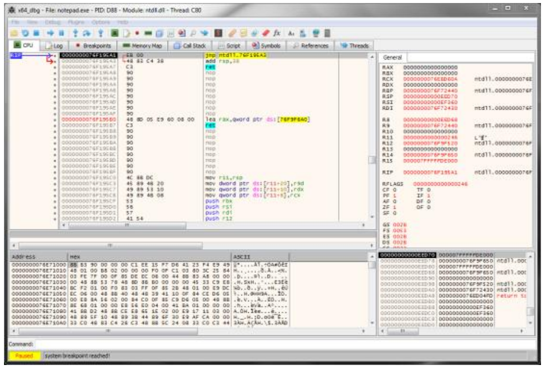


Immunity Debugger - notepad.exe - [CPU - main thread, module notepad]

Analysing notepad: 77 heuristical procedures, 516 calls to known, 151 calls to guessed functions

Paused

immunity debugger
http://www.immunityinc.com/products/debugger/



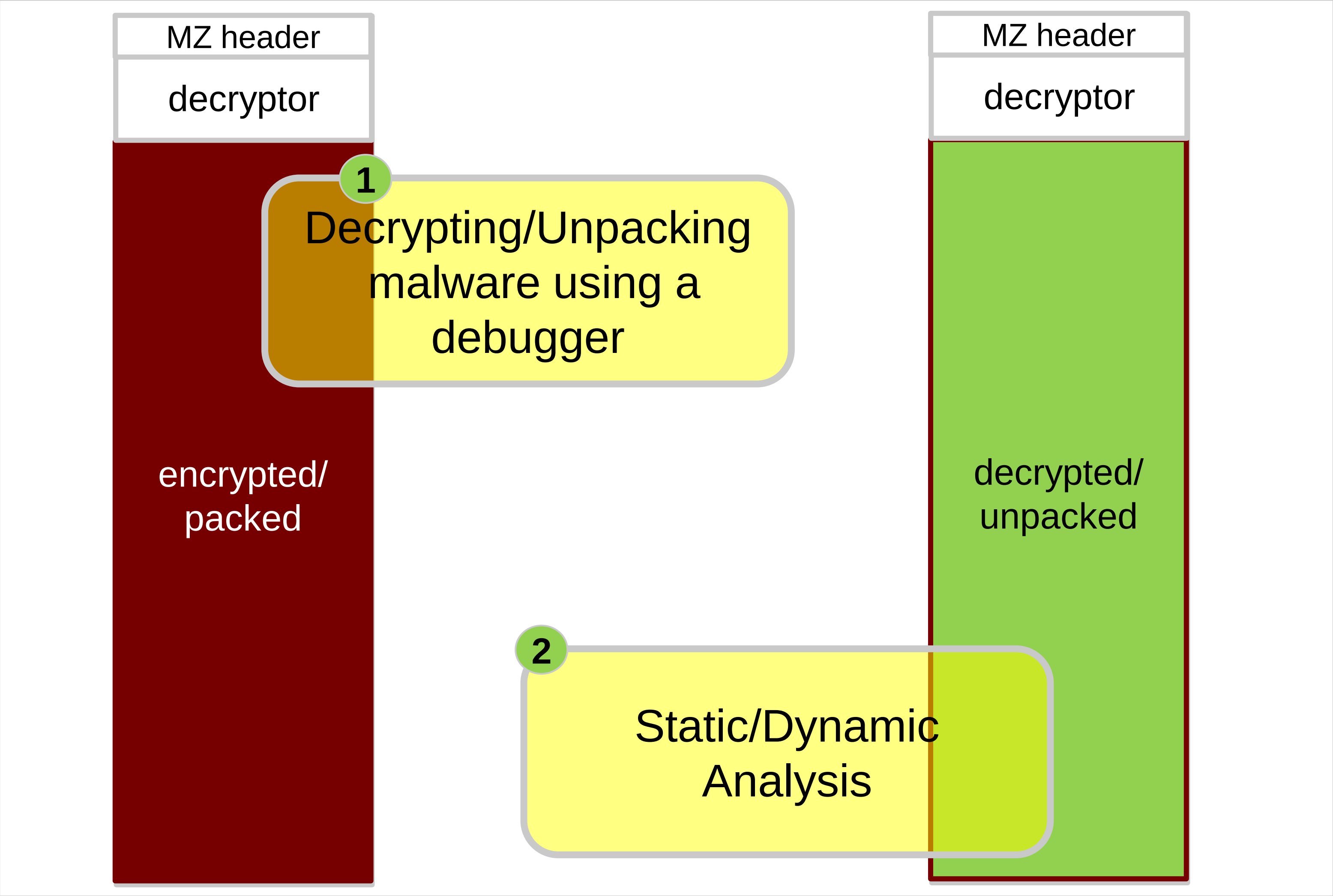
x64_dbg - File: notepad.exe - PID: D88 - Module: ntdll.dll - Thread: C80

Paused

system breakpoint reached!

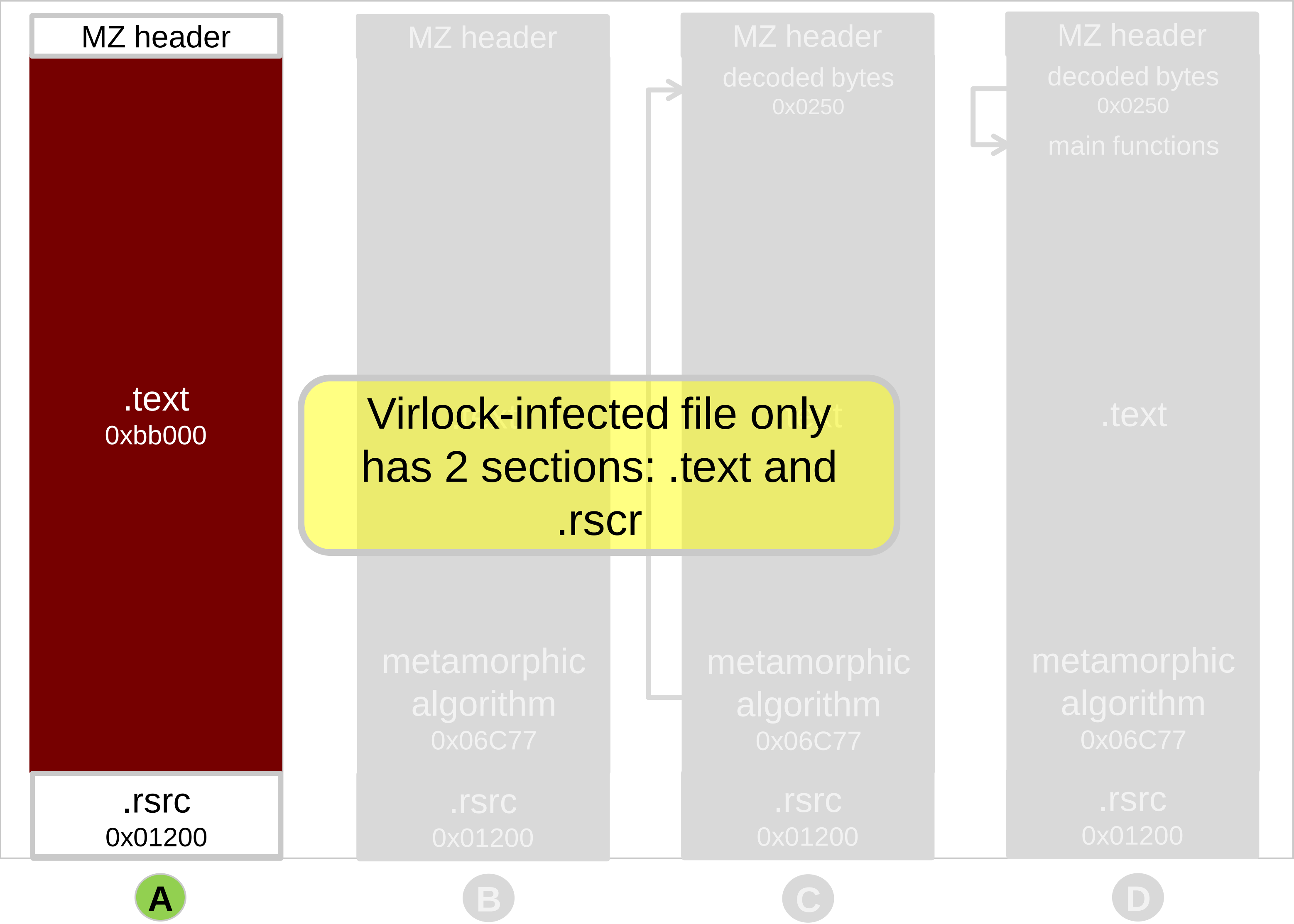
x64dbg
http://x64dbg.com/

Common Reversing



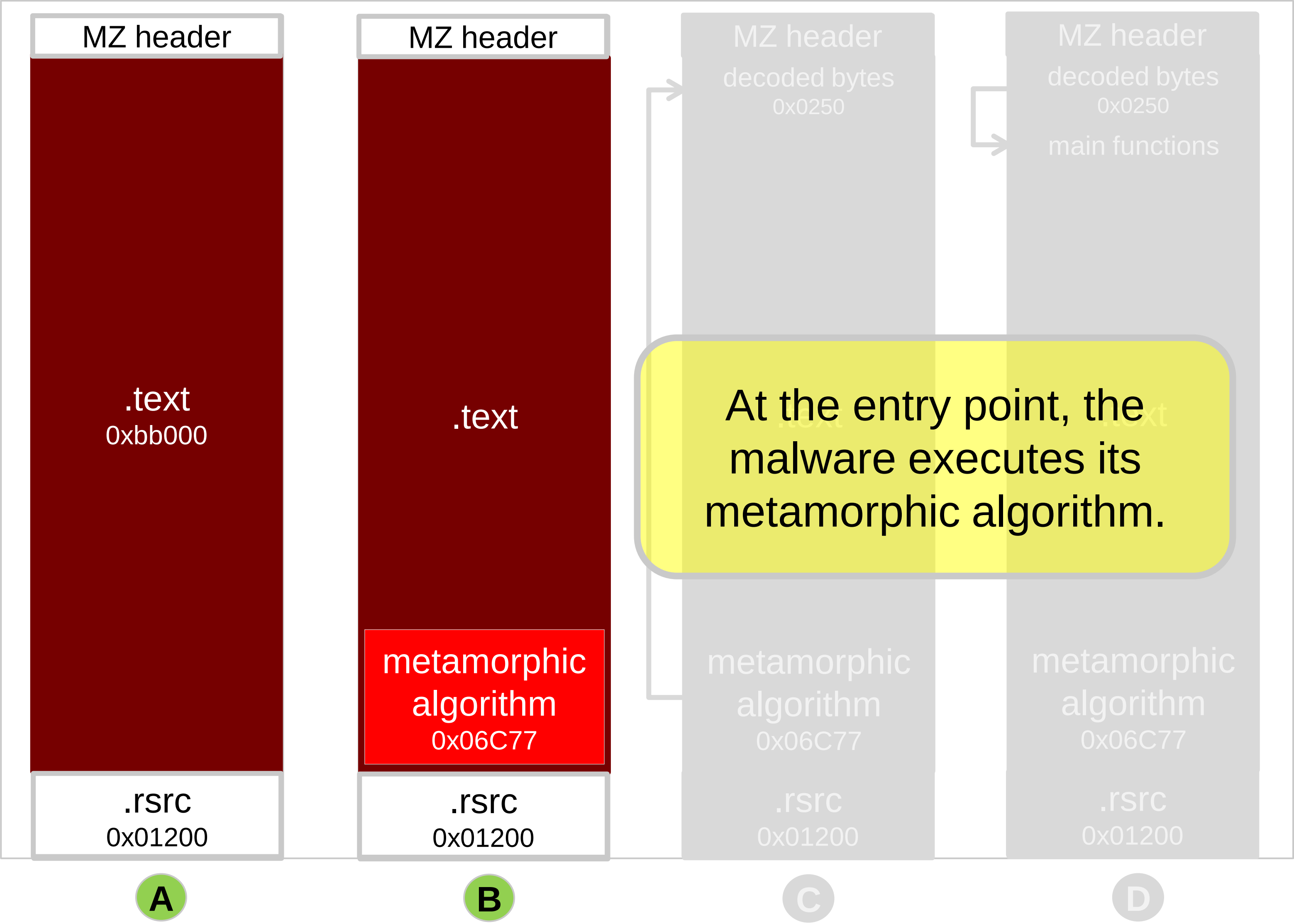
Reversing Stages

Reversing Stages



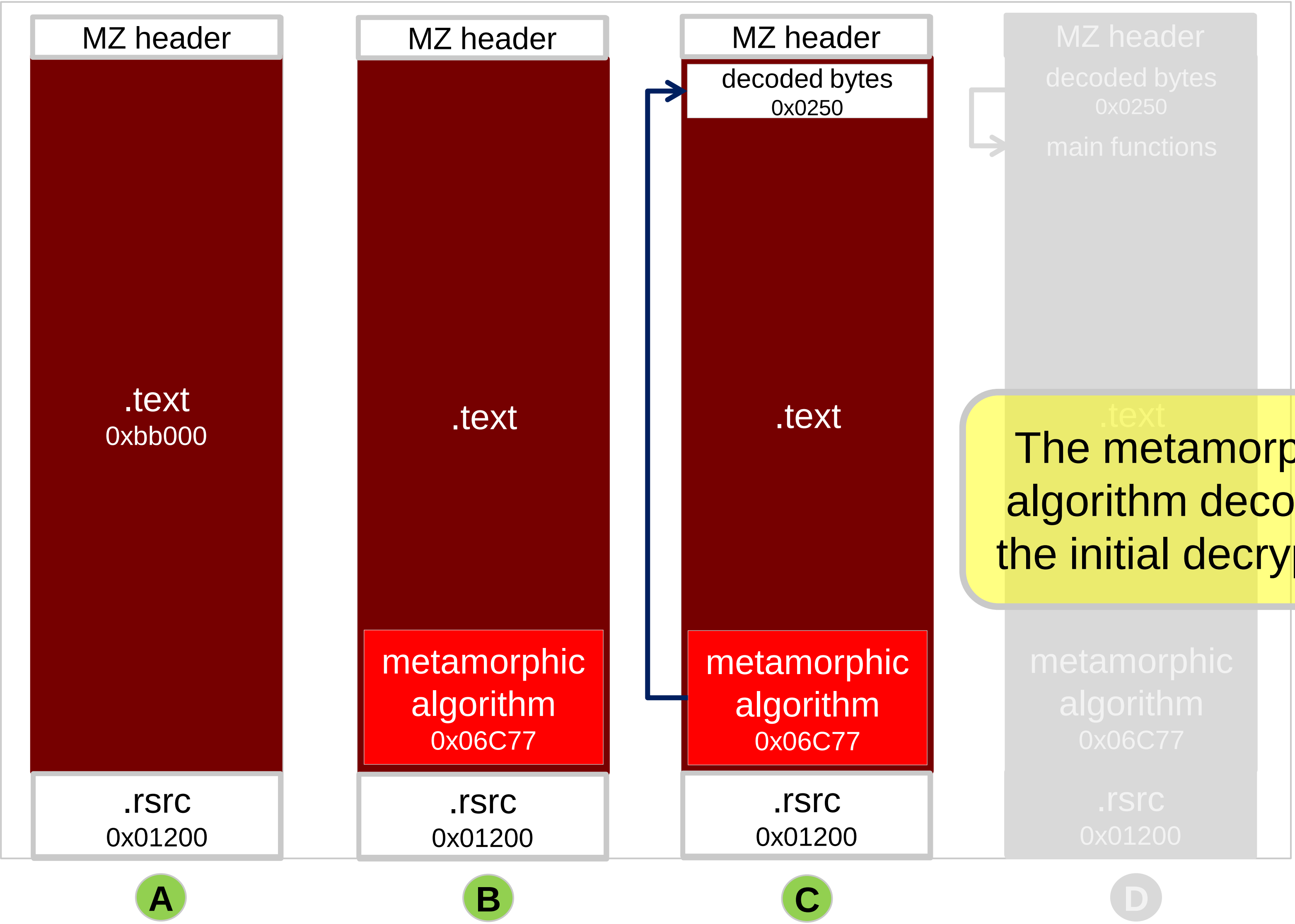
Virlock-infected file only has 2 sections: .text and .rscr

Reversing Stages



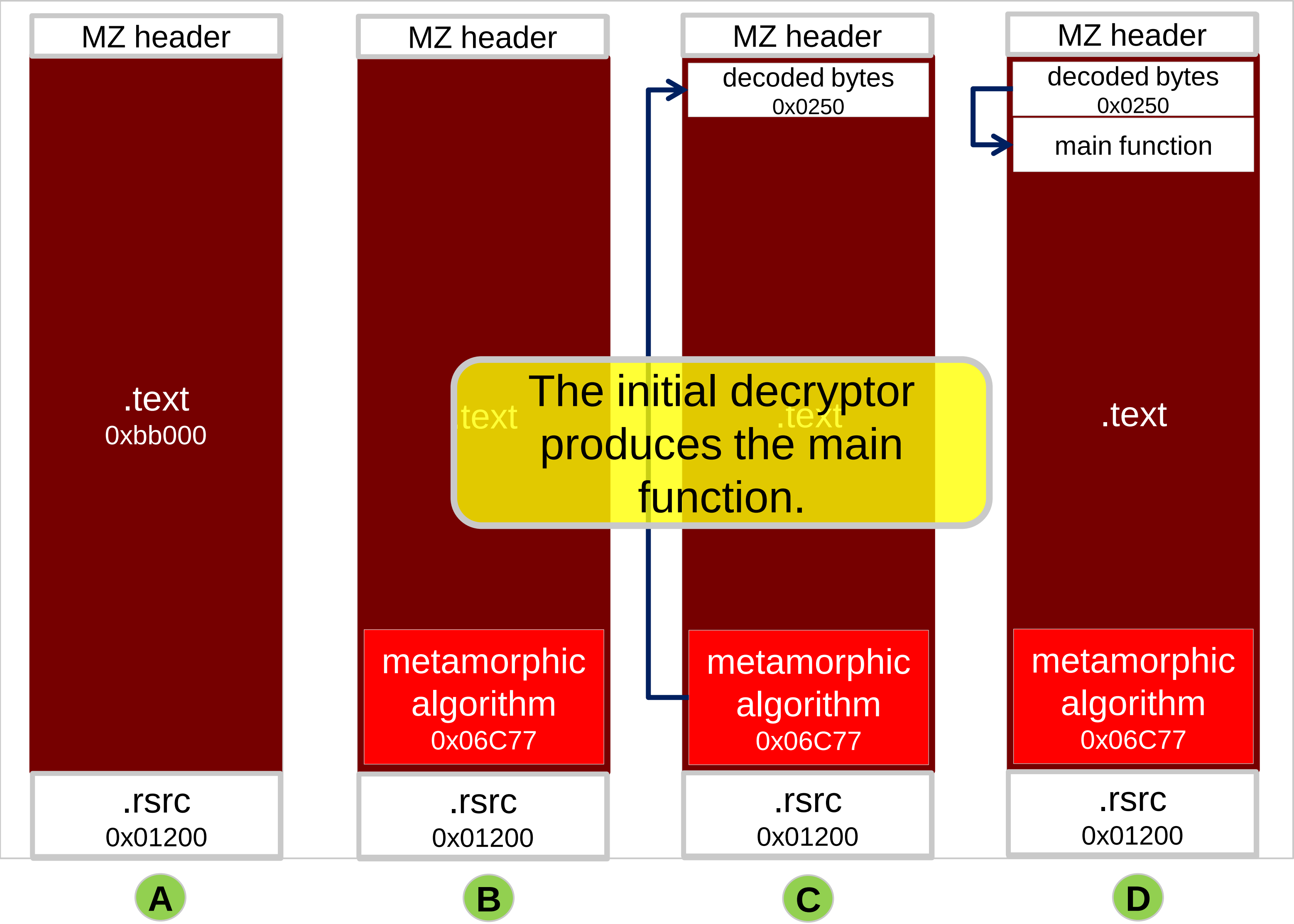
At the entry point, the malware executes its metamorphic algorithm.

Reversing Stages



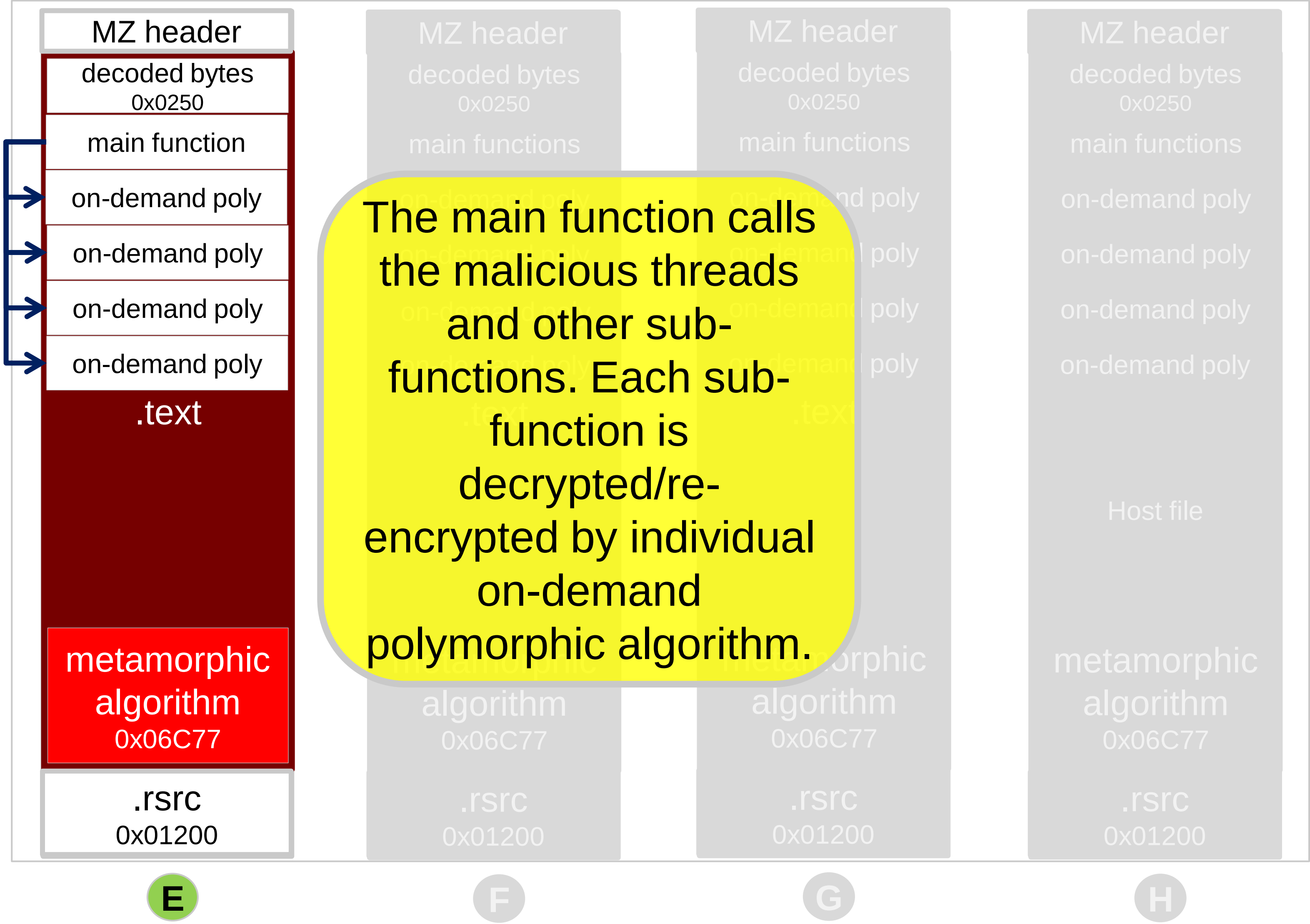
The metamorphic algorithm decodes the initial decryptor.

Reversing Stages



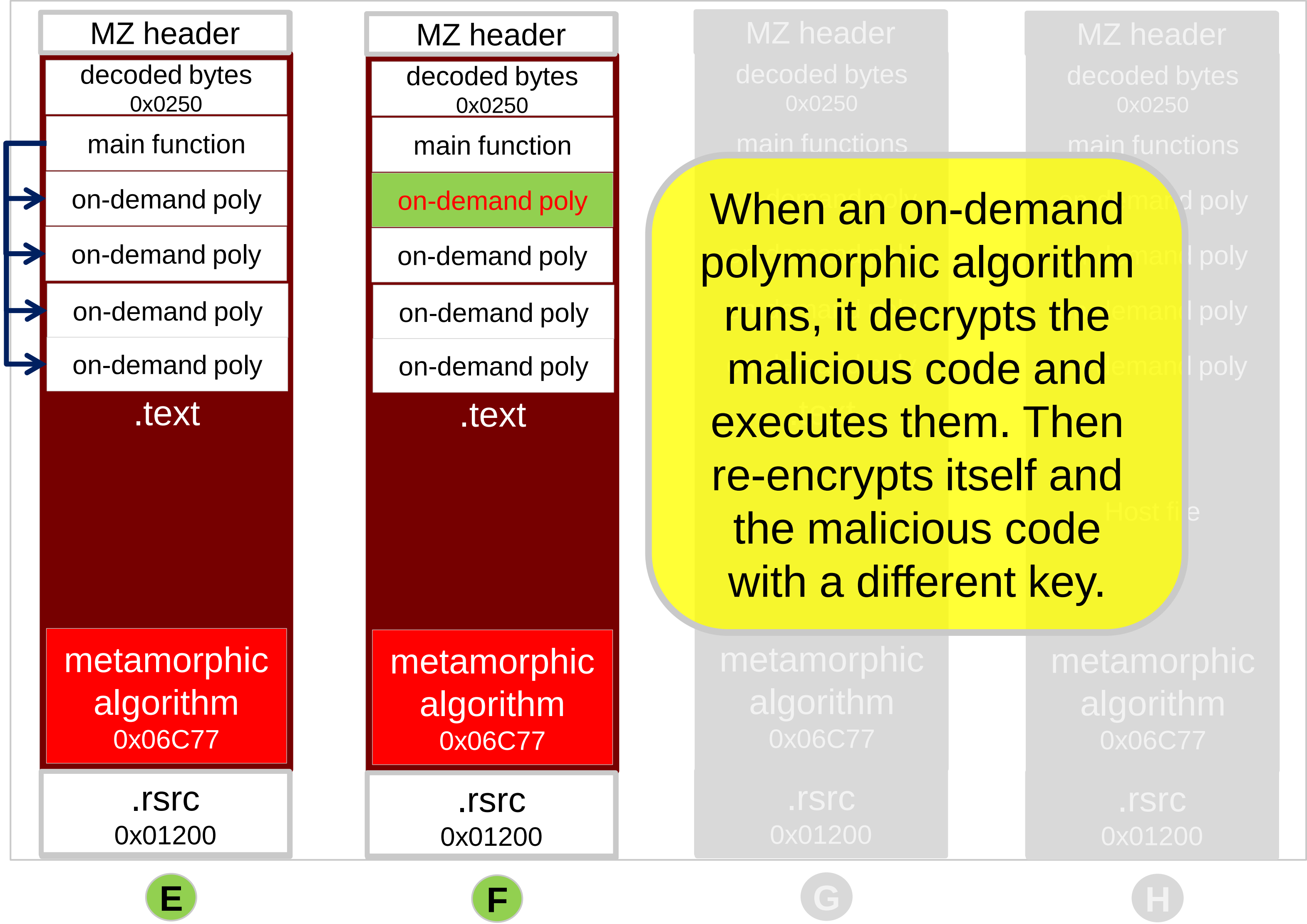
The initial decryptor produces the main function.

Reversing Stages



The main function calls the malicious threads and other sub-functions. Each sub-function is decrypted/re-encrypted by individual on-demand polymorphic algorithm.

Reversing Stages



MZ header

decoded bytes
0x0250

main function

on-demand poly

on-demand poly

on-demand poly

on-demand poly

.text

metamorphic algorithm
0x06C77

.rsrc
0x01200

E

MZ header

decoded bytes
0x0250

main function

on-demand poly

on-demand poly

on-demand poly

on-demand poly

.text

metamorphic algorithm
0x06C77

.rsrc
0x01200

F

MZ header

decoded bytes
0x0250

main functions

metamorphic algorithm
0x06C77

.rsrc
0x01200

G

MZ header

decoded bytes
0x0250

main functions

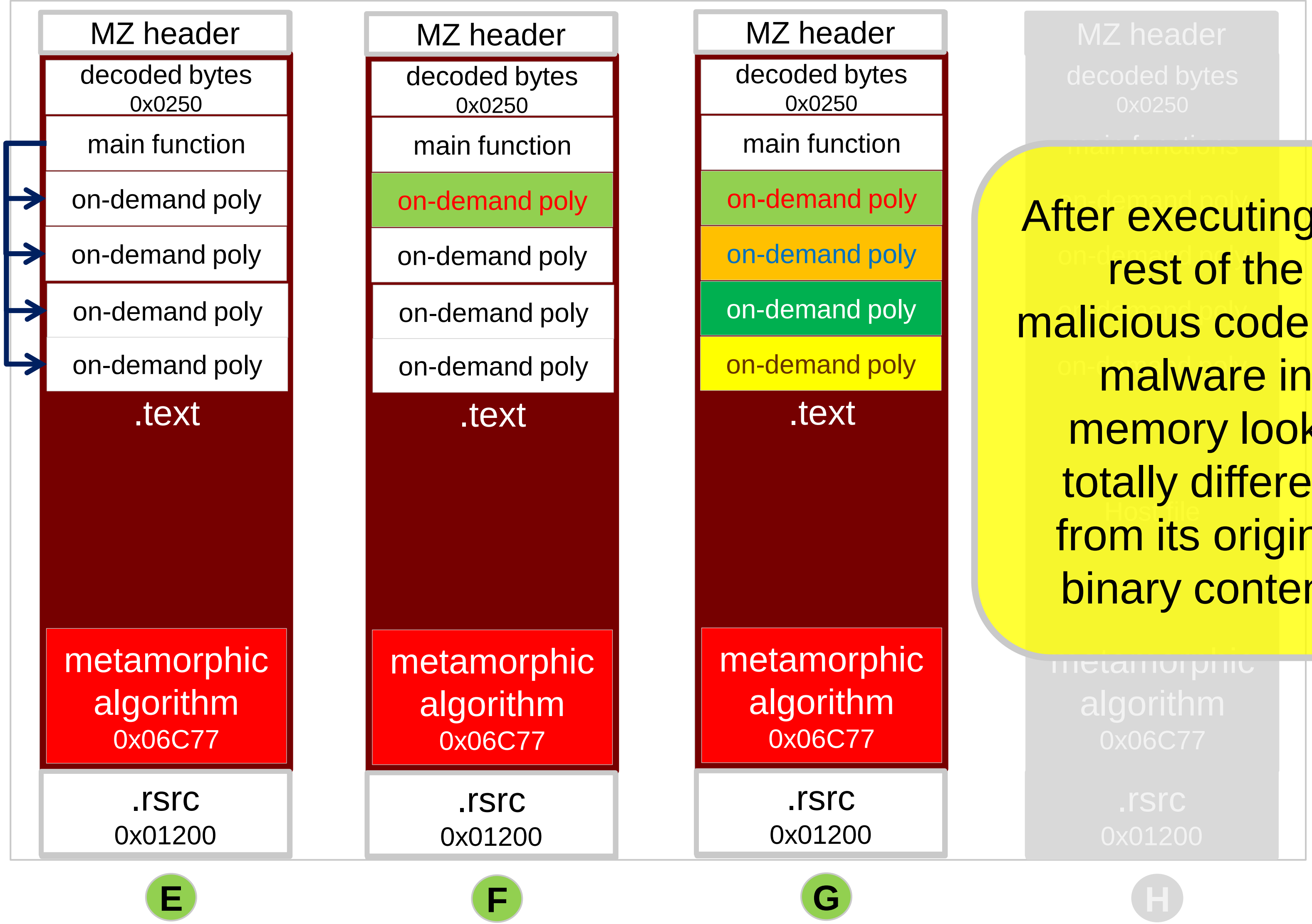
metamorphic algorithm
0x06C77

.rsrc
0x01200

H

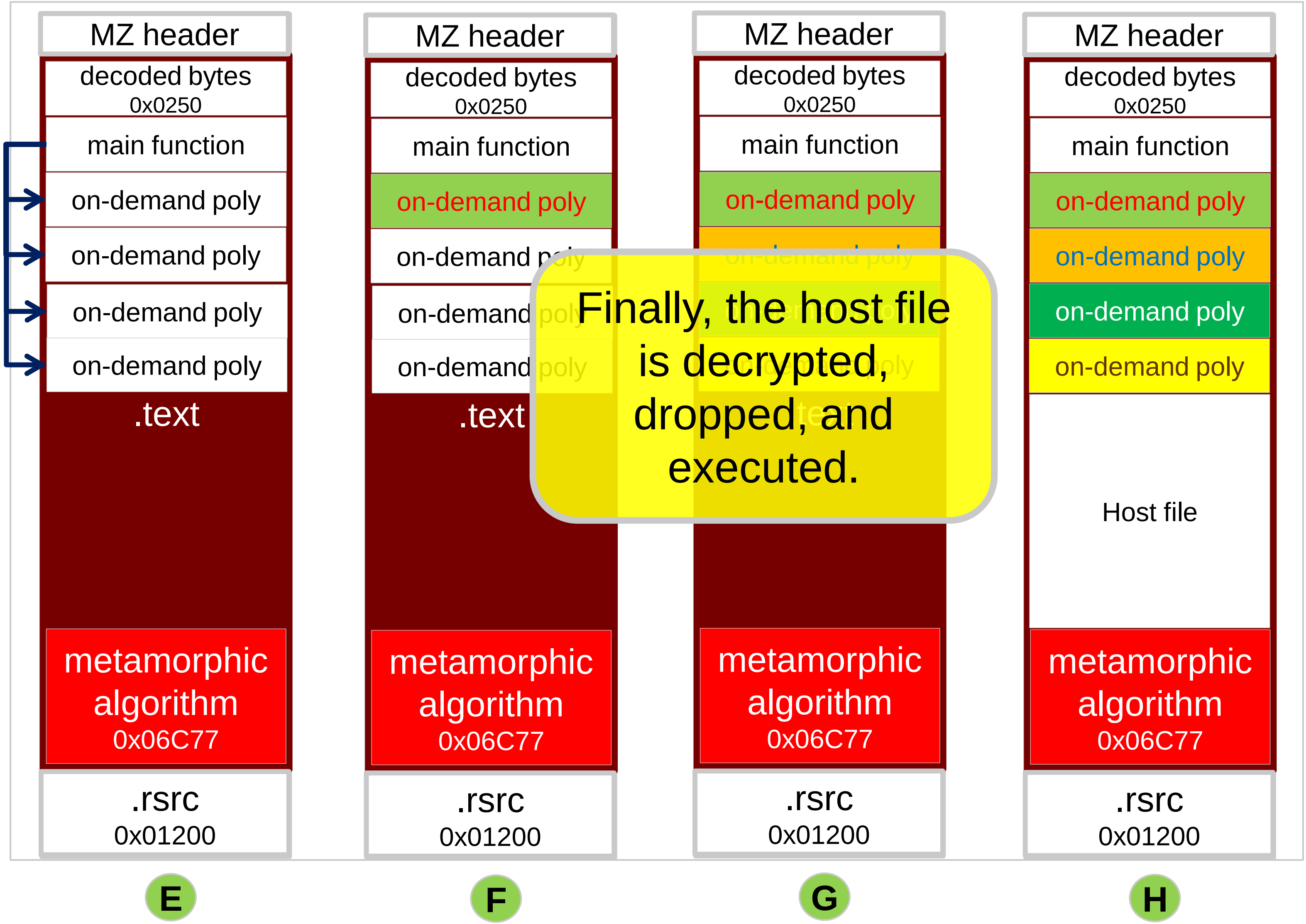
When an on-demand polymorphic algorithm runs, it decrypts the malicious code and executes them. Then re-encrypts itself and the malicious code with a different key.

Reversing Stages



After executing the rest of the malicious code, the malware in memory looks totally different from its original binary content.

Reversing Stages



Finally, the host file is decrypted, dropped, and executed.

Virlock As A File Infector

Cleaning: How To Clean An Infected File

Basics:

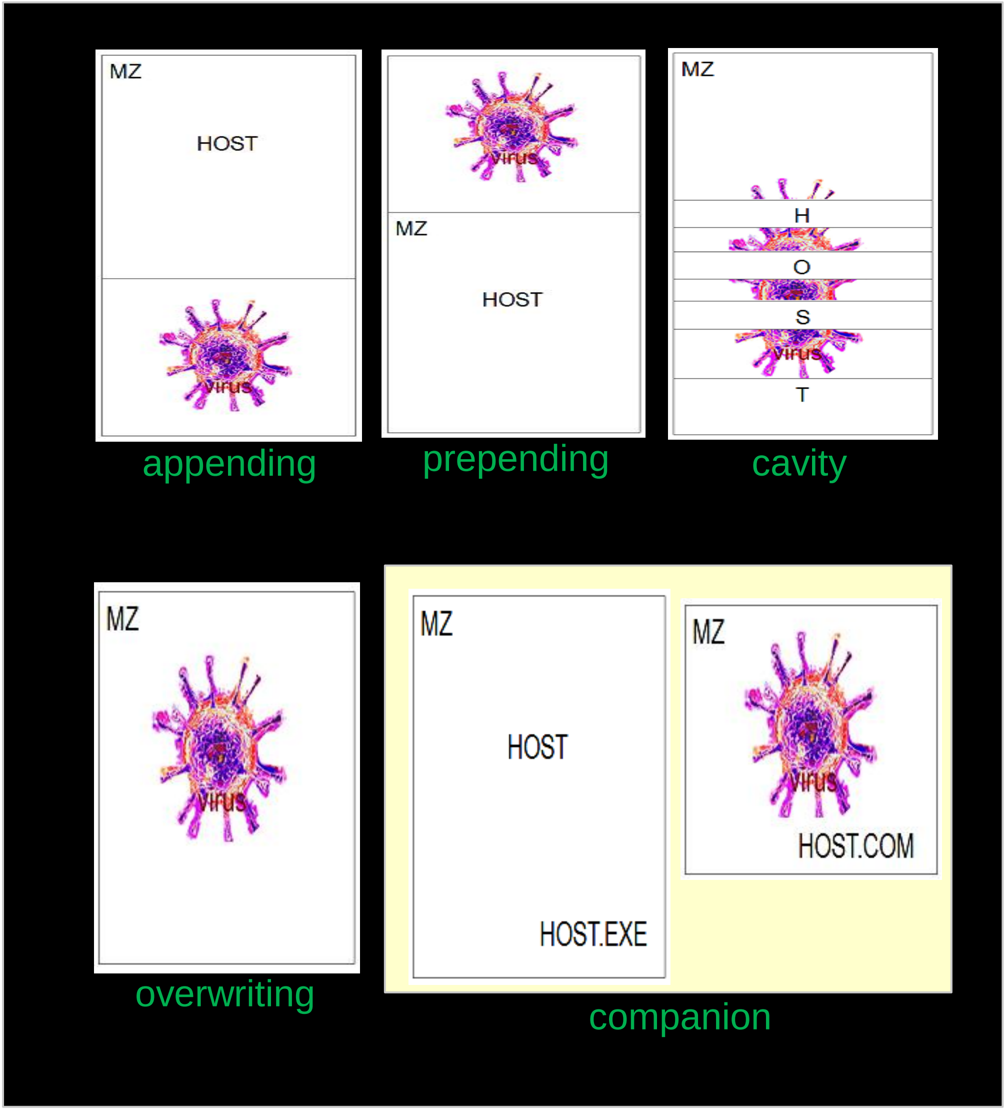
- Determine the kind of virus
- Determine how to extract and restore the host file

Different Kinds Of File Infectors

Basics:

- Appending
- Prepending
- Cavity
- Overwriting
- Companion

Different Kinds Of File Infectors



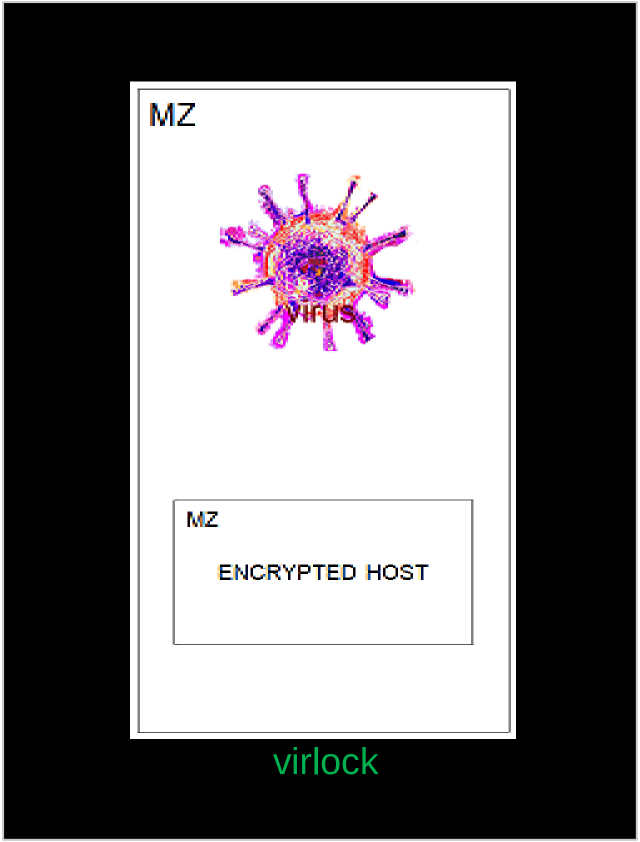
appending

prepending

cavity

overwriting

companion



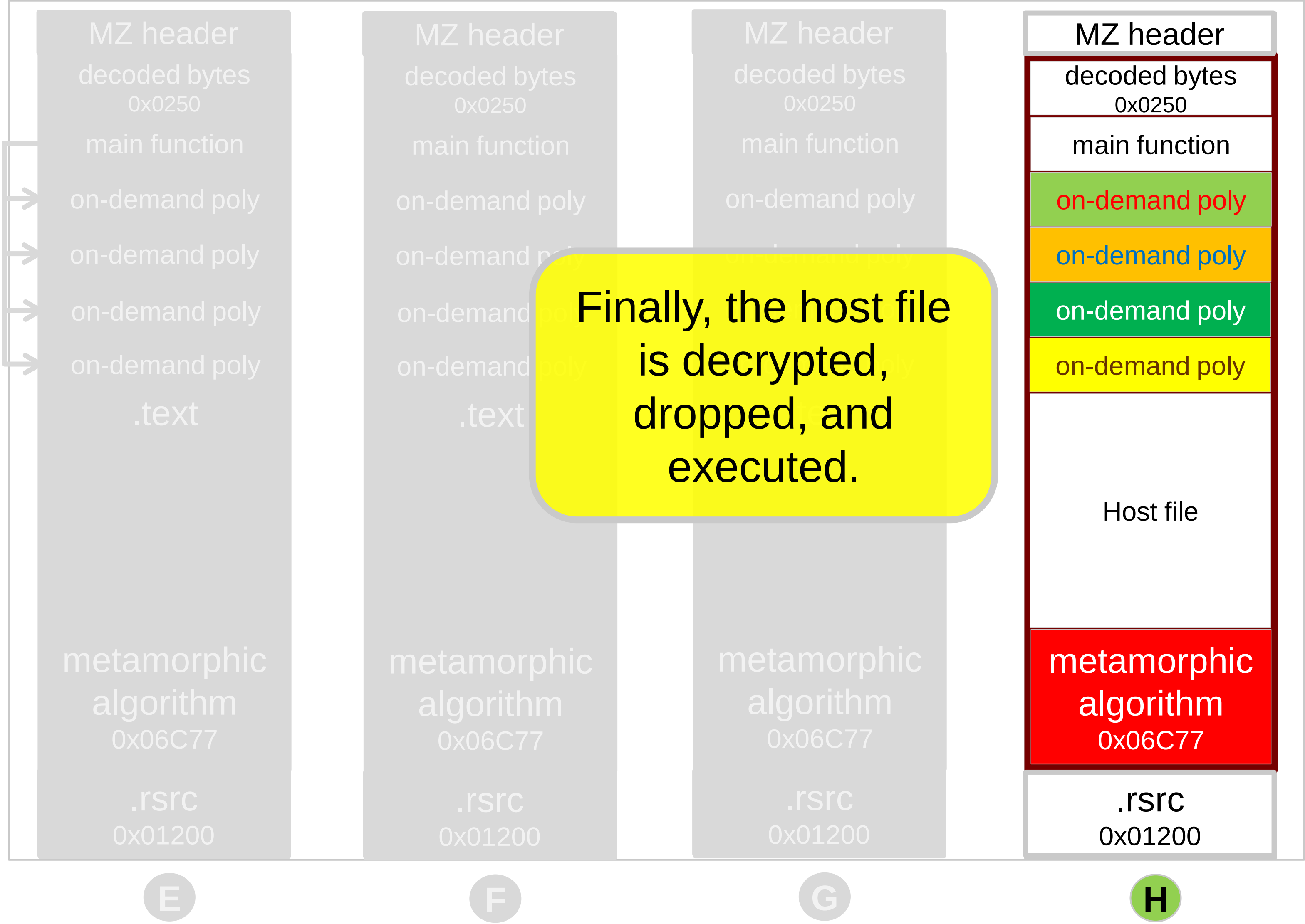
virlock

Cleaning: Extracting The Host File From Virlock

Details:

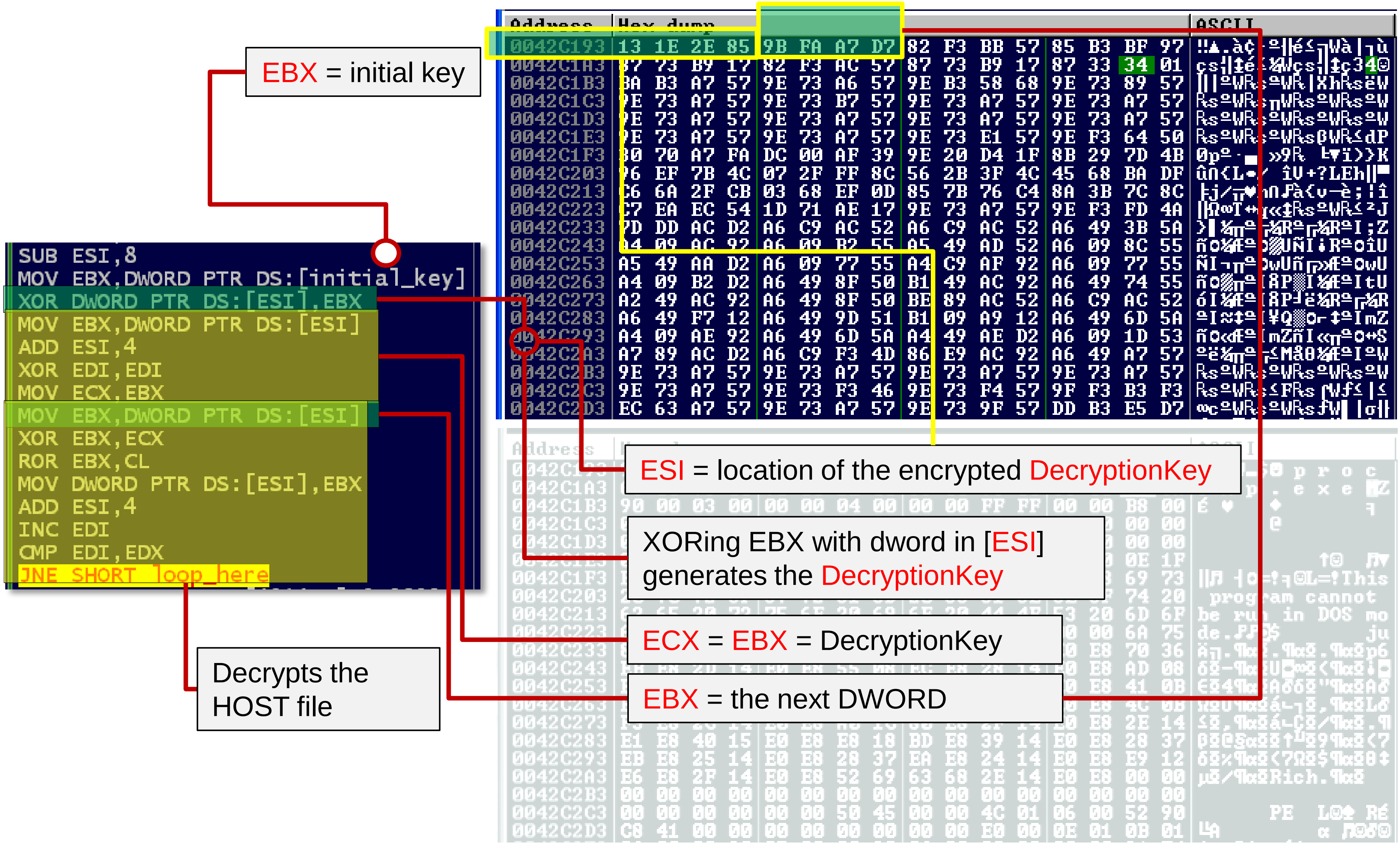
- Host file is encrypted and embedded within the malware
- DecryptionKey can be found within the malware
- DecryptionKey is encrypted using a simple XOR
- Uses a simple decryption algorithm to extract the host file

Reversing Stages



Finally, the host file is decrypted, dropped, and executed.

Cleaning: Extracting The Host File From Virlock



EBX = initial key

```
SUB ESI,8
MOV EBX,DWORD PTR DS:[initial_key]
XOR DWORD PTR DS:[ESI],EBX
MOV EBX,DWORD PTR DS:[ESI]
ADD ESI,4
XOR EDI,EDI
MOV ECX,EBX
MOV EBX,DWORD PTR DS:[ESI]
XOR EBX,ECX
ROR EBX,CL
MOV DWORD PTR DS:[ESI],EBX
ADD ESI,4
INC EDI
CMP EDI,EDX
JNE SHORT loop_here
```

```
Address   Hex dump
0042C193  13 1E 2E 85 9B FA A7 D7 82 F3 BB 57 85 B3 BF 97
0042C1A3  87 73 B9 17 82 F3 AC 57 87 73 B9 17 87 33 34 01
0042C1B3  BA B3 A7 57 9E 73 A6 57 9E B3 58 68 9E 73 89 57
0042C1C3  9E 73 A7 57 9E 73 B7 57 9E 73 A7 57 9E 73 A7 57
0042C1D3  9E 73 A7 57 9E 73 A7 57 9E 73 A7 57 9E 73 A7 57
0042C1E3  9E 73 A7 57 9E 73 A7 57 9E 73 E1 57 9E F3 64 50
0042C1F3  B0 70 A7 FA DC 00 AF 39 9E 20 D4 1F 8B 29 7D 4B
0042C203  96 EF 7B 4C 07 2F FF 8C 56 2B 3F 4C 45 68 BA DF
0042C213  C6 6A 2F CB 03 68 EF 0D 85 7B 76 C4 8A 3B 7C 8C
0042C223  C7 EA EC 54 1D 71 AE 17 9E 73 A7 57 9E F3 FD 4A
0042C233  7D DD AC D2 A6 C9 AC 52 A6 C9 AC 52 A6 49 3B 5A
0042C243  A4 09 AC 92 A6 09 B2 55 A5 49 AD 52 A6 09 8C 55
0042C253  A5 49 AA D2 A6 09 77 55 A4 C9 AF 92 A6 09 77 55
0042C263  A4 09 B2 D2 A6 49 8F 50 B1 49 AC 92 A6 49 74 55
0042C273  A2 49 AC 92 A6 49 8F 50 BE 89 AC 52 A6 C9 AC 52
0042C283  A6 49 F7 12 A6 49 9D 51 B1 09 A9 12 A6 49 6D 5A
0042C293  A4 09 AE 92 A6 49 6D 5A A4 49 AE D2 A6 09 1D 53
0042C2A3  A7 89 AC D2 A6 C9 F3 4D 86 E9 AC 92 A6 49 A7 57
0042C2B3  9E 73 A7 57 9E 73 A7 57 9E 73 A7 57 9E 73 A7 57
0042C2C3  9E 73 A7 57 9E 73 F3 46 9E 73 F4 57 9F F3 B3 F3
0042C2D3  EC 63 A7 57 9E 73 A7 57 9E 73 9F 57 DD B3 E5 D7
```

```
Address
0042C283  E1 E8 40 15 E0 E8 E8 18 BD E8 39 14 E0 E8 28 37
0042C293  EB E8 25 14 E0 E8 28 37 EA E8 24 14 E0 E8 E9 12
0042C2A3  E6 E8 2F 14 E0 E8 52 69 63 68 2E 14 E0 E8 00 00
0042C2B3  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0042C2C3  00 00 00 00 00 00 50 45 00 00 4C 01 06 00 52 90
0042C2D3  C8 41 00 00 00 00 00 00 00 00 E0 00 0E 01 0B 01
```

ESI = location of the encrypted DecryptionKey

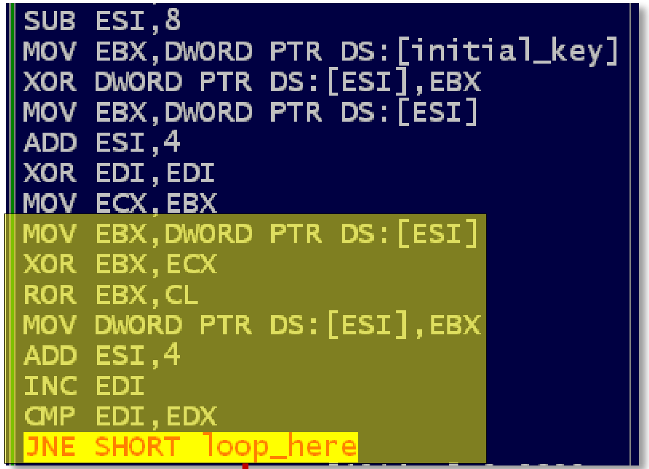
XORing EBX with dword in [ESI] generates the DecryptionKey

ECX = EBX = DecryptionKey

EBX = the next DWORD

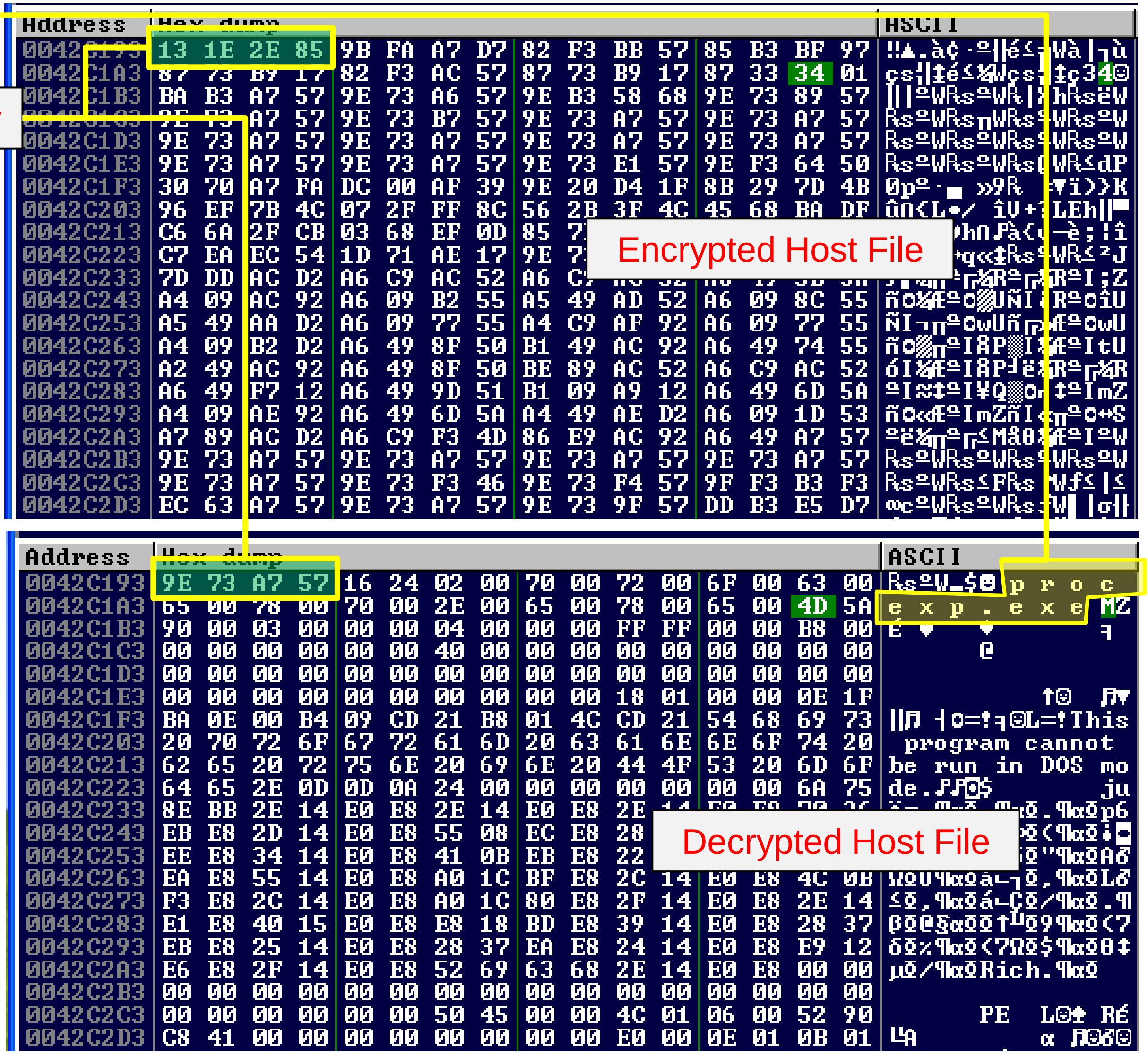
Decrypts the HOST file

Cleaning: Extracting The Host File From Virlock



```
SUB ESI,8
MOV EBX,DWORD PTR DS:[initial_key]
XOR DWORD PTR DS:[ESI],EBX
MOV EBX,DWORD PTR DS:[ESI]
ADD ESI,4
XOR EDI,EDI
MOV ECX,EBX
MOV EBX,DWORD PTR DS:[ESI]
XOR EBX,ECX
ROR EBX,CL
MOV DWORD PTR DS:[ESI],EBX
ADD ESI,4
INC EDI
CMP EDI,EDX
JNE SHORT loop_here
```

Decrypts the HOST file



DecryptionKey

Original Host Filename

Encrypted Host File

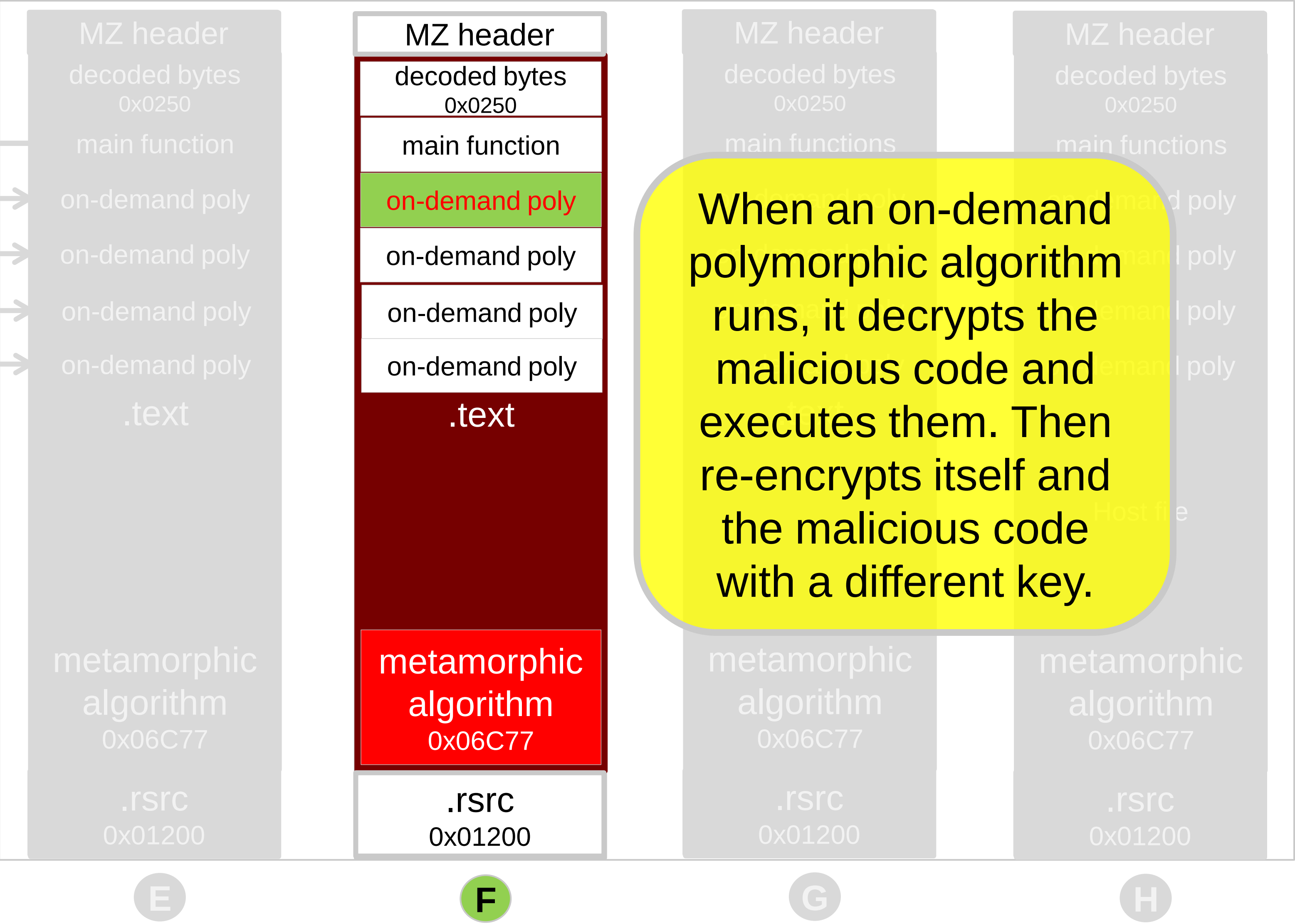
```
Address   Hex dump
0042C193  13 1E 2E 85 9B FA A7 D7 82 F3 BB 57 85 B3 BF 97
0042C1A3  87 73 B9 17 82 F3 AC 57 87 73 B9 17 87 33 34 01
0042C1B3  BA B3 A7 57 9E 73 A6 57 9E B3 58 68 9E 73 89 57
0042C1C3  9E 73 A7 57 9E 73 B7 57 9E 73 A7 57 9E 73 A7 57
0042C1D3  9E 73 A7 57 9E 73 A7 57 9E 73 A7 57 9E 73 A7 57
0042C1E3  9E 73 A7 57 9E 73 A7 57 9E 73 E1 57 9E F3 64 50
0042C1F3  30 70 A7 FA DC 00 AF 39 9E 20 D4 1F 8B 29 7D 4B
0042C203  96 EF 7B 4C 07 2F FF 8C 56 2B 3F 4C 45 68 BA DF
0042C213  C6 6A 2F CB 03 68 EF 0D 85 [illegible]
0042C223  C7 EA EC 54 1D 71 AE 17 9E [illegible]
0042C233  7D DD AC D2 A6 C9 AC 52 A6 [illegible]
0042C243  A4 09 AC 92 A6 09 B2 55 A5 49 AD 52 A6 09 8C 55
0042C253  A5 49 AA D2 A6 09 77 55 A4 C9 AF 92 A6 09 77 55
0042C263  A4 09 B2 D2 A6 49 8F 50 B1 49 AC 92 A6 49 74 55
0042C273  A2 49 AC 92 A6 49 8F 50 BE 89 AC 52 A6 C9 AC 52
0042C283  A6 49 F7 12 A6 49 9D 51 B1 09 A9 12 A6 49 6D 5A
0042C293  A4 09 AE 92 A6 49 6D 5A A4 49 AE D2 A6 09 1D 53
0042C2A3  A7 89 AC D2 A6 C9 F3 4D 86 E9 AC 92 A6 49 A7 57
0042C2B3  9E 73 A7 57 9E 73 A7 57 9E 73 A7 57 9E 73 A7 57
0042C2C3  9E 73 A7 57 9E 73 F3 46 9E 73 F4 57 9F F3 B3 F3
0042C2D3  EC 63 A7 57 9E 73 A7 57 9E 73 9F 57 DD B3 E5 D7
```

Decrypted Host File

```
Address   Hex dump                                         ASCII
0042C193  9E 73 A7 57 16 24 02 00 70 00 72 00 6F 00 63 00  p r o c
0042C1A3  65 00 78 00 70 00 2E 00 65 00 78 00 65 00 4D 5A  e x p . e x e MZ
0042C1B3  90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00
0042C1C3  00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00
0042C1D3  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0042C1E3  00 00 00 00 00 00 00 00 00 00 18 01 00 00 0E 1F
0042C1F3  BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 69 73  This
0042C203  20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20   program cannot
0042C213  62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F  be run in DOS mo
0042C223  64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 6A 75  de.    $       ju
0042C233  8E BB 2E 14 E0 E8 2E 14 E0 E8 2E [illegible]
0042C243  EB E8 2D 14 E0 E8 55 08 EC E8 28 [illegible]
0042C253  EE E8 34 14 E0 E8 41 0B EB E8 22 [illegible]
0042C263  EA E8 55 14 E0 E8 A0 1C BF E8 2C 14 E0 E8 4C 0B
0042C273  F3 E8 2C 14 E0 E8 A0 1C 80 E8 2F 14 E0 E8 2E 14
0042C283  E1 E8 40 15 E0 E8 E8 18 BD E8 39 14 E0 E8 28 37
0042C293  EB E8 25 14 E0 E8 28 37 EA E8 24 14 E0 E8 E9 12
0042C2A3  E6 E8 2F 14 E0 E8 52 69 63 68 2E 14 E0 E8 00 00  Rich.
0042C2B3  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0042C2C3  00 00 00 00 00 00 50 45 00 00 4C 01 06 00 52 90  PE
0042C2D3  C8 41 00 00 00 00 00 00 00 00 E0 00 0E 01 0B 01
```

Virlock As A Polymorphic Malware

Reversing Stages



MZ header

decoded bytes
0x0250

main function

on-demand poly

on-demand poly

on-demand poly

on-demand poly

.text

metamorphic algorithm
0x06C77

.rsrc
0x01200

E F G H

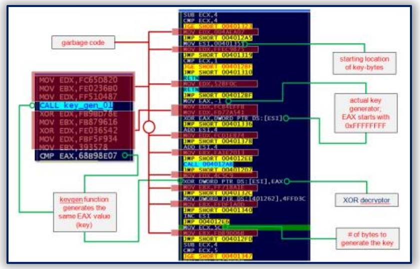
When an on-demand polymorphic algorithm runs, it decrypts the malicious code and executes them. Then re-encrypts itself and the malicious code with a different key.

On-Demand Polymorphic Algorithm

On-Demand Polymorphic Algorithm

Implementation

- Uses Decryptor to decrypt a block of code using an old key
- Executes the newly decrypted code
- Uses RDTSC (Read Time-Stamp Counter) to generate a new dword value
- Uses NewKeyGenerator to generate new key
- Uses Encryptor to encrypt the same block of code using the new key



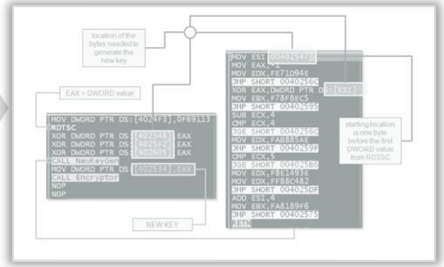
```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB98D78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,004AEA07
JMP SHORT 004012A5
MOV ESI,0040135
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 004012BF
JMP SHORT 00401310
RETN
MOV EDX,528FDC
RETN
JMP SHORT 0040128F
MOV EAX,-1
MOV EDX,FCB4EFFB
MOV EDX,FD72A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FCD1E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2013
JMP SHORT 004012EE
CALL 004012AB
JMP SHORT 00401267
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDE1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
MOV EBX,FD89006B
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

newly decrypted code

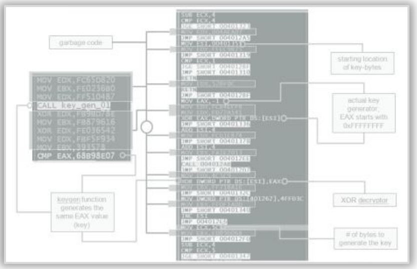
RDTSC



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[402562],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

location of the bytes needed to generate the new key
EAX = DWORD value
starting location is one byte before the first DWORD value from RDTSC
NEW KEY



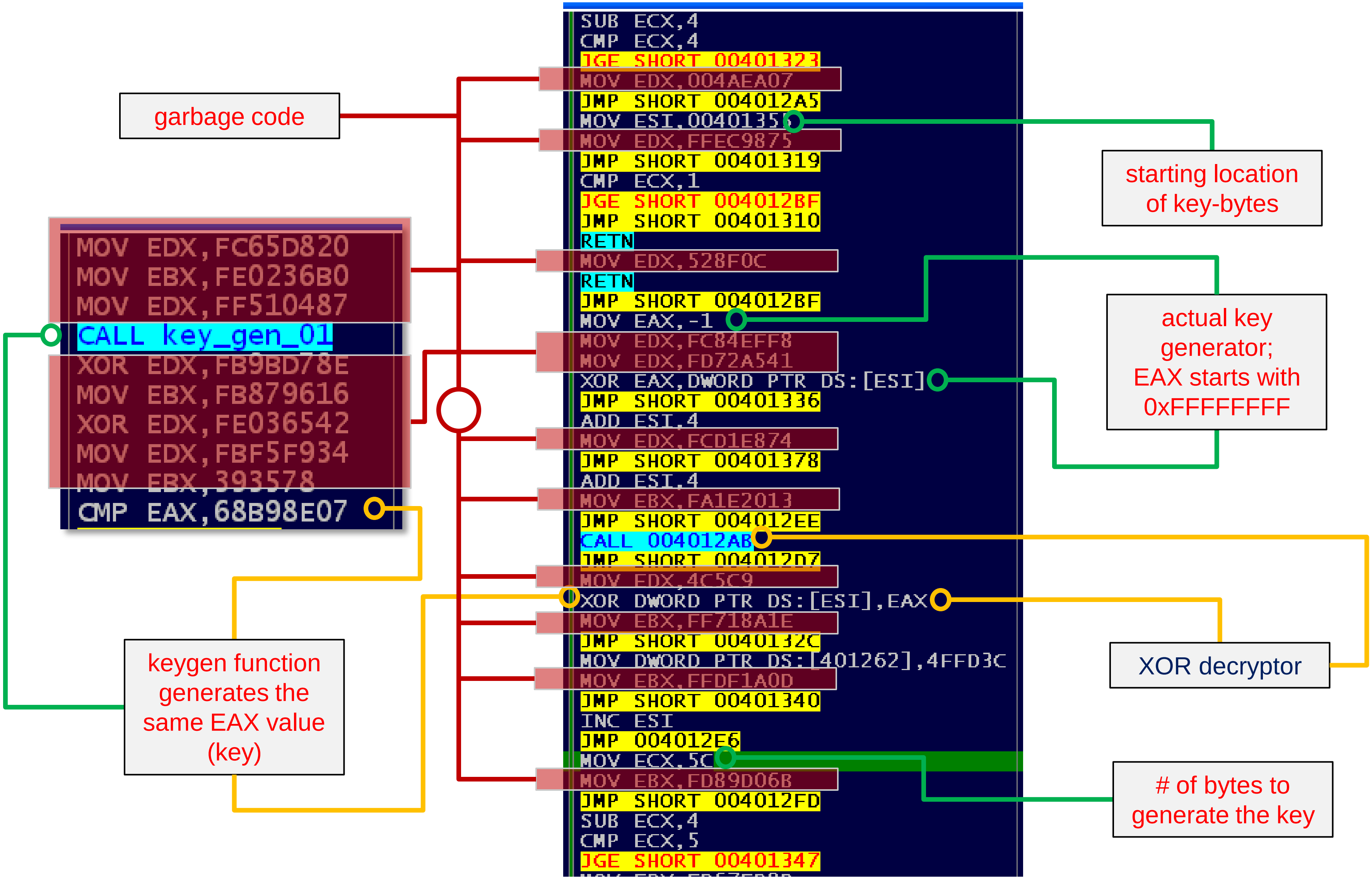
garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

Decryptor

Features:

- Uses **garbage code**
- Keygen function for **redundancy** check
- Uses XOR to generate the key
- Uses XOR to decrypt a block of code

Decryptor



```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,004AEA07
JMP SHORT 004012A5
MOV ESI,00401356
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 004012BF
JMP SHORT 00401310
RETN
MOV EDX,528F0C
RETN
JMP SHORT 004012BF
MOV EAX,-1
MOV EDX,FC84EFF8
MOV EDX,FD72A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FCD1E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2013
JMP SHORT 004012EE
CALL 004012AB
JMP SHORT 004012D7
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDF1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
MOV EBX,FD89D06B
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code

starting location of key-bytes

actual key generator; EAX starts with 0xFFFFFFFF

keygen function generates the same EAX value (key)

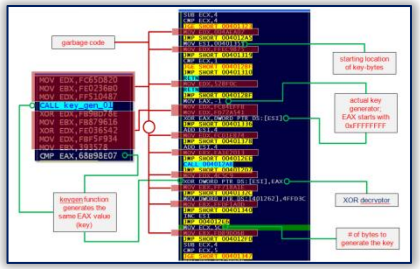
XOR decryptor

# of bytes to generate the key

On-Demand Polymorphic Algorithm

Implementation

- Uses Decryptor to decrypt a block of code using an old key
- Executes the newly decrypted code
- Uses RDTSC (Read Time-Stamp Counter) to generate a new dword value
- Uses NewKeyGenerator to generate new key
- Uses Encryptor to encrypt the same block of code using the new key

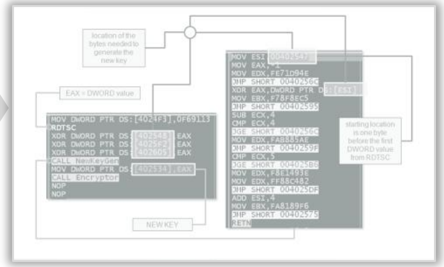

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,884AEA07
JMP SHORT 004012A5
MOV ESI,00401355
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 004012BF
JMP SHORT 00401310
RETN
MOV EDX,528FDC
RETN
JMP SHORT 0040128F
MOV EAX,-1
MOV EDX,FC84EFFB
MOV EDX,FD22A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FCD1E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2011
JMP SHORT 004012EE
CALL 004012AB
JMP SHORT 00401207
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDF1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
MOV EBX,FDB90068
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

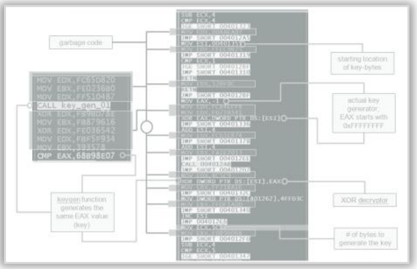
newly decrypted code

RDTSC


location of the bytes needed to generate the new key
EAX = DWORD value
NEW KEY
starting location is one byte before the first DWORD value from RDTSC

```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[402562],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

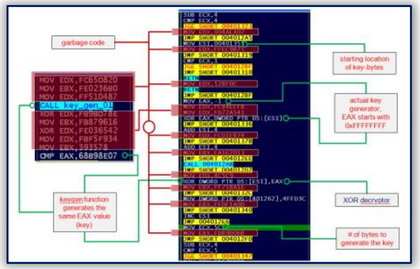
```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```


garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

On-Demand Polymorphic Algorithm

Implementation

- Uses Decryptor to decrypt a block of code using an old key
- Executes the newly decrypted code
- Uses RDTSC (Read Time-Stamp Counter) to generate a new dword value
- Uses NewKeyGenerator to generate new key
- Uses Encryptor to encrypt the same block of code using the new key



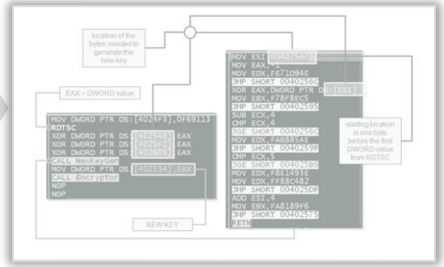
```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,004AEA07
JMP SHORT 004012A5
MOV ESI,0040135
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 004012BF
JMP SHORT 00401310
RETN
MOV EDX,528FDC
RETN
JMP SHORT 0040128F
MOV EAX,-1
MOV EDX,FC84EFFB
MOV EDX,FD22A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FC01E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2011
JMP SHORT 004012EE
CALL 004012A8
JMP SHORT 00401207
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDE1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
MOV EBX,FD890068
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

newly decrypted code

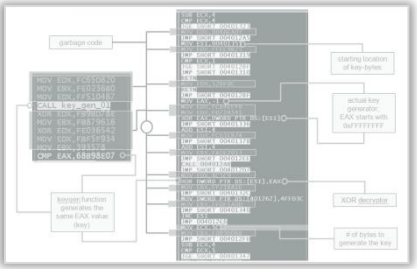
RDTSC



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[402582],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

location of the bytes needed to generate the new key
EAX = DWORD value
NEW KEY
starting location is one byte before the first DWORD value from RDTSC



```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

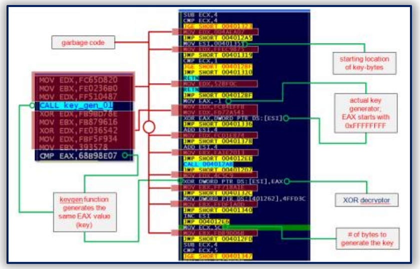
```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,004AEA07
JMP SHORT 004012A5
MOV ESI,0040135
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 004012BF
JMP SHORT 00401310
RETN
MOV EDX,528FDC
RETN
JMP SHORT 0040128F
MOV EAX,-1
MOV EDX,FC84EFFB
MOV EDX,FD22A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FC01E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2011
JMP SHORT 004012EE
CALL 004012A8
JMP SHORT 00401207
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDE1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
MOV EBX,FD890068
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

On-Demand Polymorphic Algorithm

Implementation

- Uses Decryptor to decrypt a block of code using an old key
- Executes the newly decrypted code
- Uses RDTSC (Read Time-Stamp Counter) to generate a new dword value
- Uses NewKeyGenerator to generate new key
- Uses Encryptor to encrypt the same block of code using the new key



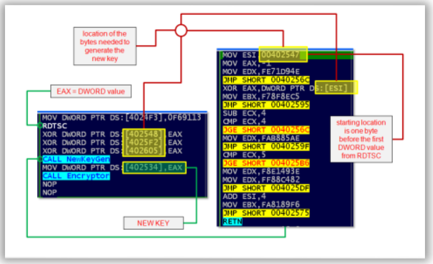
```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,004AEA07
JMP SHORT 004012A5
MOV ESI,0040135
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 0040128F
JMP SHORT 00401310
RETN
MOV EDX,528FDC
RETN
JMP SHORT 0040128F
MOV EAX,-1
MOV EDX,FCB4EFFB
MOV EDX,FD72A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FCD1E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2013
JMP SHORT 004012EE
CALL 004012AB
JMP SHORT 00401207
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDF1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E5
MOV ECX,5C
MOV EBX,FD89D06B
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

newly decrypted code

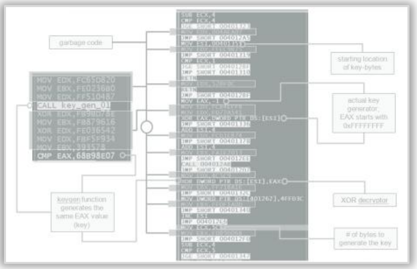
RDTSC



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[4025F2],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

location of the bytes needed to generate the new key
EAX = DWORD value
NEW KEY
starting location is one byte before the first DWORD value from RDTSC

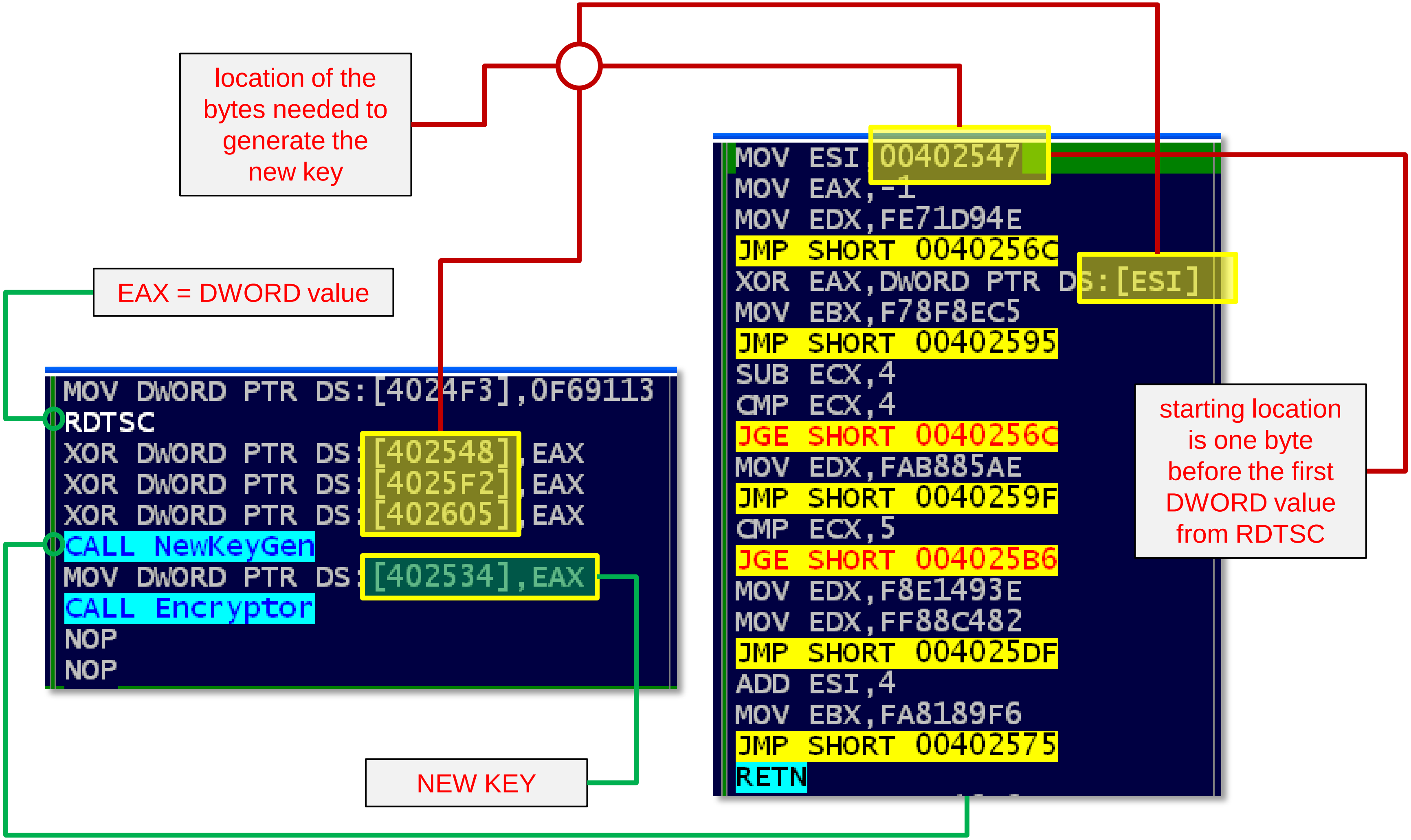


NewKeyGenerator

Implementation:

- RDTSC generates a new dword value
- Saves it in different memory locations
- The memory locations are within the memory range that contains key bytes
- Generates new key by XORing the key bytes
- Saves the new key to the original location of the old key used in the Decryptor

NewKeyGenerator



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[4025F2],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

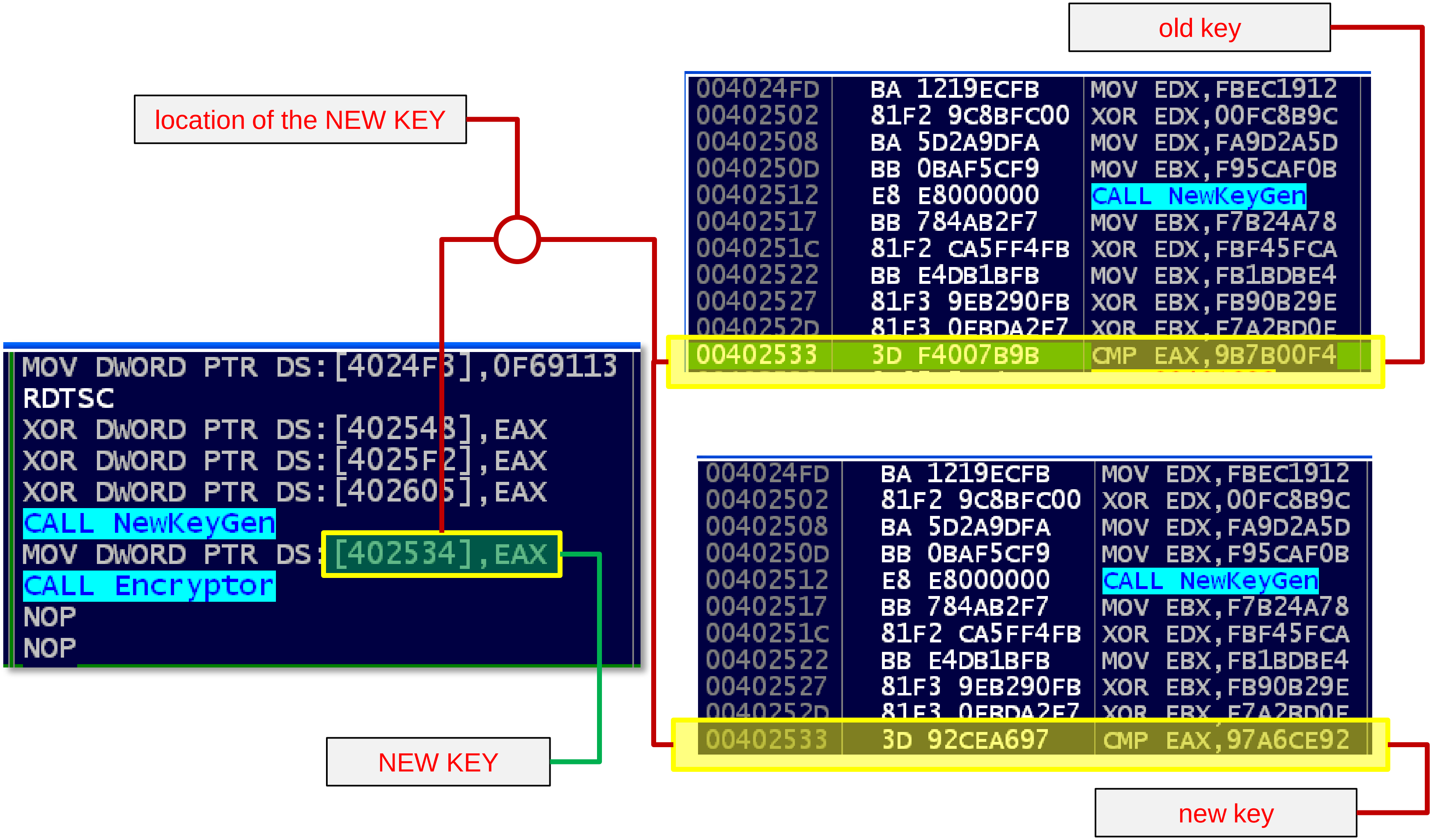
location of the bytes needed to generate the new key

EAX = DWORD value

starting location is one byte before the first DWORD value from RDTSC

NEW KEY

NewKeyGenerator



location of the NEW KEY

```
MOV  DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR  DWORD PTR DS:[402548],EAX
XOR  DWORD PTR DS:[4025F2],EAX
XOR  DWORD PTR DS:[402605],EAX
CALL  NewKeyGen
MOV  DWORD PTR DS:[402534],EAX
CALL  Encryptor
NOP
NOP
```

NEW KEY

old key

```
004024FD   BA 1219ECFB     MOV  EDX,FBEC1912
00402502   81F2 9C8BFC00   XOR  EDX,00FC8B9C
00402508   BA 5D2A9DFA     MOV  EDX,FA9D2A5D
0040250D   BB 0BAF5CF9     MOV  EBX,F95CAF0B
00402512   E8 E8000000     CALL  NewKeyGen
00402517   BB 784AB2F7     MOV  EBX,F7B24A78
0040251c   81F2 CA5FF4FB   XOR  EDX,FBF45FCA
00402522   BB E4DB1BFB     MOV  EBX,FB1BDBE4
00402527   81F3 9EB290FB   XOR  EBX,FB90B29E
0040252D   81F3 0FBDA2F7   XOR  EBX,F7A2BD0F
00402533   3D F4007B9B     CMP  EAX,9B7B00F4
```

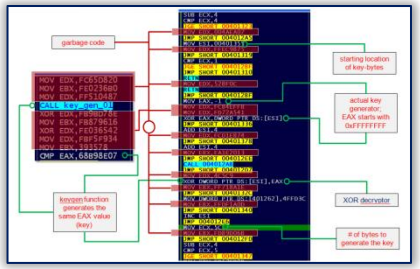
```
004024FD   BA 1219ECFB     MOV  EDX,FBEC1912
00402502   81F2 9C8BFC00   XOR  EDX,00FC8B9C
00402508   BA 5D2A9DFA     MOV  EDX,FA9D2A5D
0040250D   BB 0BAF5CF9     MOV  EBX,F95CAF0B
00402512   E8 E8000000     CALL  NewKeyGen
00402517   BB 784AB2F7     MOV  EBX,F7B24A78
0040251c   81F2 CA5FF4FB   XOR  EDX,FBF45FCA
00402522   BB E4DB1BFB     MOV  EBX,FB1BDBE4
00402527   81F3 9EB290FB   XOR  EBX,FB90B29E
0040252D   81F3 0FBDA2F7   XOR  EBX,F7A2BD0F
00402533   3D 92CEA697     CMP  EAX,97A6CE92
```

new key

On-Demand Polymorphic Algorithm

Implementation

- Uses Decryptor to decrypt a block of code using an old key
- Executes the newly decrypted code
- Uses RDTSC (Read Time-Stamp Counter) to generate a new dword value
- Uses NewKeyGenerator to generate new key
- Uses Encryptor to encrypt the same block of code using the new key



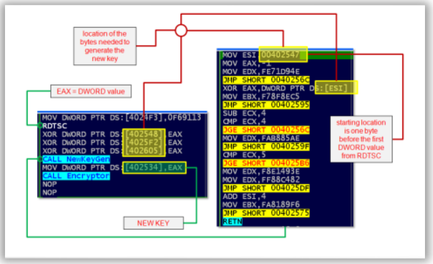
```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
MOV EDX,004AEA07
JMP SHORT 004012A5
MOV ESI,00401351
MOV EDX,FFEC9875
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 004012BF
JMP SHORT 00401310
RETN
MOV EDX,528FDC
RETN
JMP SHORT 0040128F
MOV EAX,-1
MOV EDX,FCB4EFFB
MOV EDX,FD72A541
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
MOV EDX,FCD1E874
JMP SHORT 00401378
ADD ESI,4
MOV EBX,FA1E2013
JMP SHORT 004012EE
CALL 004012A8
JMP SHORT 00401267
MOV EDX,4C5C9
XOR DWORD PTR DS:[ESI],EAX
MOV EBX,FF718A1E
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
MOV EBX,FFDE1A0D
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
MOV EBX,FDB9D06B
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
keygen function generates the same EAX value (key)
XOR decryptor
# of bytes to generate the key

newly decrypted code

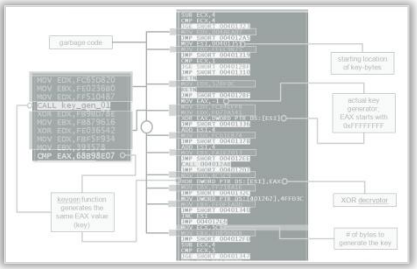
RDTSC



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[4025F2],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

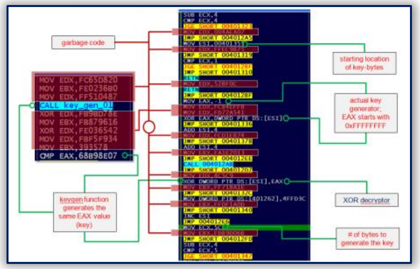
location of the bytes needed to generate the new key
EAX = DWORD value
NEW KEY
starting location is one byte before the first DWORD value from RDTSC



On-Demand Polymorphic Algorithm

Implementation

- Uses Decryptor to decrypt a block of code using an old key
- Executes the newly decrypted code
- Uses RDTSC (Read Time-Stamp Counter) to generate a new dword value
- Uses NewKeyGenerator to generate new key
- Uses Encryptor to encrypt the same block of code using the new key



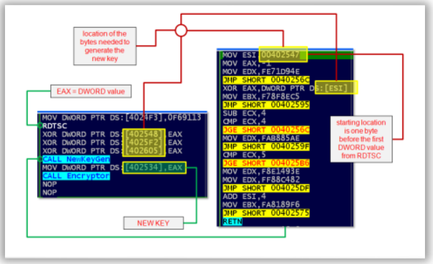
```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

```
SUB ECX,4
CMP ECX,4
JGE SHORT 00401323
[illegible]
JMP SHORT 004012A5
MOV ESI,0040135
[illegible]
JMP SHORT 00401319
CMP ECX,1
JGE SHORT 0040128F
JMP SHORT 00401310
RETN
[illegible]
RETN
JMP SHORT 0040128F
MOV EAX,-1
[illegible]
[illegible]
XOR EAX,DWORD PTR DS:[ESI]
JMP SHORT 00401336
ADD ESI,4
[illegible]
JMP SHORT 00401378
ADD ESI,4
[illegible]
JMP SHORT 004012EE
CALL 004012A8
JMP SHORT 00401207
[illegible]
XOR DWORD PTR DS:[ESI],EAX
[illegible]
JMP SHORT 0040132C
MOV DWORD PTR DS:[401262],4FFD3C
[illegible]
JMP SHORT 00401340
INC ESI
JMP 004012E6
MOV ECX,5C
[illegible]
JMP SHORT 004012FD
SUB ECX,4
CMP ECX,5
JGE SHORT 00401347
```

garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
XOR decryptor
keygen function generates the same EAX value (key)
# of bytes to generate the key

newly decrypted code

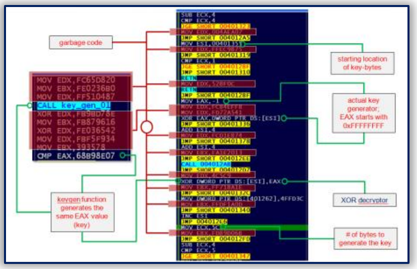
RDTSC



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[4025F2],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

location of the bytes needed to generate the new key
EAX = DWORD value
NEW KEY
starting location is one byte before the first DWORD value from RDTSC



```
MOV EDX,FC65D820
MOV EBX,FE0236B0
MOV EDX,FF510487
CALL key_gen_01
XOR EDX,FB9BD78E
MOV EBX,FB879616
XOR EDX,FE036542
MOV EDX,FBF5F934
MOV EBX,393578
CMP EAX,68B98E07
```

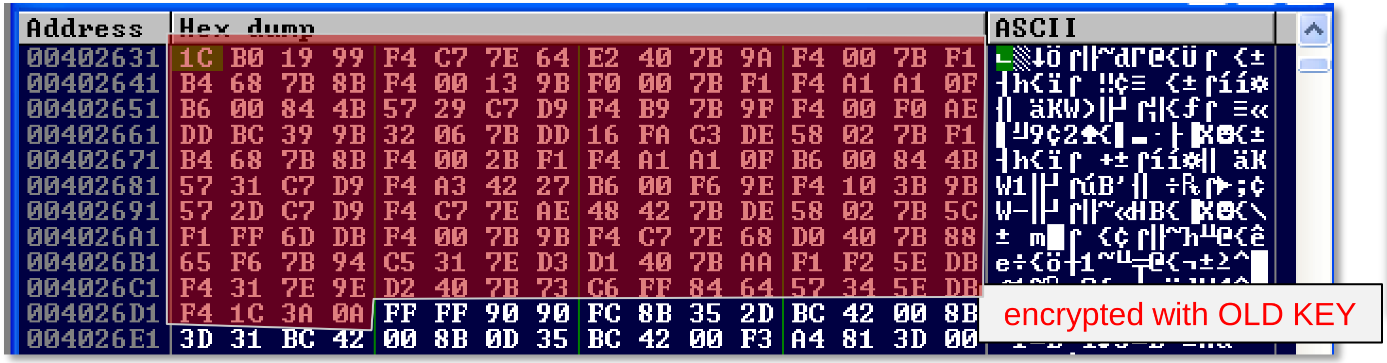
garbage code
starting location of key-bytes
actual key generator; EAX starts with 0xFFFFFFFF
XOR decryptor
keygen function generates the same EAX value (key)
# of bytes to generate the key

Encryptor

Features:

- Uses the same algorithm as the Decryptor
- Uses the new key to encrypt the same block of code

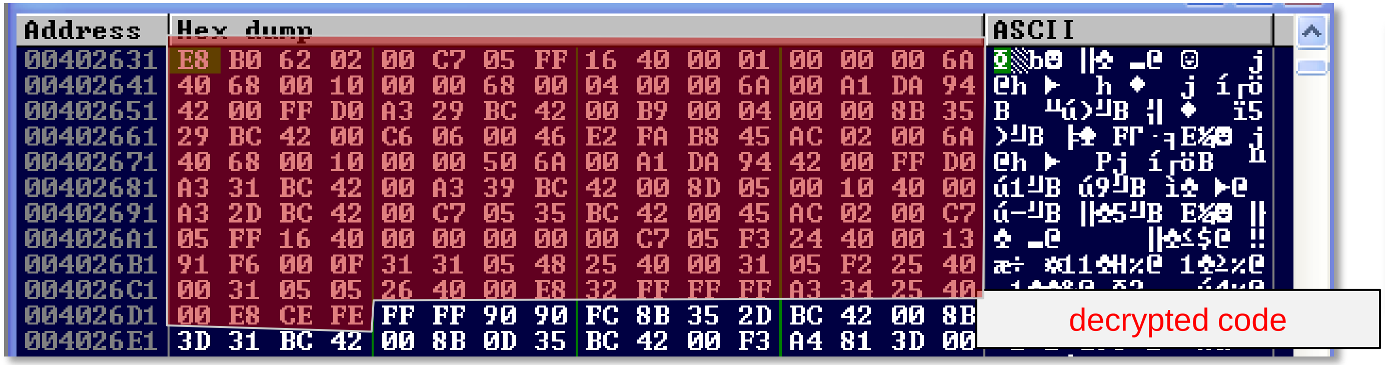
Sample On-demand Polymorphic Values



```
Address   Hex dump                                         ASCII
00402631  1C B0 19 99 F4 C7 7E 64 E2 40 7B 9A F4 00 7B F1
00402641  B4 68 7B 8B F4 00 13 9B F0 00 7B F1 F4 A1 A1 0F
00402651  B6 00 84 4B 57 29 C7 D9 F4 B9 7B 9F F4 00 F0 AE
00402661  DD BC 39 9B 32 06 7B DD 16 FA C3 DE 58 02 7B F1
00402671  B4 68 7B 8B F4 00 2B F1 F4 A1 A1 0F B6 00 84 4B
00402681  57 31 C7 D9 F4 A3 42 27 B6 00 F6 9E F4 10 3B 9B
00402691  57 2D C7 D9 F4 C7 7E AE 48 42 7B DE 58 02 7B 5C
004026A1  F1 FF 6D DB F4 00 7B 9B F4 C7 7E 68 D0 40 7B 88
004026B1  65 F6 7B 94 C5 31 7E D3 D1 40 7B AA F1 F2 5E DB
004026C1  F4 31 7E 9E D2 40 7B 73 C6 FF 84 64 57 34 5E DB
004026D1  F4 1C 3A 0A FF FF 90 90 FC 8B 35 2D BC 42 00 8B
004026E1  3D 31 BC 42 00 8B 0D 35 BC 42 00 F3 A4 81 3D 00
```

encrypted with OLD KEY

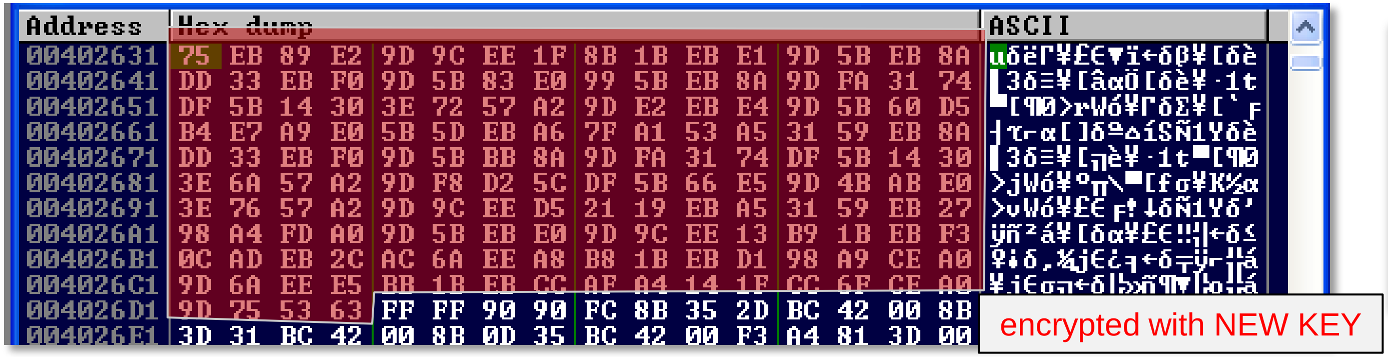
```
1C B0 19 99
F4 C7 7E 64
E2 40 7B 9A
F4 00 7B F1
```



```
Address   Hex dump                                         ASCII
00402631  E8 B0 62 02 00 C7 05 FF 16 40 00 01 00 00 00 6A
00402641  40 68 00 10 00 00 68 00 04 00 00 6A 00 A1 DA 94
00402651  42 00 FF D0 A3 29 BC 42 00 B9 00 04 00 00 8B 35
00402661  29 BC 42 00 C6 06 00 46 E2 FA B8 45 AC 02 00 6A
00402671  40 68 00 10 00 00 50 6A 00 A1 DA 94 42 00 FF D0
00402681  A3 31 BC 42 00 A3 39 BC 42 00 8D 05 00 10 40 00
00402691  A3 2D BC 42 00 C7 05 35 BC 42 00 45 AC 02 00 C7
004026A1  05 FF 16 40 00 00 00 00 00 C7 05 F3 24 40 00 13
004026B1  91 F6 00 0F 31 31 05 48 25 40 00 31 05 F2 25 40
004026C1  00 31 05 05 26 40 00 E8 32 FF FF FF A3 34 25 40
004026D1  00 E8 CE FE FF FF 90 90 FC 8B 35 2D BC 42 00 8B
004026E1  3D 31 BC 42 00 8B 0D 35 BC 42 00 F3 A4 81 3D 00
```

decrypted code

```
E8 B0 62 02
00 C7 05 FF
16 40 00 01
00 00 00 6A
```

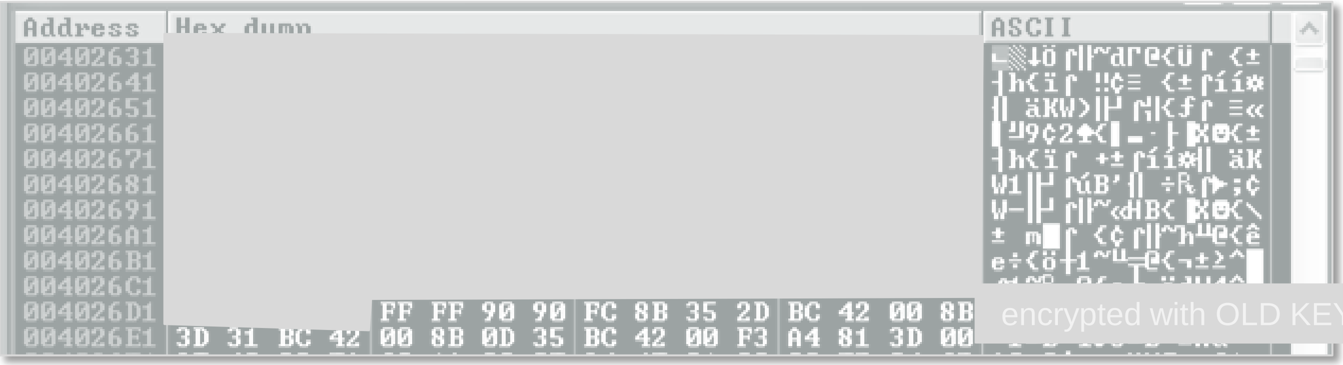


```
Address   Hex dump                                         ASCII
00402631  75 EB 89 E2 9D 9C EE 1F 8B 1B EB E1 9D 5B EB 8A
00402641  DD 33 EB F0 9D 5B 83 E0 99 5B EB 8A 9D FA 31 74
00402651  DF 5B 14 30 3E 72 57 A2 9D E2 EB E4 9D 5B 60 D5
00402661  B4 E7 A9 E0 5B 5D EB A6 7F A1 53 A5 31 59 EB 8A
00402671  DD 33 EB F0 9D 5B BB 8A 9D FA 31 74 DF 5B 14 30
00402681  3E 6A 57 A2 9D F8 D2 5C DF 5B 66 E5 9D 4B AB E0
00402691  3E 76 57 A2 9D 9C EE D5 21 19 EB A5 31 59 EB 27
004026A1  98 A4 FD A0 9D 5B EB E0 9D 9C EE 13 B9 1B EB F3
004026B1  0C AD EB 2C AC 6A EE A8 B8 1B EB D1 98 A9 CE A0
004026C1  9D 6A EE E5 BB 1B EB CC AF A4 14 1F CC 6F CE A0
004026D1  9D 75 53 63 FF FF 90 90 FC 8B 35 2D BC 42 00 8B
004026E1  3D 31 BC 42 00 8B 0D 35 BC 42 00 F3 A4 81 3D 00
```

encrypted with NEW KEY

```
75 EB 89 E2
9D 9C EE 1F
8B 1B EB E1
9D 5B EB 8A
```

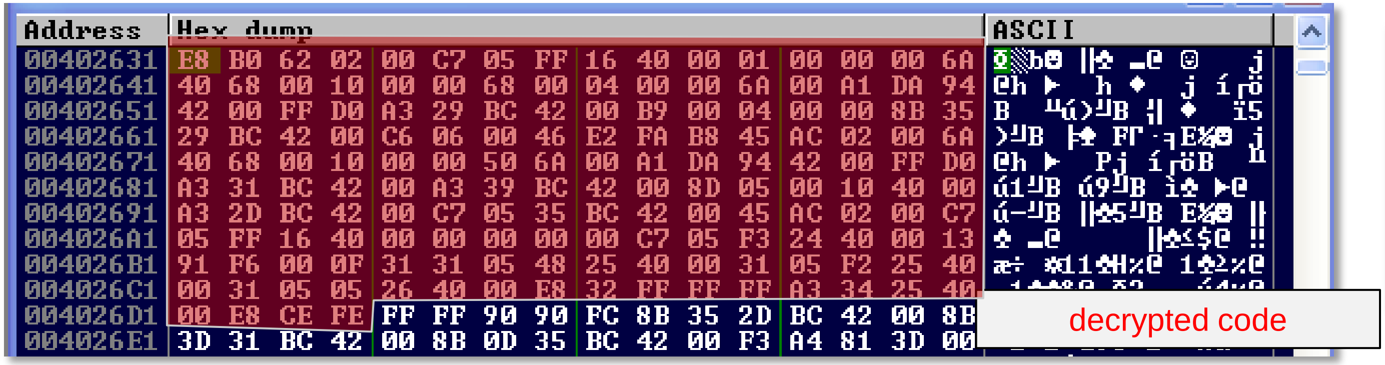
Detection



```
Address   Hex dump                                          ASCII
00402631
00402641
00402651
00402661
00402671
00402681
00402691
004026A1
004026B1
004026C1
004026D1              FF FF 90 90 FC 8B 35 2D BC 42 00 8B
004026E1  3D 31 BC 42 00 8B 0D 35 BC 42 00 F3 A4 81 3D 00
```

encrypted with OLD KEY

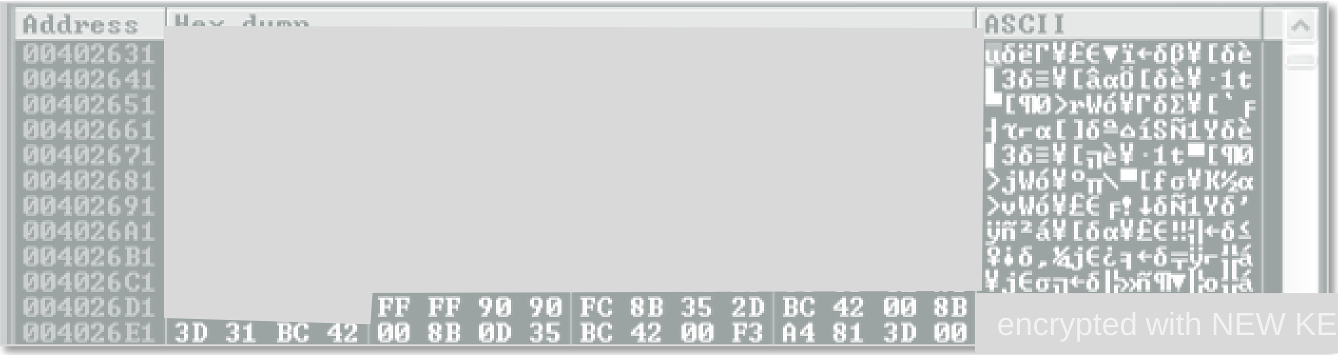
```
1C B0 19 99
F4 C7 7E 64
E2 40 7B 9A
F4 00 7B F1
```



```
Address   Hex dump                                          ASCII
00402631  E8 B0 62 02 00 C7 05 FF 16 40 00 01 00 00 00 6A
00402641  40 68 00 10 00 00 68 00 04 00 00 6A 00 A1 DA 94
00402651  42 00 FF D0 A3 29 BC 42 00 B9 00 04 00 00 8B 35
00402661  29 BC 42 00 C6 06 00 46 E2 FA B8 45 AC 02 00 6A
00402671  40 68 00 10 00 00 50 6A 00 A1 DA 94 42 00 FF D0
00402681  A3 31 BC 42 00 A3 39 BC 42 00 8D 05 00 10 40 00
00402691  A3 2D BC 42 00 C7 05 35 BC 42 00 45 AC 02 00 C7
004026A1  05 FF 16 40 00 00 00 00 00 C7 05 F3 24 40 00 13
004026B1  91 F6 00 0F 31 31 05 48 25 40 00 31 05 F2 25 40
004026C1  00 31 05 05 26 40 00 E8 32 FF FF FF A3 34 25 40
004026D1  00 E8 CE FE FF FF 90 90 FC 8B 35 2D BC 42 00 8B
004026E1  3D 31 BC 42 00 8B 0D 35 BC 42 00 F3 A4 81 3D 00
```

decrypted code

```
E8 B0 62 02
00 C7 05 FF
16 40 00 01
00 00 00 6A
```



```
Address   Hex dump                                          ASCII
00402631
00402641
00402651
00402661
00402671
00402681
00402691
004026A1
004026B1
004026C1
004026D1              FF FF 90 90 FC 8B 35 2D BC 42 00 8B
004026E1  3D 31 BC 42 00 8B 0D 35 BC 42 00 F3 A4 81 3D 00
```

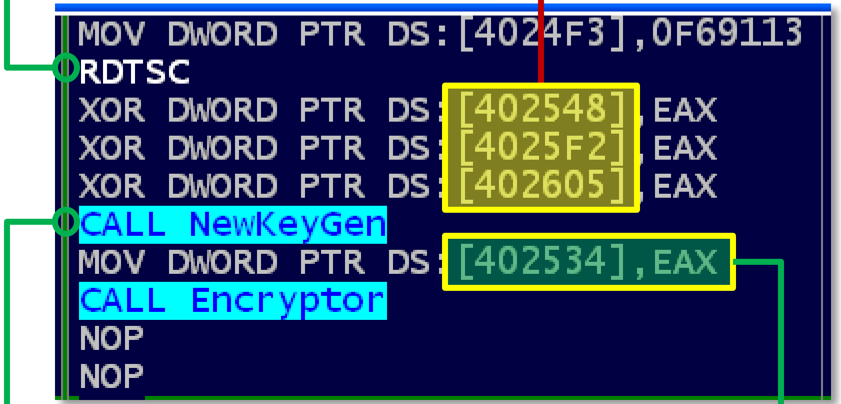
encrypted with NEW KEY

```
75 EB 89 E2
9D 9C EE 1F
8B 1B EB E1
9D 5B EB 8A
```

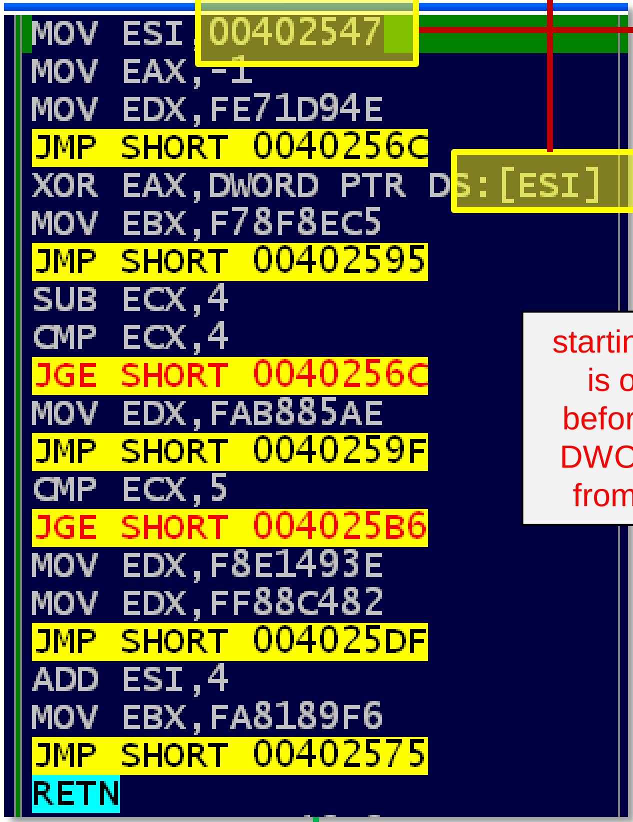
Detection

location of the bytes needed to generate the new key

EAX = DWORD value



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[4025F2],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

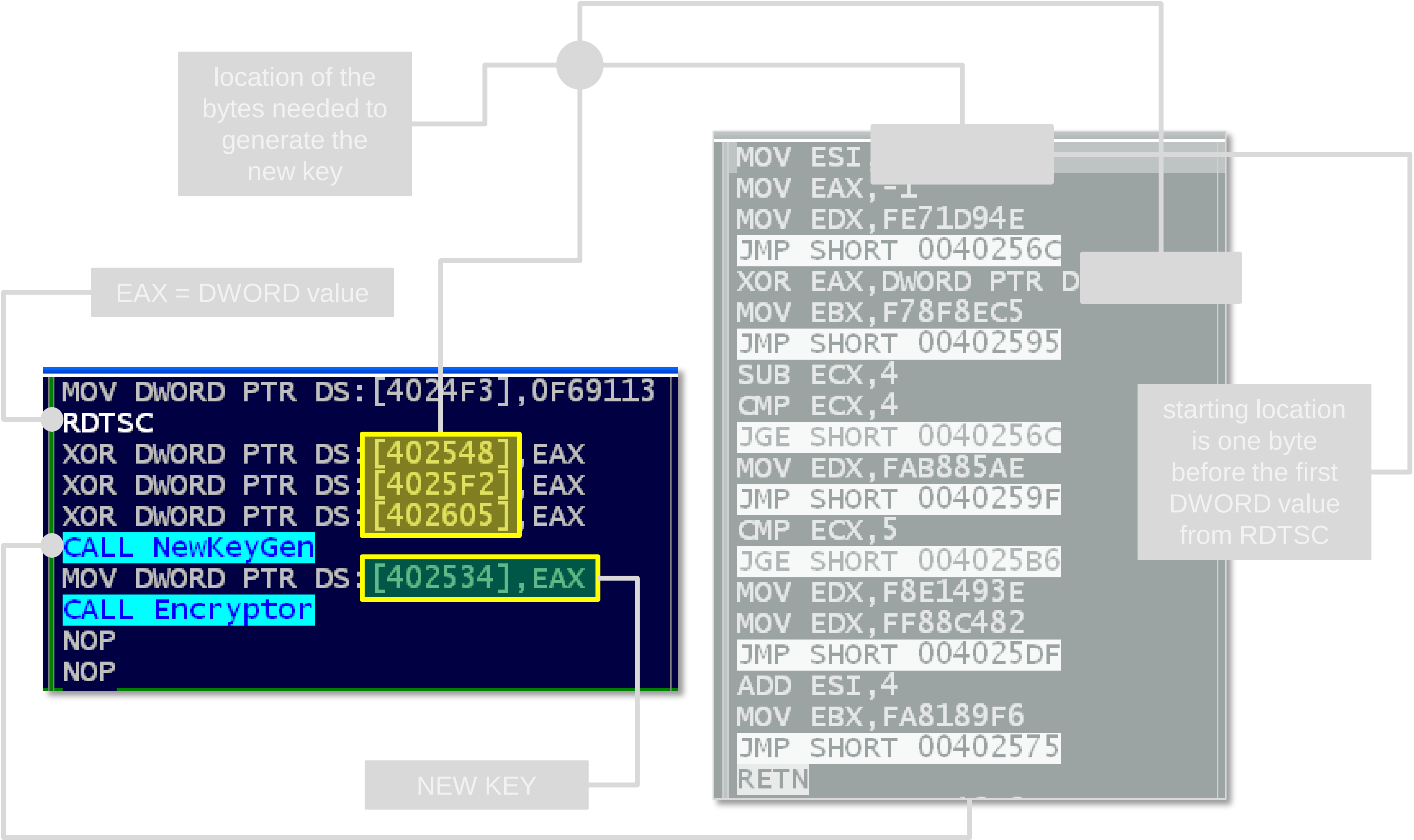


```
MOV ESI,00402547
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR DS:[ESI]
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

starting location is one byte before the first DWORD value from RDTSC

NEW KEY

Detection



```
MOV DWORD PTR DS:[4024F3],0F69113
RDTSC
XOR DWORD PTR DS:[402548],EAX
XOR DWORD PTR DS:[4025F2],EAX
XOR DWORD PTR DS:[402605],EAX
CALL NewKeyGen
MOV DWORD PTR DS:[402534],EAX
CALL Encryptor
NOP
NOP
```

```
MOV ESI,
MOV EAX,-1
MOV EDX,FE71D94E
JMP SHORT 0040256C
XOR EAX,DWORD PTR D
MOV EBX,F78F8EC5
JMP SHORT 00402595
SUB ECX,4
CMP ECX,4
JGE SHORT 0040256C
MOV EDX,FAB885AE
JMP SHORT 0040259F
CMP ECX,5
JGE SHORT 004025B6
MOV EDX,F8E1493E
MOV EDX,FF88C482
JMP SHORT 004025DF
ADD ESI,4
MOV EBX,FA8189F6
JMP SHORT 00402575
RETN
```

location of the bytes needed to generate the new key

EAX = DWORD value

starting location is one byte before the first DWORD value from RDTSC

NEW KEY

Virlock As A Metamorphic Malware

Metamorphic Algorithm

Basics:

Putting a value(0) in a register(EAX)

MOV EAX,0	EAX register gets 0 directly
XOR EAX,EAX	XORing the same register by itself also generates a zero value placed into a given register
SUB EAX,EAX	SUBtracting any register by itself also generates the same result.
MOV EAX, 0x10 ADD EAX, 0x10 SUB EAX, 0x20	EAX also gets 0

Metamorphic Algorithm

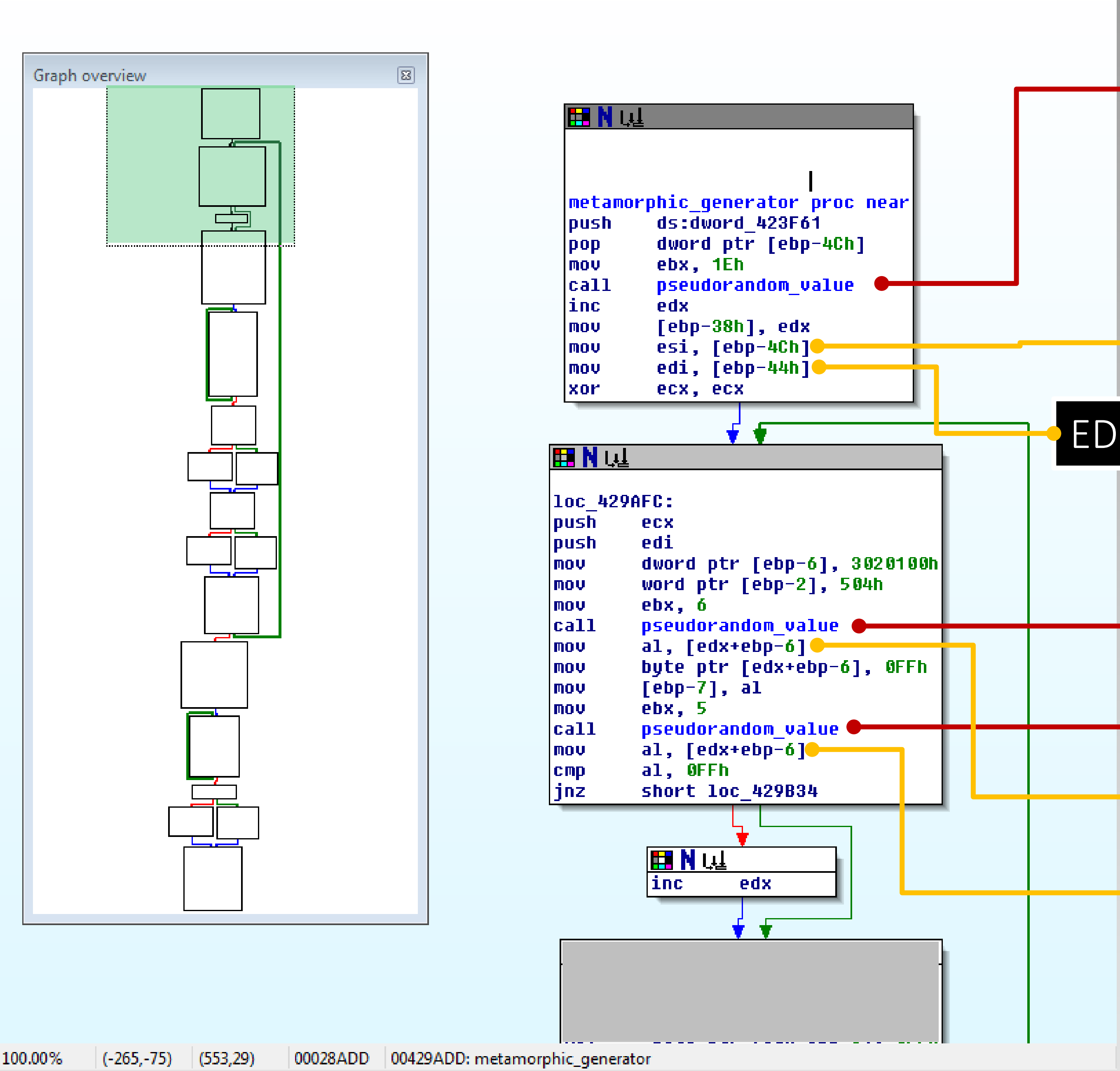
Detection Limitation

- Hard to find similar bytes
- Unknown length of bytes
- Unpredictable code

Metamorphic Engine

Raw Ingredients

- Number of instructions to generate
- Registers used per instruction
- Number of bytes
- Pseudorandom value generator
- Instruction generator 1
- Instruction generator 2
- Length of code to encode



```
metamorphic_generator proc near
push    ds:dword_423F61
pop     dword ptr [ebp-4Ch]
mov     ebx, 1Eh
call    pseudorandom_value
inc     edx
mov     [ebp-38h], edx
mov     esi, [ebp-4Ch]
mov     edi, [ebp-44h]
xor     ecx, ecx
```

```
loc_429AFC:
push    ecx
push    edi
mov     dword ptr [ebp-6], 3020100h
mov     word ptr [ebp-2], 504h
mov     ebx, 6
call    pseudorandom_value
mov     al, [edx+ebp-6]
mov     byte ptr [edx+ebp-6], 0FFh
mov     [ebp-7], al
mov     ebx, 5
call    pseudorandom_value
mov     al, [edx+ebp-6]
cmp     al, 0FFh
jnz     short loc_429B34
```

```
inc     edx
```

```
100.00%    (-265,-75)    (553,29)    00028ADD    00429ADD: metamorphic_generator
```

Pseudorandom value generator

ESI = buffer

EDI = malware buffer

address register

code register

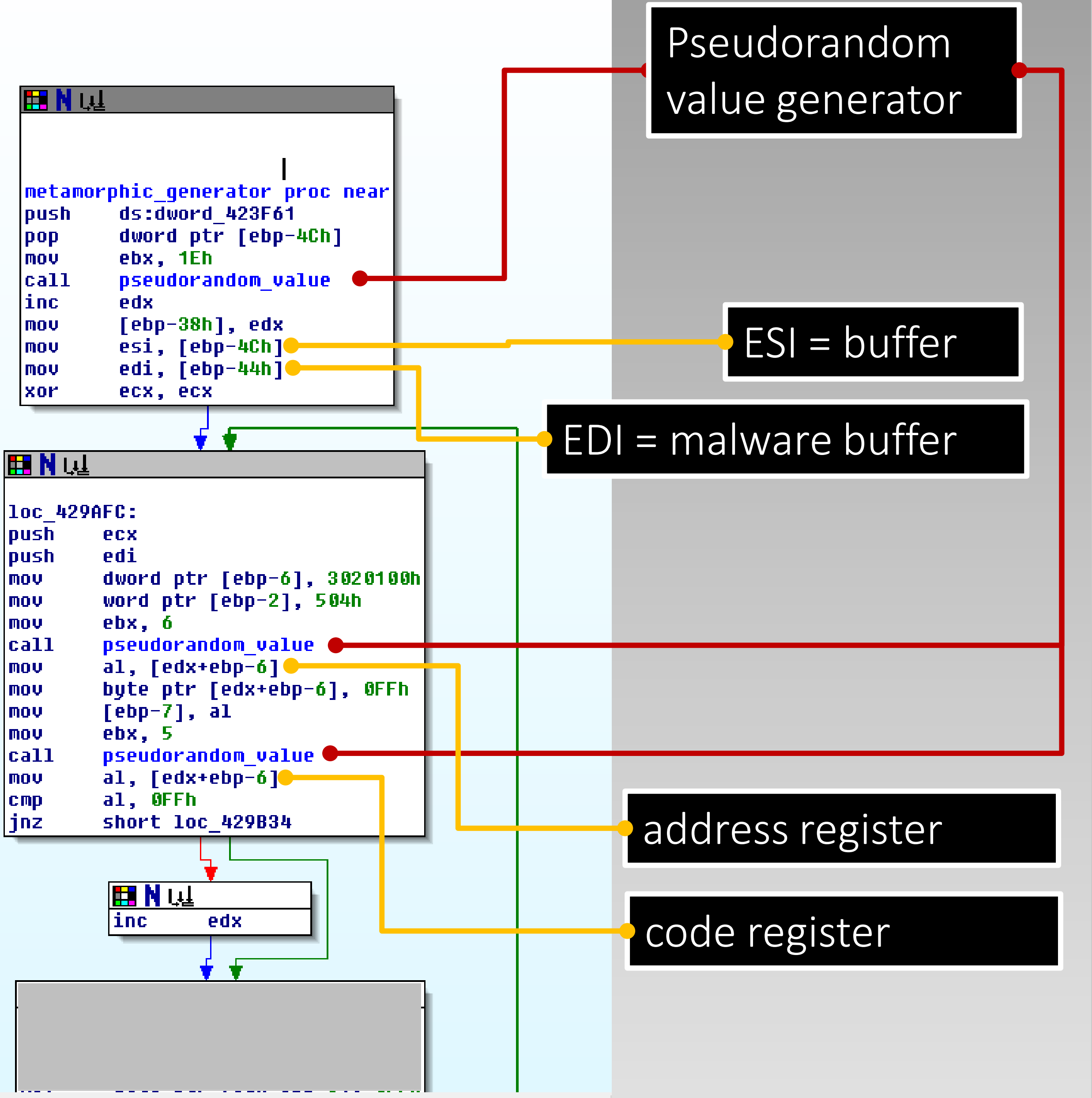
Pseudorandom value generator
-function that generates the randomized value

buffer - temporary memory location that collects the metamorphic code e.g., 0x009c0000

malware buffer – holds the code to be encoded, e.g., 0x01130000

address register(addreg) – randomly selected register that points to the address of the encoded bytes

code register(codereg) – randomly selected register that holds the encoded bytes



```
metamorphic_generator proc near
push    ds:dword_423F61
pop     dword ptr [ebp-4Ch]
mov     ebx, 1Eh
call    pseudorandom_value
inc     edx
mov     [ebp-38h], edx
mov     esi, [ebp-4Ch]
mov     edi, [ebp-44h]
xor     ecx, ecx
```

```
loc_429AFC:
push    ecx
push    edi
mov     dword ptr [ebp-6], 3020100h
mov     word ptr [ebp-2], 504h
mov     ebx, 6
call    pseudorandom_value
mov     al, [edx+ebp-6]
mov     byte ptr [edx+ebp-6], 0FFh
mov     [ebp-7], al
mov     ebx, 5
call    pseudorandom_value
mov     al, [edx+ebp-6]
cmp     al, 0FFh
jnz     short loc_429B34
```

```
inc     edx
```

Pseudorandom value generator

ESI = buffer

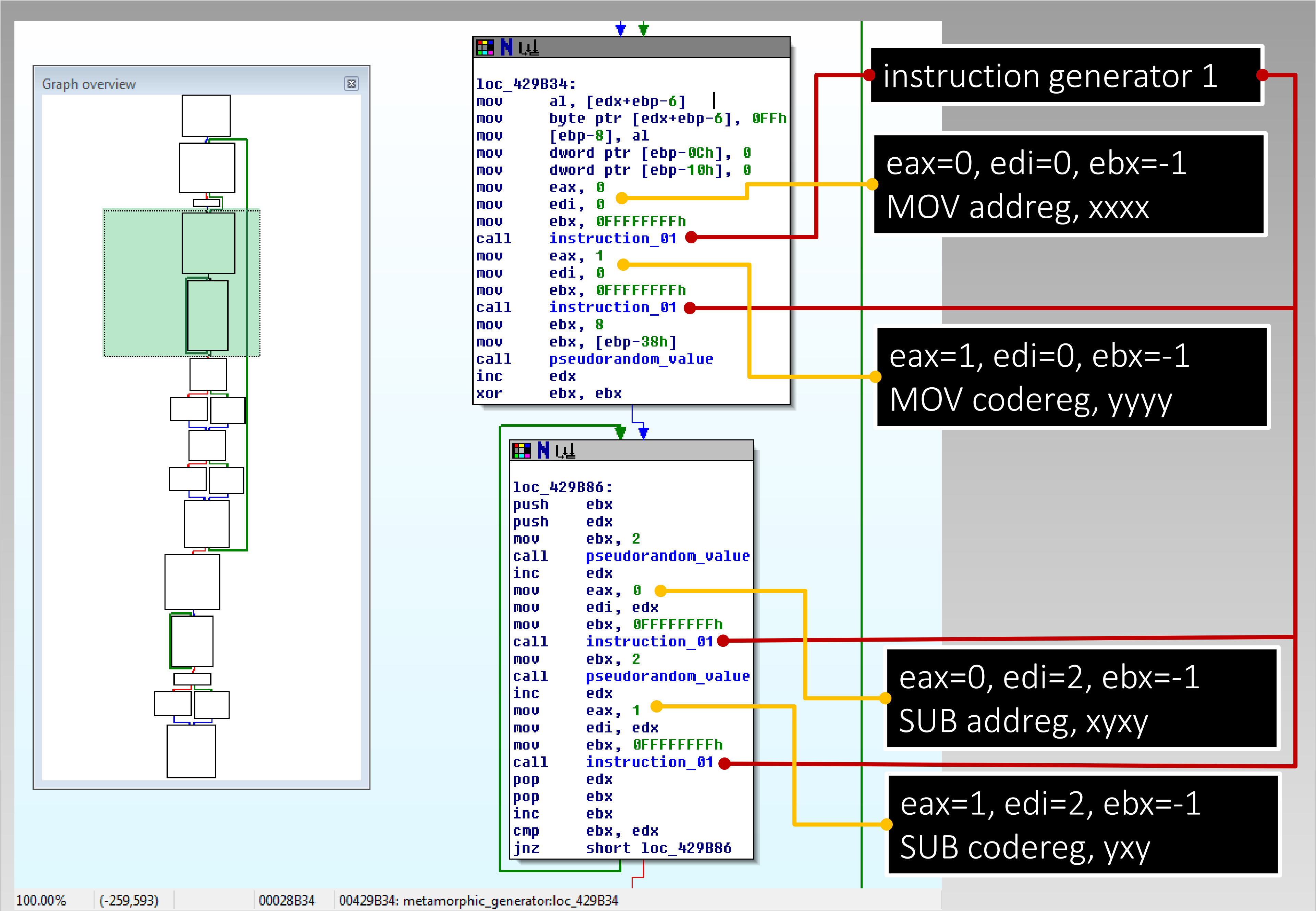
EDI = malware buffer

address register

code register

100.00% (-265,-75) (553,29) 00028ADD 00429ADD: metamorphic_generator

FORTINET



```
loc_429B34:
mov     al, [edx+ebp-6]
mov     byte ptr [edx+ebp-6], 0FFh
mov     [ebp-8], al
mov     dword ptr [ebp-0Ch], 0
mov     dword ptr [ebp-10h], 0
mov     eax, 0
mov     edi, 0
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     eax, 1
mov     edi, 0
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     ebx, 8
mov     ebx, [ebp-38h]
call    pseudorandom_value
inc     edx
xor     ebx, ebx
```

```
loc_429B86:
push    ebx
push    edx
mov     ebx, 2
call    pseudorandom_value
inc     edx
mov     eax, 0
mov     edi, edx
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     ebx, 2
call    pseudorandom_value
inc     edx
mov     eax, 1
mov     edi, edx
mov     ebx, 0FFFFFFFFh
call    instruction_01
pop     edx
pop     ebx
inc     ebx
cmp     ebx, edx
jnz     short loc_429B86
```

```
100.00%    (-259,593)    00028B34    00429B34: metamorphic_generator:loc_429B34
```

instruction generator 1

eax=0, edi=0, ebx=-1
MOV addreg, xxxx

eax=1, edi=0, ebx=-1
MOV codereg, yyyy

eax=0, edi=2, ebx=-1
SUB addreg, xyxy

eax=1, edi=2, ebx=-1
SUB codereg, yxy

instruction generator 1

-function that generates the initial **MOV** instructions for both the **addreg** and **codereg** registers

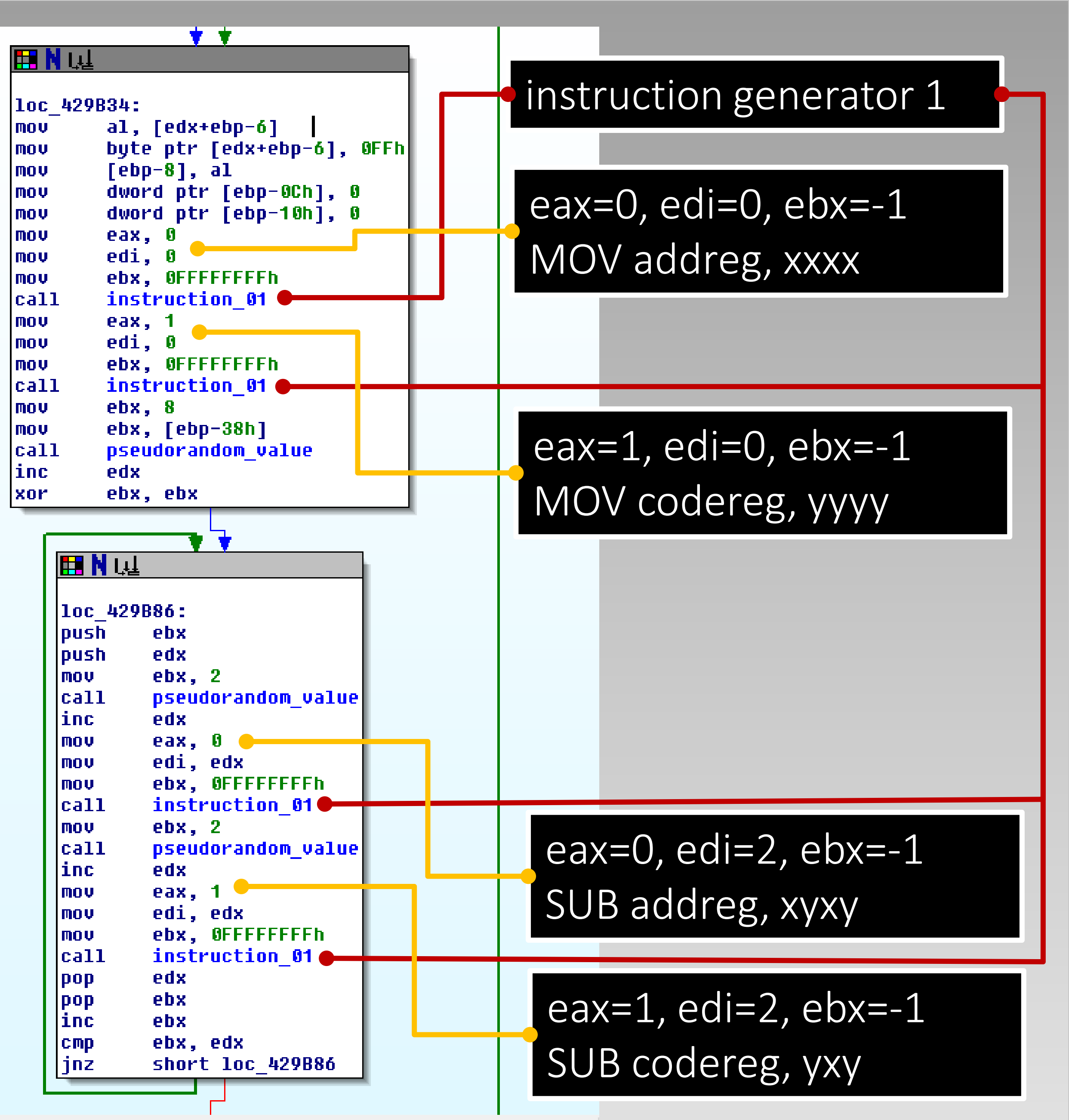
-e.g

MOV ESI, 6D442
MOV EDX, 142A

- it also generates the subsequent **ADD** and **SUB** instructions for the **addreg** and **codereg**

-e.g.,

SUB ESI, 0D8D47
SUB EDX, 6415E
ADD ESI, 1234
ADD EDX, ABCD



```
loc_429B34:
mov     al, [edx+ebp-6]
mov     byte ptr [edx+ebp-6], 0FFh
mov     [ebp-8], al
mov     dword ptr [ebp-0Ch], 0
mov     dword ptr [ebp-10h], 0
mov     eax, 0
mov     edi, 0
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     eax, 1
mov     edi, 0
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     ebx, 8
mov     ebx, [ebp-38h]
call    pseudorandom_value
inc     edx
xor     ebx, ebx
```

```
loc_429B86:
push    ebx
push    edx
mov     ebx, 2
call    pseudorandom_value
inc     edx
mov     eax, 0
mov     edi, edx
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     ebx, 2
call    pseudorandom_value
inc     edx
mov     eax, 1
mov     edi, edx
mov     ebx, 0FFFFFFFFh
call    instruction_01
pop     edx
pop     ebx
inc     ebx
cmp     ebx, edx
jnz     short loc_429B86
```

instruction generator 1

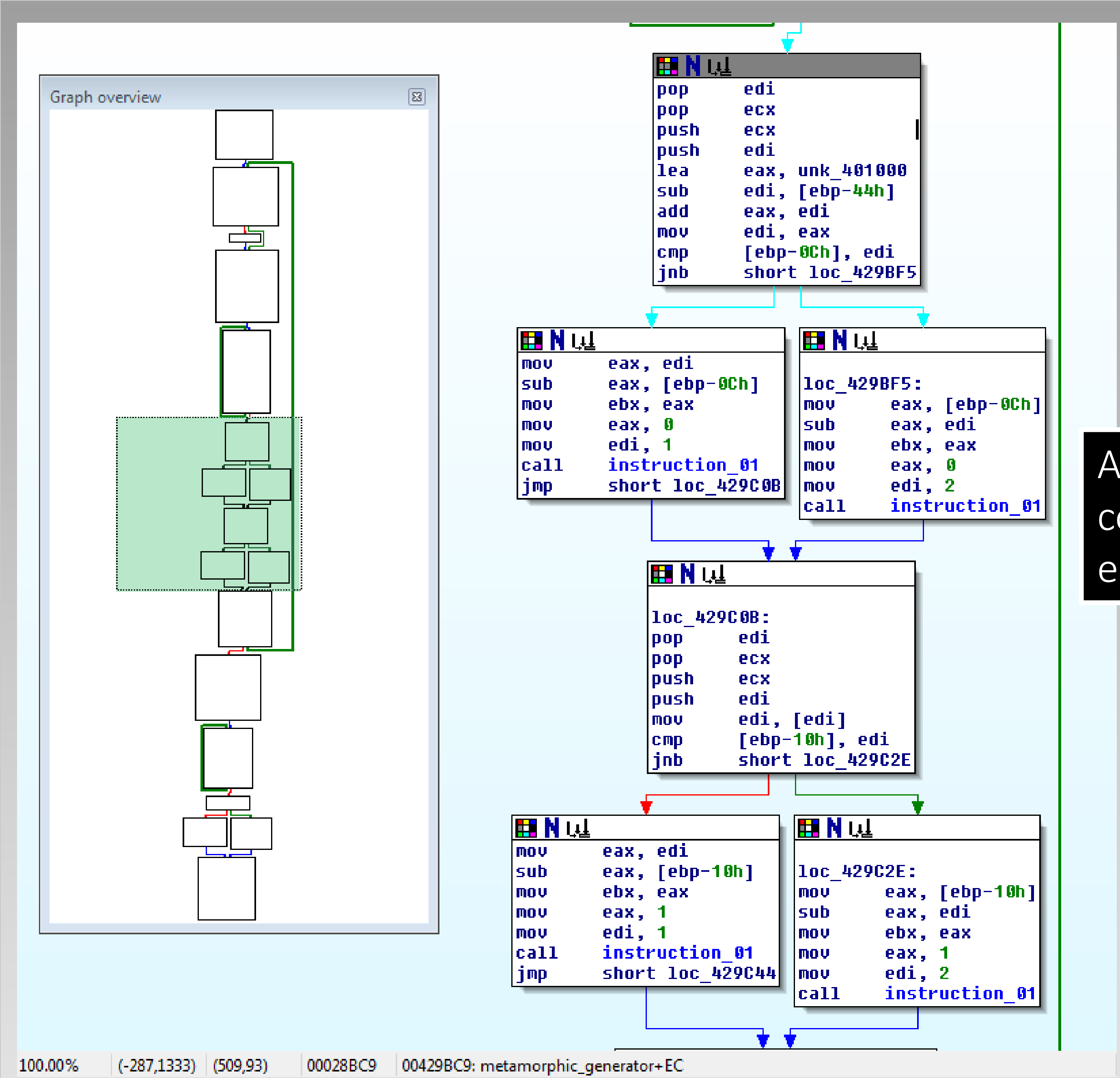
eax=0, edi=0, ebx=-1
MOV addreg, xxxx

eax=1, edi=0, ebx=-1
MOV codereg, yyyy

eax=0, edi=2, ebx=-1
SUB addreg, xyxy

eax=1, edi=2, ebx=-1
SUB codereg, yxy

100.00% (-259,593) 00028B34 00429B34: metamorphic_generator:loc_429B34



```
pop     edi
pop     ecx
push    ecx
push    edi
lea     eax, unk_401000
sub     edi, [ebp-44h]
add     eax, edi
mov     edi, eax
cmp     [ebp-0Ch], edi
jnb     short loc_429BF5
```

```
mov     eax, edi
sub     eax, [ebp-0Ch]
mov     ebx, eax
mov     eax, 0
mov     edi, 1
call    instruction_01
jmp     short loc_429C0B
```

```
loc_429BF5:
mov     eax, [ebp-0Ch]
sub     eax, edi
mov     ebx, eax
mov     eax, 0
mov     edi, 2
call    instruction_01
```

```
loc_429C0B:
pop     edi
pop     ecx
push    ecx
push    edi
mov     edi, [edi]
cmp     [ebp-10h], edi
jnb     short loc_429C2E
```

```
mov     eax, edi
sub     eax, [ebp-10h]
mov     ebx, eax
mov     eax, 1
mov     edi, 1
call    instruction_01
jmp     short loc_429C44
```

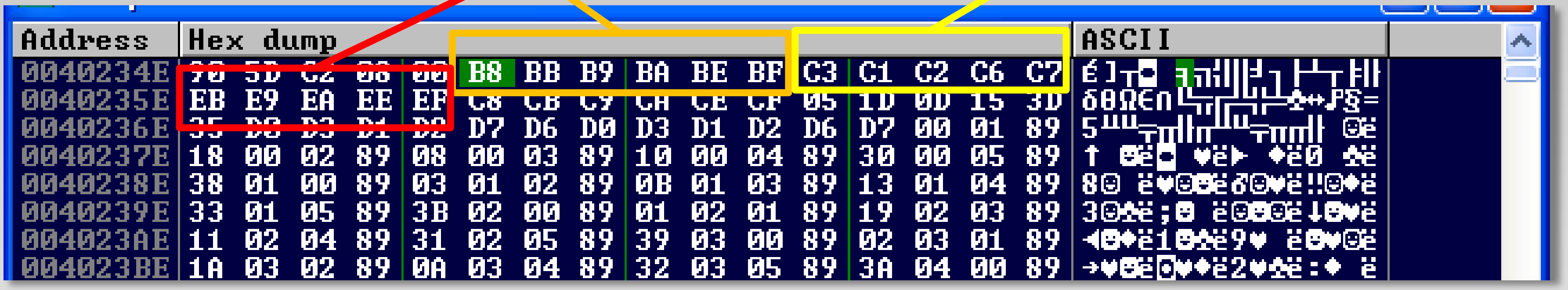
```
loc_429C2E:
mov     eax, [ebp-10h]
sub     eax, edi
mov     ebx, eax
mov     eax, 1
mov     edi, 2
call    instruction_01
```

100.00% (-287,1333) (509,93) 00028BC9 00429BC9: metamorphic_generator+EC

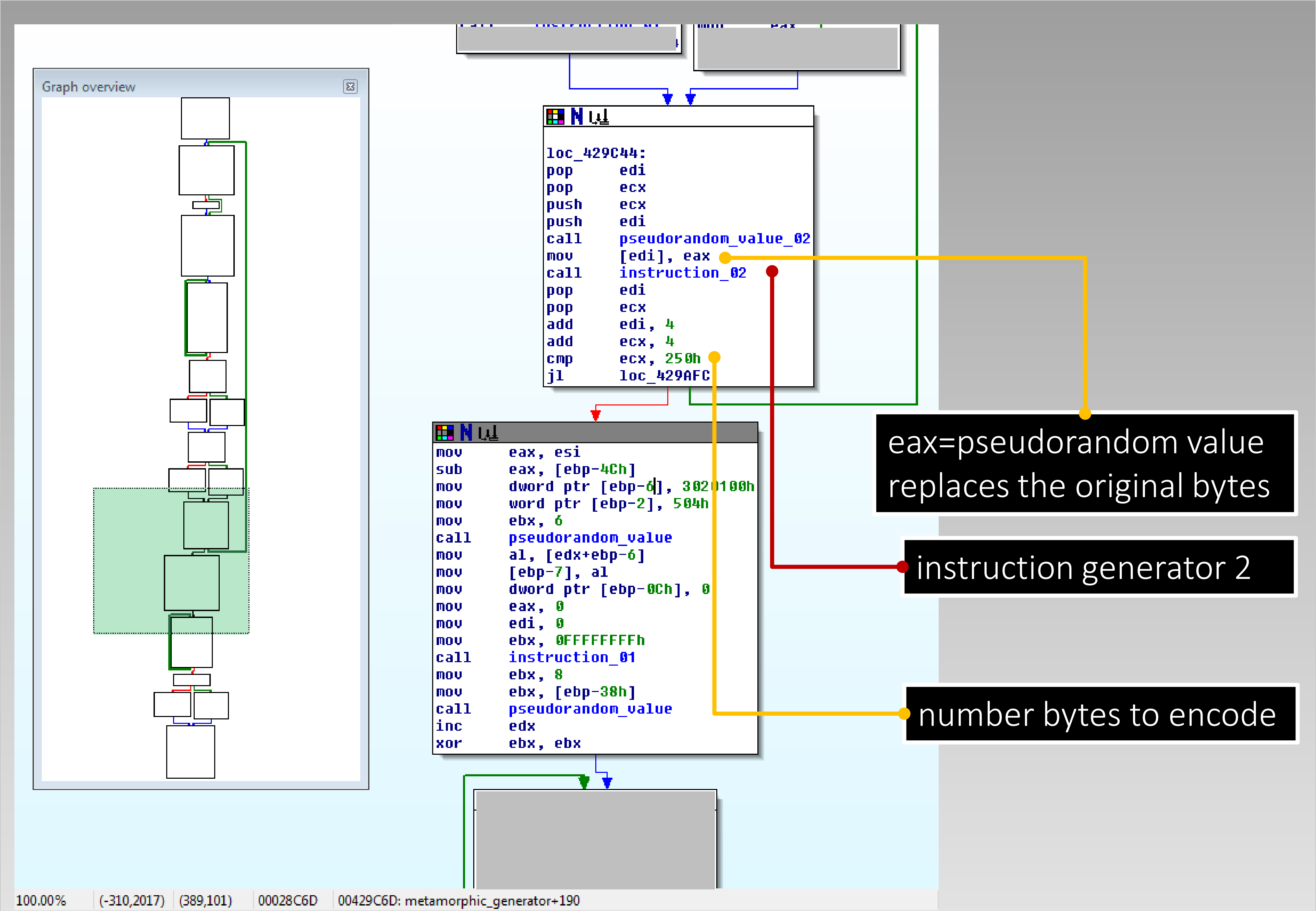
A few more combinations of eax, edi, and ebx registers

Combination of Instructions

MOV			SUB			ADD	
B8	MOV EAX		2D	SUB EAX		05	ADD EAX
BB	MOV EBX		81EB	SUB EBX		81C3	ADD EBX
B9	MOV ECX		81E9	SUB ECX		81C1	ADD ECX
BA	MOV EDX		81EA	SUB EDX		81C2	ADD EDX
BE	MOV ESI		81EE	SUB ESI		81C6	ADD ESI
BF	MOV EDI		81EF	SUB EDI		81C7	ADD EDI



```
Address   Hex dump                                          ASCII
0040234E  90 5D C2 08 00 B8 BB B9 BA BE BF C3 C1 C2 C6 C7  É]┬◘ ╕╗╣║╛┐├┴┬╞╟
0040235E  EB E9 EA EE EF C8 CB C9 CA CE CF 05 1D 0D 15 3D  δΘΩε∩╚╦╔╩╬╧♣↔♪§=
0040236E  35 D0 D3 D1 D2 D7 D6 D0 D3 D1 D2 D6 D7 00 01 89  5╨╙╤╥╫╓╨╙╤╥╓╫ ☺ë
0040237E  18 00 02 89 08 00 03 89 10 00 04 89 30 00 05 89  ↑ ☻ë◘ ♥ë► ♦ë0 ♣ë
0040238E  38 01 00 89 03 01 02 89 0B 01 03 89 13 01 04 89  8☺ ë♥☺☻ë♂☺♥ë‼☺♦ë
0040239E  33 01 05 89 3B 02 00 89 01 02 01 89 19 02 03 89  3☺♣ë;☻ ë☺☻☺ë↓☻♥ë
004023AE  11 02 04 89 31 02 05 89 39 03 00 89 02 03 01 89  ◄☻♦ë1☻♣ë9♥ ë☻♥☺ë
004023BE  1A 03 02 89 0A 03 04 89 32 03 05 89 3A 04 00 89  →♥☻ë◙♥♦ë2♥♣ë:♦ ë
```



```
loc_429C44:
pop     edi
pop     ecx
push    ecx
push    edi
call    pseudorandom_value_02
mov     [edi], eax
call    instruction_02
pop     edi
pop     ecx
add     edi, 4
add     ecx, 4
cmp     ecx, 250h
jl      loc_429AFC
```

```
mov       eax, esi
sub       eax, [ebp-4Ch]
mov       dword ptr [ebp-6], 3020100h
mov       word ptr [ebp-2], 504h
mov       ebx, 6
call      pseudorandom_value
mov       al, [edx+ebp-6]
mov       [ebp-7], al
mov       dword ptr [ebp-0Ch], 0
mov       eax, 0
mov       edi, 0
mov       ebx, 0FFFFFFFFh
call      instruction_01
mov       ebx, 8
mov       ebx, [ebp-38h]
call      pseudorandom_value
inc       edx
xor       ebx, ebx
```

100.00% (-310,2017) (389,101) 00028C6D 00429C6D: metamorphic_generator+190

eax=pseudorandom value replaces the original bytes

instruction generator 2

number bytes to encode

FORTINET

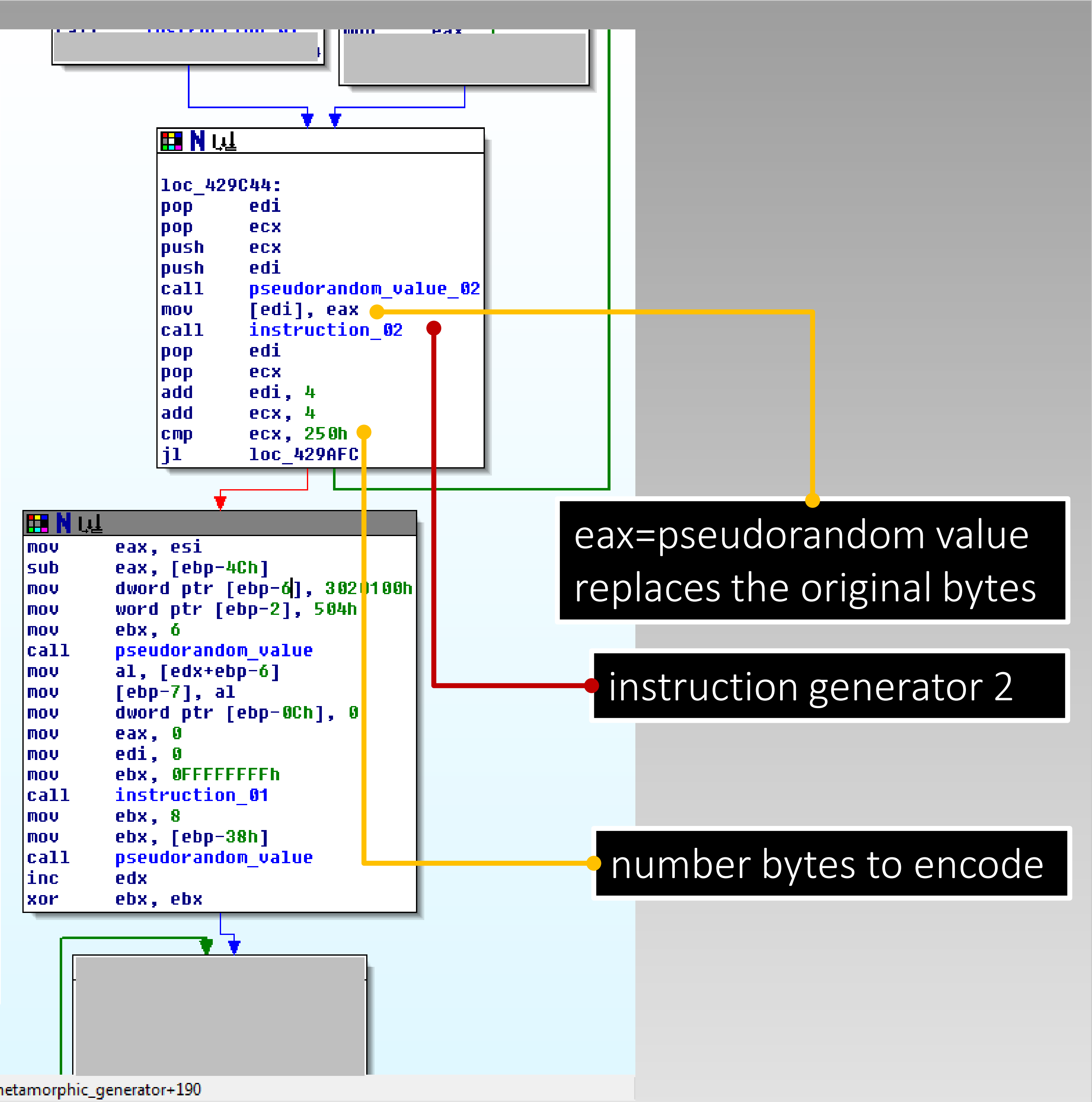
instruction generator 2

-function that generates the final MOV instructions

MOV [addreg], codereg

e.g.,

MOV[ESI], EDX



```
loc_429C44:
pop     edi
pop     ecx
push    ecx
push    edi
call    pseudorandom_value_02
mov     [edi], eax
call    instruction_02
pop     edi
pop     ecx
add     edi, 4
add     ecx, 4
cmp     ecx, 250h
jl      loc_429AFC
```

```
mov     eax, esi
sub     eax, [ebp-4Ch]
mov     dword ptr [ebp-6], 3020100h
mov     word ptr [ebp-2], 504h
mov     ebx, 6
call    pseudorandom_value
mov     al, [edx+ebp-6]
mov     [ebp-7], al
mov     dword ptr [ebp-0Ch], 0
mov     eax, 0
mov     edi, 0
mov     ebx, 0FFFFFFFFh
call    instruction_01
mov     ebx, 8
mov     ebx, [ebp-38h]
call    pseudorandom_value
inc     edx
xor     ebx, ebx
```

100.00% (-310,2017) (389,101) 00028C6D 00429C6D: metamorphic_generator+190

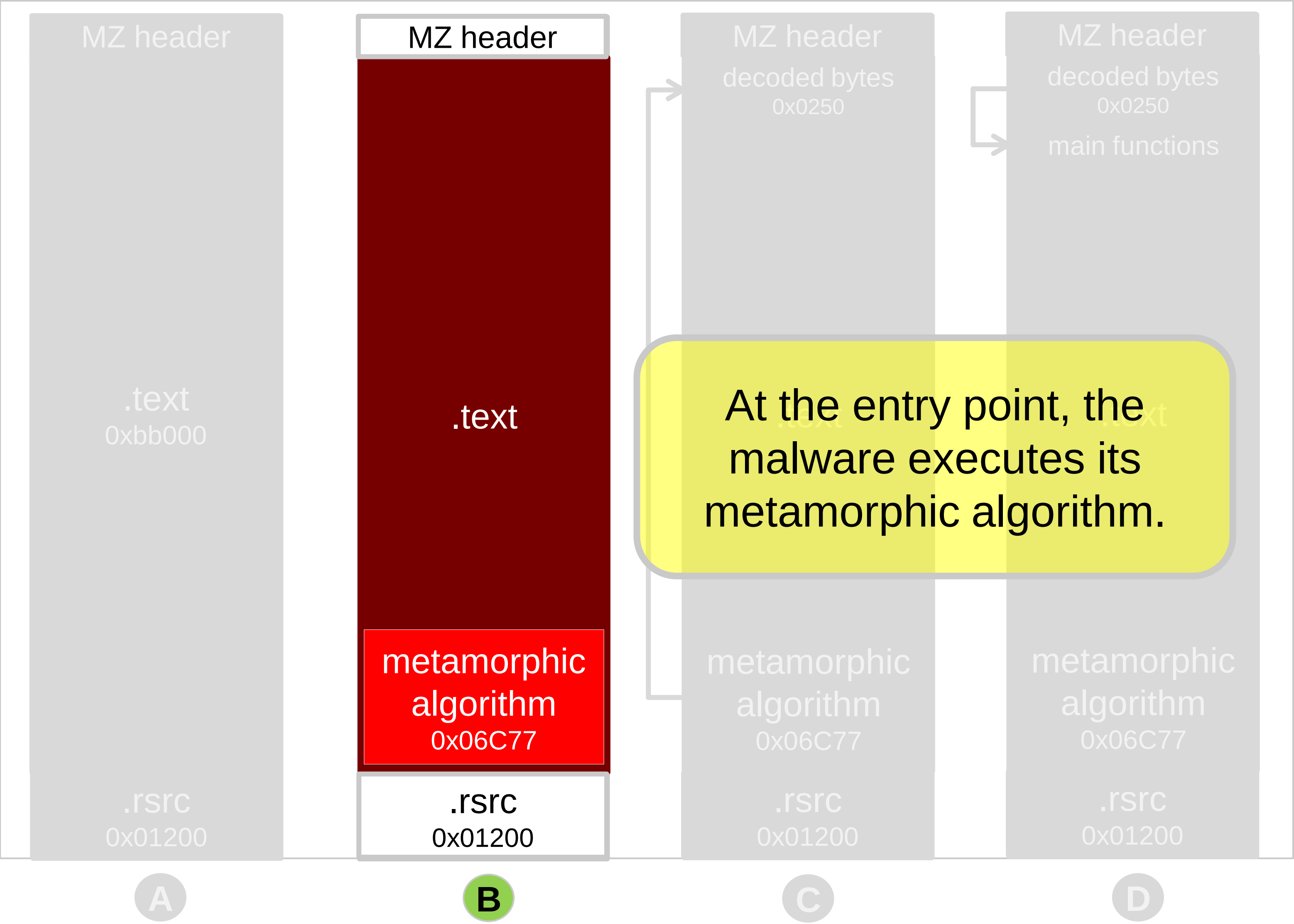
eax=pseudorandom value replaces the original bytes

instruction generator 2

number bytes to encode

Generated Metamorphic Algorithm

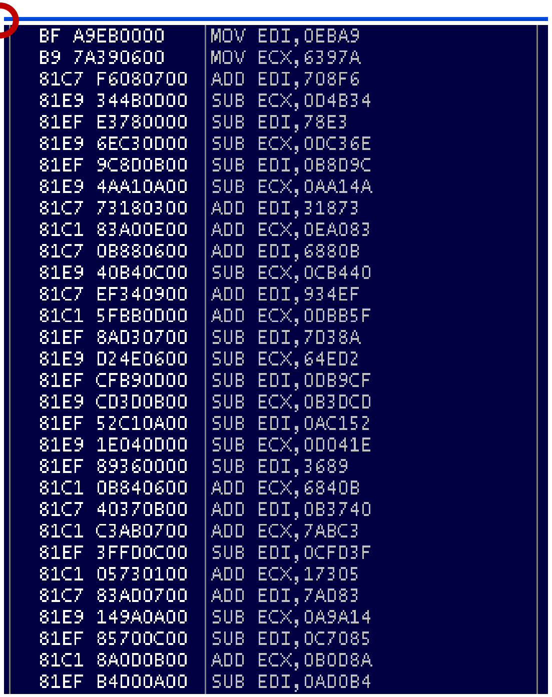
Reversing Stages



At the entry point, the malware executes its metamorphic algorithm.

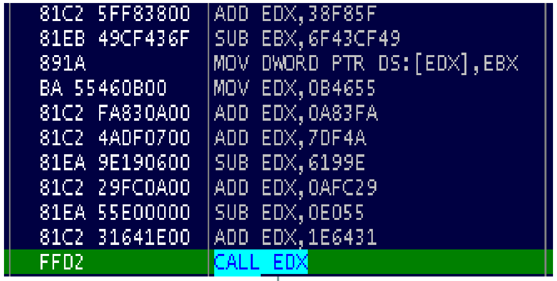
Metamorphic Algorithm (sample 1)

Entry Point



```
BF A9EB0000     MOV EDI,0EBA9
B9 7A390600     MOV ECX,6397A
81C7 F6080700   ADD EDI,708F6
81E9 344B0D00   SUB ECX,0D4B34
81EF E3780000   SUB EDI,78E3
81E9 6EC30D00   SUB ECX,0DC36E
81EF 9C8D0B00   SUB EDI,0B8D9C
81E9 4AA10A00   SUB ECX,0AA14A
81C7 73180300   ADD EDI,31873
81C1 83A00E00   ADD ECX,0EA083
81C7 0B880600   ADD EDI,6880B
81E9 40B40C00   SUB ECX,0CB440
81C7 EF340900   ADD EDI,934EF
81C1 5FBB0D00   ADD ECX,0DBB5F
81EF 8AD30700   SUB EDI,7D38A
81E9 D24E0600   SUB ECX,64ED2
81EF CFB90D00   SUB EDI,0DB9CF
81E9 CD3D0B00   SUB ECX,0B3DCD
81EF 52C10A00   SUB EDI,0AC152
81E9 1E040D00   SUB ECX,0D041E
81EF 89360000   SUB EDI,3689
81C1 0B840600   ADD ECX,6840B
81C7 40370B00   ADD EDI,0B3740
81C1 C3AB0700   ADD ECX,7ABC3
81EF 3FFD0C00   SUB EDI,0CFD3F
81C1 05730100   ADD ECX,17305
81C7 83AD0700   ADD EDI,7AD83
81E9 149A0A00   SUB ECX,0A9A14
81EF 85700C00   SUB EDI,0C7085
81C1 8A0D0B00   ADD ECX,0B0D8A
81EF B4D00A00   SUB EDI,0AD0B4
```

■ ■ ■



```
81C2 5FF83800   ADD EDX,38F85F
81EB 49CF436F   SUB EBX,6F43CF49
891A            MOV DWORD PTR DS:[EDX],EBX
BA 55460B00     MOV EDX,0B4655
81C2 FA830A00   ADD EDX,0A83FA
81C2 4ADF0700   ADD EDX,7DF4A
81EA 9E190600   SUB EDX,6199E
81C2 29FC0A00   ADD EDX,0AFC29
81EA 55E00000   SUB EDX,0E055
81C2 31641E00   ADD EDX,1E6431
FFD2            CALL EDX
```

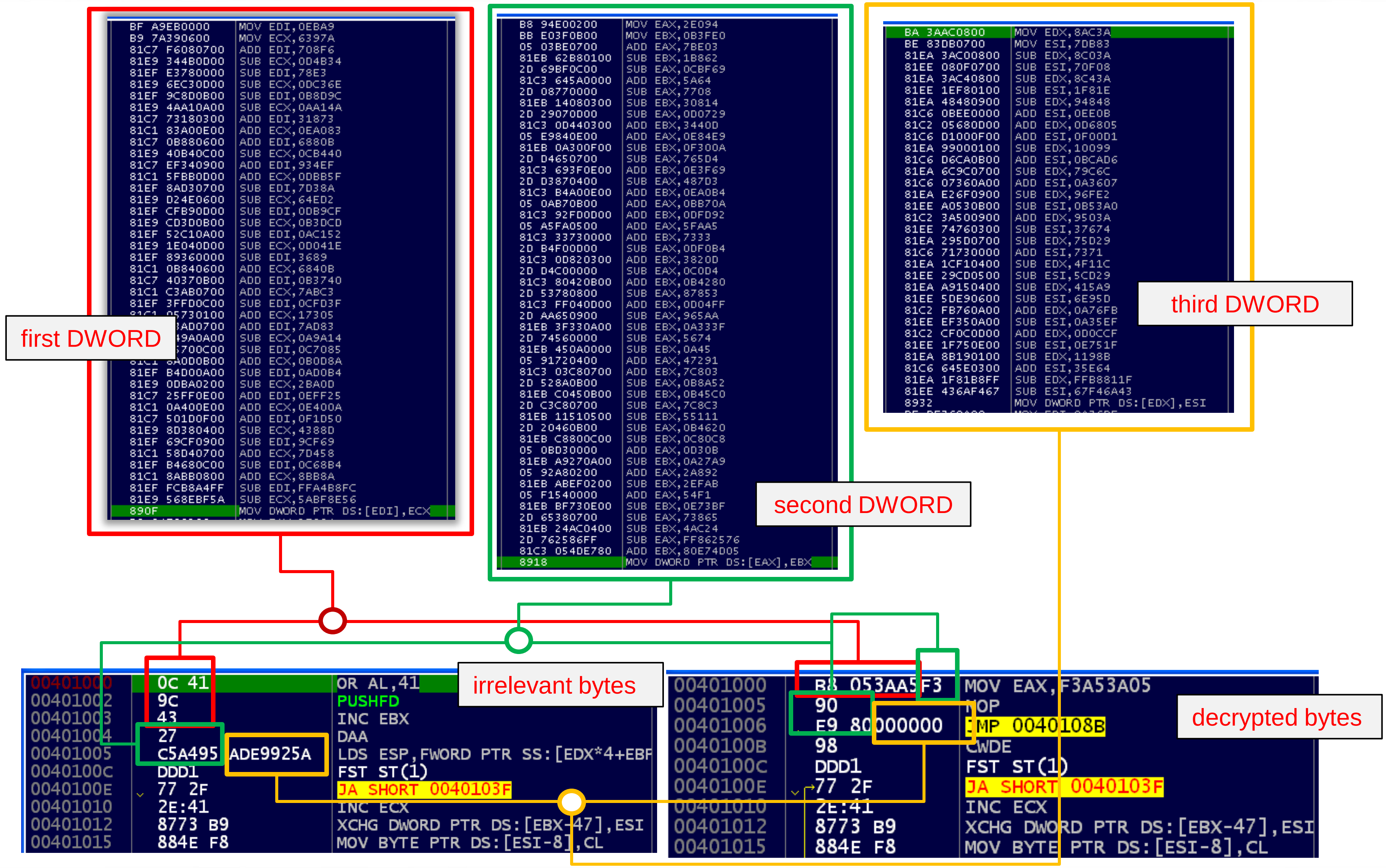
~ 28 kilobytes

The size of the metamorphic code varies per infected file.

Approximately 28kb of code constitutes the metamorphic algorithm that generates the rest of the malicious code, including the polymorphic algorithm.

Call to the decrypted bytes at the start of the .text section.

Metamorphic Algorithm (sample 1)



first DWORD

```
BF A9EB0000      MOV EDI,0EBA9
B9 7A390600      MOV ECX,6397A
81C7 F6080700    ADD EDI,708F6
81E9 344B0D00    SUB ECX,0D4B34
81EF E3780000    SUB EDI,78E3
81E9 6EC30D00    SUB ECX,0DC36E
81EF 9C8D0B00    SUB EDI,0B8D9C
81E9 4AA10A00    SUB ECX,0AA14A
81C7 73180300    ADD EDI,31873
81C1 83A00E00    ADD ECX,0EA083
81C7 0B880600    ADD EDI,6880B
81E9 40B40C00    SUB ECX,0CB440
81C7 EF340900    ADD EDI,934EF
81C1 5FBB0D00    ADD ECX,0DBB5F
81EF 8AD30700    SUB EDI,7D38A
81E9 D24E0600    SUB ECX,64ED2
81EF CFB90D00    SUB EDI,0DB9CF
81E9 CD3D0B00    SUB ECX,0B3DCD
81EF 52C10A00    SUB EDI,0AC152
81E9 1E040D00    SUB ECX,0D041E
81EF 89360000    SUB EDI,3689
81C1 0B840600    ADD ECX,6840B
81C7 40370B00    ADD EDI,0B3740
81C1 C3AB0700    ADD ECX,7ABC3
81EF 3FFD0C00    SUB EDI,0CFD3F
81C1 05730100    ADD ECX,17305
      3AD0700    ADD EDI,7AD83
      49A0A00    SUB ECX,0A9A14
      5700C00    SUB EDI,0C7085
81C1 8A0D0B00    ADD ECX,0B0D8A
81EF B4D00A00    SUB EDI,0AD0B4
81E9 0DBA0200    SUB ECX,2BA0D
81C7 25FF0E00    ADD EDI,0EFF25
81C1 0A400E00    ADD ECX,0E400A
81C7 501D0F00    ADD EDI,0F1D50
81E9 8D380400    SUB ECX,4388D
81EF 69CF0900    SUB EDI,9CF69
81C1 58D40700    ADD ECX,7D458
81EF B4680C00    SUB EDI,0C68B4
81C1 8ABB0800    ADD ECX,8BB8A
81EF FCB8A4FF    SUB EDI,FFA4B8FC
81E9 568EBF5A    SUB ECX,5ABF8E56
890F             MOV DWORD PTR DS:[EDI],ECX
```

second DWORD

```
B8 94E00200      MOV EAX,2E094
BB E03F0B00      MOV EBX,0B3FE0
05 03BE0700      ADD EAX,7BE03
81EB 62B80100    SUB EBX,1B862
2D 69BF0C00      SUB EAX,0CBF69
81C3 645A0000    ADD EBX,5A64
2D 08770000      SUB EAX,7708
81EB 14080300    SUB EBX,30814
2D 29070D00      SUB EAX,0D0729
81C3 0D440300    ADD EBX,3440D
05 E9840E00      ADD EAX,0E84E9
81EB 0A300F00    SUB EBX,0F300A
2D D4650700      SUB EAX,765D4
81C3 693F0E00    ADD EBX,0E3F69
2D D3870400      SUB EAX,487D3
81C3 B4A00E00    ADD EBX,0EA0B4
05 0AB70B00      ADD EAX,0BB70A
81C3 92FD0D00    ADD EBX,0DFD92
05 A5FA0500      ADD EAX,5FAA5
81C3 33730000    ADD EBX,7333
2D B4F00D00      SUB EAX,0DF0B4
81C3 0D820300    ADD EBX,3820D
2D D4C00000      SUB EAX,0C0D4
81C3 80420B00    ADD EBX,0B4280
2D 53780800      SUB EAX,87853
81C3 FF040D00    ADD EBX,0D04FF
2D AA650900      SUB EAX,965AA
81EB 3F330A00    SUB EBX,0A333F
2D 74560000      SUB EAX,5674
81EB 450A0000    SUB EBX,0A45
05 91720400      ADD EAX,47291
81C3 03C80700    ADD EBX,7C803
2D 528A0B00      SUB EAX,0B8A52
81EB C0450B00    SUB EBX,0B45C0
2D C3C80700      SUB EAX,7C8C3
81EB 11510500    SUB EBX,55111
2D 20460B00      SUB EAX,0B4620
81EB C8800C00    SUB EBX,0C80C8
05 0BD30000      ADD EAX,0D30B
81EB A9270A00    SUB EBX,0A27A9
05 92A80200      ADD EAX,2A892
81EB ABEF0200    SUB EBX,2EFAB
05 F1540000      ADD EAX,54F1
81EB BF730E00    SUB EBX,0E73BF
2D 65380700      SUB EAX,73865
81EB 24AC0400    SUB EBX,4AC24
2D 762586FF      SUB EAX,FF862576
81C3 054DE780    ADD EBX,80E74D05
8918             MOV DWORD PTR DS:[EAX],EBX
```

third DWORD

```
BA 3AAC0800      MOV EDX,8AC3A
BE 83DB0700      MOV ESI,7DB83
81EA 3AC00800    SUB EDX,8C03A
81EE 080F0700    SUB ESI,70F08
81EA 3AC40800    SUB EDX,8C43A
81EE 1EF80100    SUB ESI,1F81E
81EA 48480900    SUB EDX,94848
81C6 0BEE0000    ADD ESI,0EE0B
81C2 05680D00    ADD EDX,0D6805
81C6 D1000F00    ADD ESI,0F00D1
81EA 99000100    SUB EDX,10099
81C6 D6CA0B00    ADD ESI,0BCAD6
81EA 6C9C0700    SUB EDX,79C6C
81C6 07360A00    ADD ESI,0A3607
81EA E26F0900    SUB EDX,96FE2
81EE A0530B00    SUB ESI,0B53A0
81C2 3A500900    ADD EDX,9503A
81EE 74760300    SUB ESI,37674
81EA 295D0700    SUB EDX,75D29
81C6 71730000    ADD ESI,7371
81EA 1CF10400    SUB EDX,4F11C
81EE 29CD0500    SUB ESI,5CD29
81EA A9150400    SUB EDX,415A9
81EE 5DE90600    SUB ESI,6E95D
81C2 FB760A00    ADD EDX,0A76FB
81EE EF350A00    SUB ESI,0A35EF
81C2 CF0C0D00    ADD EDX,0D0CCF
81EE 1F750E00    SUB ESI,0E751F
81EA 8B190100    SUB EDX,1198B
81C6 645E0300    ADD ESI,35E64
81EA 1F81B8FF    SUB EDX,FFB8811F
81EE 436AF467    SUB ESI,67F46A43
8932             MOV DWORD PTR DS:[EDX],ESI
```

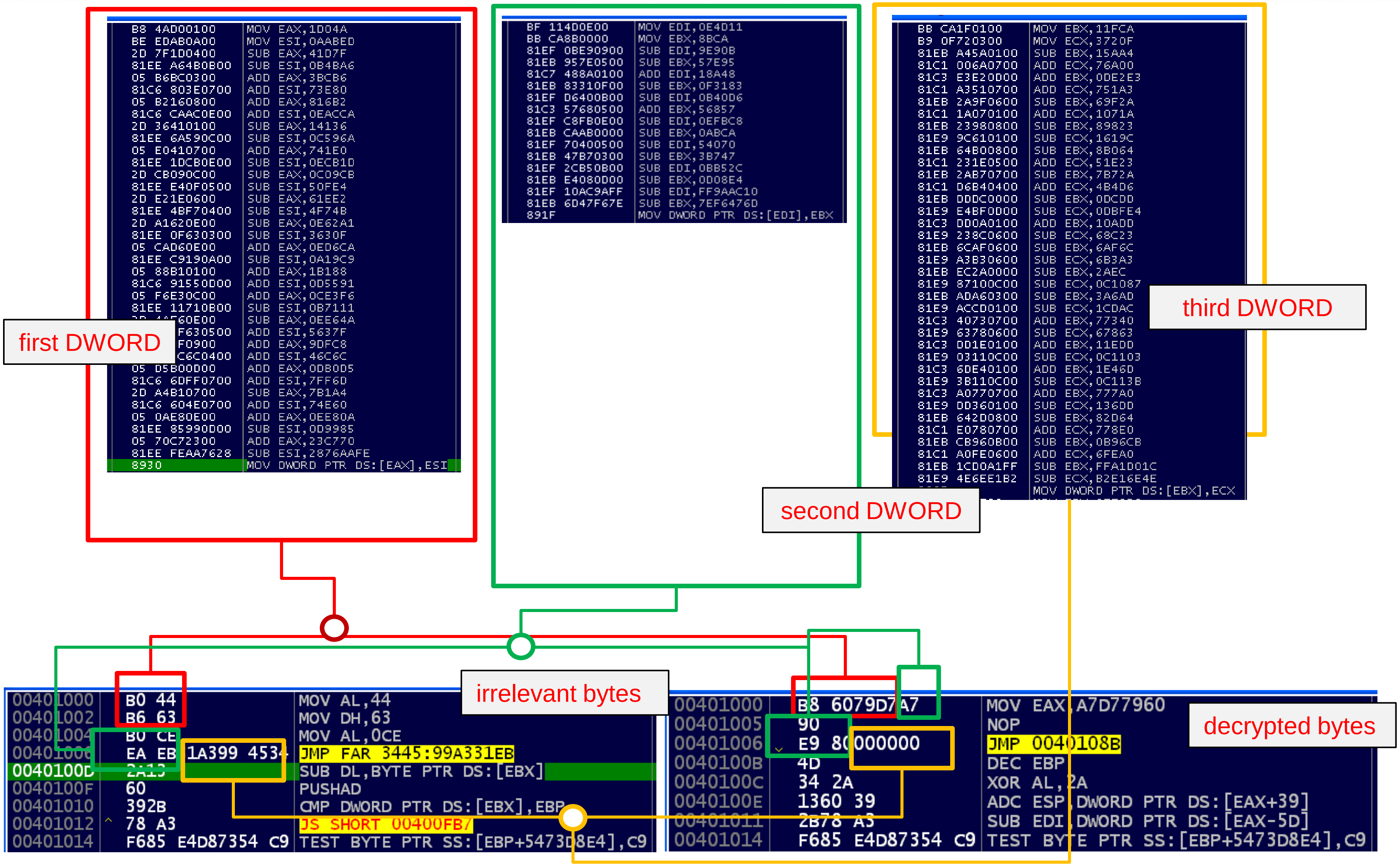
irrelevant bytes

```
00401000   0C 41               OR AL,41
00401002   9C                  PUSHFD
00401003   43                  INC EBX
00401004   27                  DAA
00401005   C5A495 ADE9925A     LDS ESP,FWORD PTR SS:[EDX*4+EBF
0040100C   DDD1                FST ST(1)
0040100E   77 2F               JA SHORT 0040103F
00401010   2E:41               INC ECX
00401012   8773 B9             XCHG DWORD PTR DS:[EBX-47],ESI
00401015   884E F8             MOV BYTE PTR DS:[ESI-8],CL
```

decrypted bytes

```
00401000   B8 053AA5F3         MOV EAX,F3A53A05
00401005   90                  NOP
00401006   E9 80000000         JMP 0040108B
0040100B   98                  CWDE
0040100C   DDD1                FST ST(1)
0040100E   77 2F               JA SHORT 0040103F
00401010   2E:41               INC ECX
00401012   8773 B9             XCHG DWORD PTR DS:[EBX-47],ESI
00401015   884E F8             MOV BYTE PTR DS:[ESI-8],CL
```

Metamorphic Algorithm (sample 2)



first DWORD

```
B8 4AD00100     MOV EAX,1D04A
BE EDAB0A00     MOV ESI,0AABED
2D 7F1D0400     SUB EAX,41D7F
81EE A64B0B00   SUB ESI,0B4BA6
05 B6BC0300     ADD EAX,3BCB6
81C6 803E0700   ADD ESI,73E80
05 B2160800     ADD EAX,816B2
81C6 CAAC0E00   ADD ESI,0EACCA
2D 36410100     SUB EAX,14136
81EE 6A590C00   SUB ESI,0C596A
05 E0410700     ADD EAX,741E0
81EE 1DCB0E00   SUB ESI,0ECB1D
2D CB090C00     SUB EAX,0C09CB
81EE E40F0500   SUB ESI,50FE4
2D E21E0600     SUB EAX,61EE2
81EE 4BF70400   SUB ESI,4F74B
2D A1620E00     SUB EAX,0E62A1
81EE 0F630300   SUB ESI,3630F
05 CAD60E00     ADD EAX,0ED6CA
81EE C9190A00   SUB ESI,0A19C9
05 88B10100     ADD EAX,1B188
81C6 91550D00   ADD ESI,0D5591
05 F6E30C00     ADD EAX,0CE3F6
81EE 11710B00   SUB ESI,0B7111
2D 4AE60E00     SUB EAX,0EE64A
      F630500   ADD ESI,5637F
      F0900     ADD EAX,9DFC8
      C6C0400   ADD ESI,46C6C
05 D5B00D00     ADD EAX,0DB0D5
81C6 6DFF0700   ADD ESI,7FF6D
2D A4B10700     SUB EAX,7B1A4
81C6 604E0700   ADD ESI,74E60
05 0AE80E00     ADD EAX,0EE80A
81EE 85990D00   SUB ESI,0D9985
05 70C72300     ADD EAX,23C770
81EE FEAA7628   SUB ESI,2876AAFE
8930            MOV DWORD PTR DS:[EAX],ESI
```

second DWORD

```
BF 114D0E00     MOV EDI,0E4D11
BB CA8B0000     MOV EBX,8BCA
81EF 0BE90900   SUB EDI,9E90B
81EB 957E0500   SUB EBX,57E95
81C7 488A0100   ADD EDI,18A48
81EB 83310F00   SUB EBX,0F3183
81EF D6400B00   SUB EDI,0B40D6
81C3 57680500   ADD EBX,56857
81EF C8FB0E00   SUB EDI,0EFBC8
81EB CAAB0000   SUB EBX,0ABCA
81EF 70400500   SUB EDI,54070
81EB 47B70300   SUB EBX,3B747
81EF 2CB50B00   SUB EDI,0BB52C
81EB E4080D00   SUB EBX,0D08E4
81EF 10AC9AFF   SUB EDI,FF9AAC10
81EB 6D47F67E   SUB EBX,7EF6476D
891F            MOV DWORD PTR DS:[EDI],EBX
```

third DWORD

```
BB CA1F0100     MOV EBX,11FCA
B9 0F720300     MOV ECX,3720F
81EB A45A0100   SUB EBX,15AA4
81C1 006A0700   ADD ECX,76A00
81C3 E3E20D00   ADD EBX,0DE2E3
81C1 A3510700   ADD ECX,751A3
81EB 2A9F0600   SUB EBX,69F2A
81C1 1A070100   ADD ECX,1071A
81EB 23980800   SUB EBX,89823
81E9 9C610100   SUB ECX,1619C
81EB 64B00800   SUB EBX,8B064
81C1 231E0500   ADD ECX,51E23
81EB 2AB70700   SUB EBX,7B72A
81C1 D6B40400   ADD ECX,4B4D6
81EB DDDC0000   SUB EBX,0DCDD
81E9 E4BF0D00   SUB ECX,0DBFE4
81C3 DD0A0100   ADD EBX,10ADD
81E9 238C0600   SUB ECX,68C23
81EB 6CAF0600   SUB EBX,6AF6C
81E9 A3B30600   SUB ECX,6B3A3
81EB EC2A0000   SUB EBX,2AEC
81E9 87100C00   SUB ECX,0C1087
81EB ADA60300   SUB EBX,3A6AD
81E9 ACCD0100   SUB ECX,1CDAC
81C3 40730700   ADD EBX,77340
81E9 63780600   SUB ECX,67863
81C3 DD1E0100   ADD EBX,11EDD
81E9 03110C00   SUB ECX,0C1103
81C3 6DE40100   ADD EBX,1E46D
81E9 3B110C00   SUB ECX,0C113B
81C3 A0770700   ADD EBX,777A0
81E9 DD360100   SUB ECX,136DD
81EB 642D0800   SUB EBX,82D64
81C1 E0780700   ADD ECX,778E0
81EB CB960B00   SUB EBX,0B96CB
81C1 A0FE0600   ADD ECX,6FEA0
81EB 1CD0A1FF   SUB EBX,FFA1D01C
81E9 4E6EE1B2   SUB ECX,B2E16E4E
                MOV DWORD PTR DS:[EBX],ECX
```

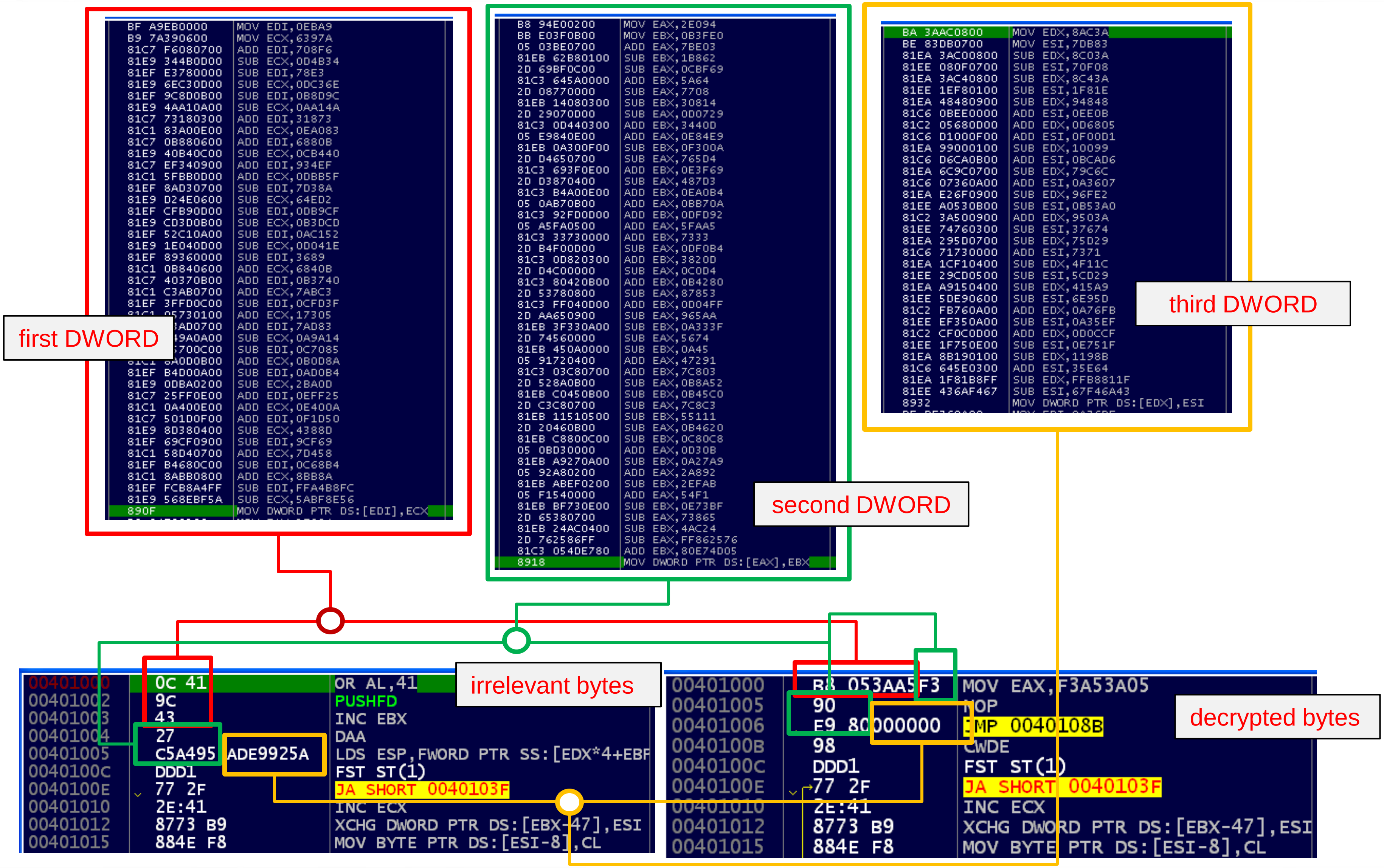
irrelevant bytes

```
00401000  B0 44                  MOV AL,44
00401002  B6 63                  MOV DH,63
00401004  B0 CE                  MOV AL,0CE
00401006  EA EB1A399 4534        JMP FAR 3445:99A331EB
0040100D  2A13                   SUB DL,BYTE PTR DS:[EBX]
0040100F  60                     PUSHAD
00401010  392B                   CMP DWORD PTR DS:[EBX],EBP
00401012  78 A3                  JS SHORT 00400FB7
00401014  F685 E4D87354 C9       TEST BYTE PTR SS:[EBP+5473D8E4],C9
```

decrypted bytes

```
00401000  B8 6079D7A7            MOV EAX,A7D77960
00401005  90                     NOP
00401006  E9 80000000            JMP 0040108B
0040100B  4D                     DEC EBP
0040100C  34 2A                  XOR AL,2A
0040100E  1360 39                ADC ESP,DWORD PTR DS:[EAX+39]
00401011  2B78 A3                SUB EDI,DWORD PTR DS:[EAX-5D]
00401014  F685 E4D87354 C9       TEST BYTE PTR SS:[EBP+5473D8E4],C9
```

Metamorphic Algorithm (sample 1)



first DWORD

```
BF A9EB0000     MOV EDI,0EBA9
B9 7A390600     MOV ECX,6397A
81C7 F6080700   ADD EDI,708F6
81E9 344B0D00   SUB ECX,0D4B34
81EF E3780000   SUB EDI,78E3
81E9 6EC30D00   SUB ECX,0DC36E
81EF 9C8D0B00   SUB EDI,0B8D9C
81E9 4AA10A00   SUB ECX,0AA14A
81C7 73180300   ADD EDI,31873
81C1 83A00E00   ADD ECX,0EA083
81C7 0B880600   ADD EDI,6880B
81E9 40B40C00   SUB ECX,0CB440
81C7 EF340900   ADD EDI,934EF
81C1 5FBB0D00   ADD ECX,0DBB5F
81EF 8AD30700   SUB EDI,7D38A
81E9 D24E0600   SUB ECX,64ED2
81EF CFB90D00   SUB EDI,0DB9CF
81E9 CD3D0B00   SUB ECX,0B3DCD
81EF 52C10A00   SUB EDI,0AC152
81E9 1E040D00   SUB ECX,0D041E
81EF 89360000   SUB EDI,3689
81C1 0B840600   ADD ECX,6840B
81C7 40370B00   ADD EDI,0B3740
81C1 C3AB0700   ADD ECX,7ABC3
81EF 3FFD0C00   SUB EDI,0CFD3F
81C1 05730100   ADD ECX,17305
      3AD0700   ADD EDI,7AD83
      49A0A00   SUB ECX,0A9A14
      5700C00   SUB EDI,0C7085
81C1 8A0D0B00   ADD ECX,0B0D8A
81EF B4D00A00   SUB EDI,0AD0B4
81E9 0DBA0200   SUB ECX,2BA0D
81C7 25FF0E00   ADD EDI,0EFF25
81C1 0A400E00   ADD ECX,0E400A
81C7 501D0F00   ADD EDI,0F1D50
81E9 8D380400   SUB ECX,4388D
81EF 69CF0900   SUB EDI,9CF69
81C1 58D40700   ADD ECX,7D458
81EF B4680C00   SUB EDI,0C68B4
81C1 8ABB0800   ADD ECX,8BB8A
81EF FCB8A4FF   SUB EDI,FFA4B8FC
81E9 568EBF5A   SUB ECX,5ABF8E56
890F            MOV DWORD PTR DS:[EDI],ECX
```

second DWORD

```
B8 94E00200     MOV EAX,2E094
BB E03F0B00     MOV EBX,0B3FE0
05 03BE0700     ADD EAX,7BE03
81EB 62B80100   SUB EBX,1B862
2D 69BF0C00     SUB EAX,0CBF69
81C3 645A0000   ADD EBX,5A64
2D 08770000     SUB EAX,7708
81EB 14080300   SUB EBX,30814
2D 29070D00     SUB EAX,0D0729
81C3 0D440300   ADD EBX,3440D
05 E9840E00     ADD EAX,0E84E9
81EB 0A300F00   SUB EBX,0F300A
2D D4650700     SUB EAX,765D4
81C3 693F0E00   ADD EBX,0E3F69
2D D3870400     SUB EAX,487D3
81C3 B4A00E00   ADD EBX,0EA0B4
05 0AB70B00     ADD EAX,0BB70A
81C3 92FD0D00   ADD EBX,0DFD92
05 A5FA0500     ADD EAX,5FAA5
81C3 33730000   ADD EBX,7333
2D B4F00D00     SUB EAX,0DF0B4
81C3 0D820300   ADD EBX,3820D
2D D4C00000     SUB EAX,0C0D4
81C3 80420B00   ADD EBX,0B4280
2D 53780800     SUB EAX,87853
81C3 FF040D00   ADD EBX,0D04FF
2D AA650900     SUB EAX,965AA
81EB 3F330A00   SUB EBX,0A333F
2D 74560000     SUB EAX,5674
81EB 450A0000   SUB EBX,0A45
05 91720400     ADD EAX,47291
81C3 03C80700   ADD EBX,7C803
2D 528A0B00     SUB EAX,0B8A52
81EB C0450B00   SUB EBX,0B45C0
2D C3C80700     SUB EAX,7C8C3
81EB 11510500   SUB EBX,55111
2D 20460B00     SUB EAX,0B4620
81EB C8800C00   SUB EBX,0C80C8
05 0BD30000     ADD EAX,0D30B
81EB A9270A00   SUB EBX,0A27A9
05 92A80200     ADD EAX,2A892
81EB ABEF0200   SUB EBX,2EFAB
05 F1540000     ADD EAX,54F1
81EB BF730E00   SUB EBX,0E73BF
2D 65380700     SUB EAX,73865
81EB 24AC0400   SUB EBX,4AC24
2D 762586FF     SUB EAX,FF862576
81C3 054DE780   ADD EBX,80E74D05
8918            MOV DWORD PTR DS:[EAX],EBX
```

third DWORD

```
BA 3AAC0800     MOV EDX,8AC3A
BE 83DB0700     MOV ESI,7DB83
81EA 3AC00800   SUB EDX,8C03A
81EE 080F0700   SUB ESI,70F08
81EA 3AC40800   SUB EDX,8C43A
81EE 1EF80100   SUB ESI,1F81E
81EA 48480900   SUB EDX,94848
81C6 0BEE0000   ADD ESI,0EE0B
81C2 05680D00   ADD EDX,0D6805
81C6 D1000F00   ADD ESI,0F00D1
81EA 99000100   SUB EDX,10099
81C6 D6CA0B00   ADD ESI,0BCAD6
81EA 6C9C0700   SUB EDX,79C6C
81C6 07360A00   ADD ESI,0A3607
81EA E26F0900   SUB EDX,96FE2
81EE A0530B00   SUB ESI,0B53A0
81C2 3A500900   ADD EDX,9503A
81EE 74760300   SUB ESI,37674
81EA 295D0700   SUB EDX,75D29
81C6 71730000   ADD ESI,7371
81EA 1CF10400   SUB EDX,4F11C
81EE 29CD0500   SUB ESI,5CD29
81EA A9150400   SUB EDX,415A9
81EE 5DE90600   SUB ESI,6E95D
81C2 FB760A00   ADD EDX,0A76FB
81EE EF350A00   SUB ESI,0A35EF
81C2 CF0C0D00   ADD EDX,0D0CCF
81EE 1F750E00   SUB ESI,0E751F
81EA 8B190100   SUB EDX,1198B
81C6 645E0300   ADD ESI,35E64
81EA 1F81B8FF   SUB EDX,FFB8811F
81EE 436AF467   SUB ESI,67F46A43
8932            MOV DWORD PTR DS:[EDX],ESI
```

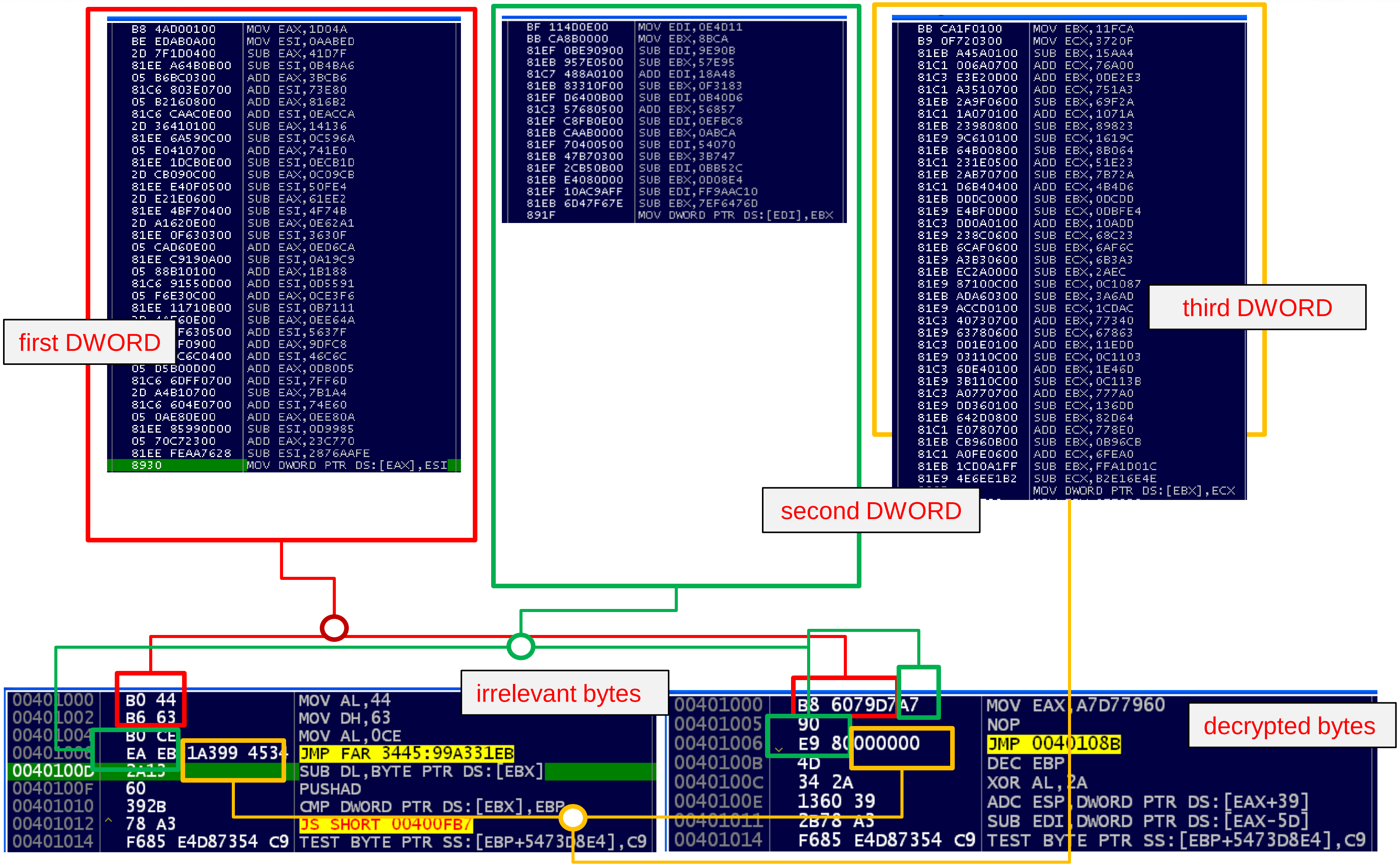
irrelevant bytes

```
00401000   0C 41                OR AL,41
00401002   9C                   PUSHFD
00401003   43                   INC EBX
00401004   27                   DAA
00401005   C5A495 ADE9925A      LDS ESP,FWORD PTR SS:[EDX*4+EBF
0040100C   DDD1                 FST ST(1)
0040100E   77 2F                JA SHORT 0040103F
00401010   2E:41                INC ECX
00401012   8773 B9              XCHG DWORD PTR DS:[EBX-47],ESI
00401015   884E F8              MOV BYTE PTR DS:[ESI-8],CL
```

decrypted bytes

```
00401000   B8 053AA5F3          MOV EAX,F3A53A05
00401005   90                   NOP
00401006   E9 80000000          JMP 0040108B
0040100B   98                   CWDE
0040100C   DDD1                 FST ST(1)
0040100E   77 2F                JA SHORT 0040103F
00401010   2E:41                INC ECX
00401012   8773 B9              XCHG DWORD PTR DS:[EBX-47],ESI
00401015   884E F8              MOV BYTE PTR DS:[ESI-8],CL
```

Metamorphic Algorithm (sample 2)



first DWORD

```
B8 4AD00100      MOV EAX,1D04A
BE EDAB0A00      MOV ESI,0AABED
2D 7F1D0400      SUB EAX,41D7F
81EE A64B0B00    SUB ESI,0B4BA6
05 B6BC0300      ADD EAX,3BCB6
81C6 803E0700    ADD ESI,73E80
05 B2160800      ADD EAX,816B2
81C6 CAAC0E00    ADD ESI,0EACCA
2D 36410100      SUB EAX,14136
81EE 6A590C00    SUB ESI,0C596A
05 E0410700      ADD EAX,741E0
81EE 1DCB0E00    SUB ESI,0ECB1D
2D CB090C00      SUB EAX,0C09CB
81EE E40F0500    SUB ESI,50FE4
2D E21E0600      SUB EAX,61EE2
81EE 4BF70400    SUB ESI,4F74B
2D A1620E00      SUB EAX,0E62A1
81EE 0F630300    SUB ESI,3630F
05 CAD60E00      ADD EAX,0ED6CA
81EE C9190A00    SUB ESI,0A19C9
05 88B10100      ADD EAX,1B188
81C6 91550D00    ADD ESI,0D5591
05 F6E30C00      ADD EAX,0CE3F6
81EE 11710B00    SUB ESI,0B7111
2D 4AE60E00      SUB EAX,0EE64A
      F630500    ADD ESI,5637F
       F0900     ADD EAX,9DFC8
      C6C0400    ADD ESI,46C6C
05 D5B00D00      ADD EAX,0DB0D5
81C6 6DFF0700    ADD ESI,7FF6D
2D A4B10700      SUB EAX,7B1A4
81C6 604E0700    ADD ESI,74E60
05 0AE80E00      ADD EAX,0EE80A
81EE 85990D00    SUB ESI,0D9985
05 70C72300      ADD EAX,23C770
81EE FEAA7628    SUB ESI,2876AAFE
8930             MOV DWORD PTR DS:[EAX],ESI
```

second DWORD

```
BF 114D0E00      MOV EDI,0E4D11
BB CA8B0000      MOV EBX,8BCA
81EF 0BE90900    SUB EDI,9E90B
81EB 957E0500    SUB EBX,57E95
81C7 488A0100    ADD EDI,18A48
81EB 83310F00    SUB EBX,0F3183
81EF D6400B00    SUB EDI,0B40D6
81C3 57680500    ADD EBX,56857
81EF C8FB0E00    SUB EDI,0EFBC8
81EB CAAB0000    SUB EBX,0ABCA
81EF 70400500    SUB EDI,54070
81EB 47B70300    SUB EBX,3B747
81EF 2CB50B00    SUB EDI,0BB52C
81EB E4080D00    SUB EBX,0D08E4
81EF 10AC9AFF    SUB EDI,FF9AAC10
81EB 6D47F67E    SUB EBX,7EF6476D
891F             MOV DWORD PTR DS:[EDI],EBX
```

third DWORD

```
BB CA1F0100      MOV EBX,11FCA
B9 0F720300      MOV ECX,3720F
81EB A45A0100    SUB EBX,15AA4
81C1 006A0700    ADD ECX,76A00
81C3 E3E20D00    ADD EBX,0DE2E3
81C1 A3510700    ADD ECX,751A3
81EB 2A9F0600    SUB EBX,69F2A
81C1 1A070100    ADD ECX,1071A
81EB 23980800    SUB EBX,89823
81E9 9C610100    SUB ECX,1619C
81EB 64B00800    SUB EBX,8B064
81C1 231E0500    ADD ECX,51E23
81EB 2AB70700    SUB EBX,7B72A
81C1 D6B40400    ADD ECX,4B4D6
81EB DDDC0000    SUB EBX,0DCDD
81E9 E4BF0D00    SUB ECX,0DBFE4
81C3 DD0A0100    ADD EBX,10ADD
81E9 238C0600    SUB ECX,68C23
81EB 6CAF0600    SUB EBX,6AF6C
81E9 A3B30600    SUB ECX,6B3A3
81EB EC2A0000    SUB EBX,2AEC
81E9 87100C00    SUB ECX,0C1087
81EB ADA60300    SUB EBX,3A6AD
81E9 ACCD0100    SUB ECX,1CDAC
81C3 40730700    ADD EBX,77340
81E9 63780600    SUB ECX,67863
81C3 DD1E0100    ADD EBX,11EDD
81E9 03110C00    SUB ECX,0C1103
81C3 6DE40100    ADD EBX,1E46D
81E9 3B110C00    SUB ECX,0C113B
81C3 A0770700    ADD EBX,777A0
81E9 DD360100    SUB ECX,136DD
81EB 642D0800    SUB EBX,82D64
81C1 E0780700    ADD ECX,778E0
81EB CB960B00    SUB EBX,0B96CB
81C1 A0FE0600    ADD ECX,6FEA0
81EB 1CD0A1FF    SUB EBX,FFA1D01C
81E9 4E6EE1B2    SUB ECX,B2E16E4E
                 MOV DWORD PTR DS:[EBX],ECX
```

irrelevant bytes

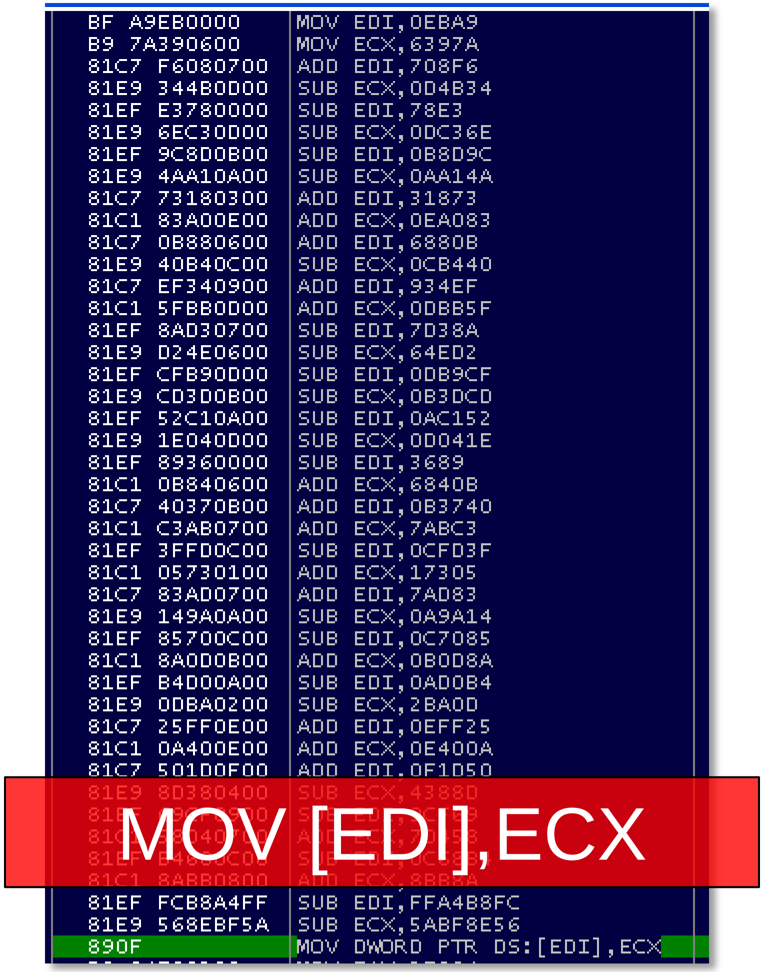
```
00401000    B0 44                   MOV AL,44
00401002    B6 63                   MOV DH,63
00401004    B0 CE                   MOV AL,0CE
00401006    EA EB1A399 4534         JMP FAR 3445:99A331EB
0040100D    2A13                    SUB DL,BYTE PTR DS:[EBX]
0040100F    60                      PUSHAD
00401010    392B                    CMP DWORD PTR DS:[EBX],EBP
00401012  ^ 78 A3                   JS SHORT 00400FB7
00401014    F685 E4D87354 C9        TEST BYTE PTR SS:[EBP+5473D8E4],C9
```

decrypted bytes

```
00401000    B8 6079D7A7             MOV EAX,A7D77960
00401005    90                      NOP
00401006  v E9 80000000             JMP 0040108B
0040100B    4D                      DEC EBP
0040100C    34 2A                   XOR AL,2A
0040100E    1360 39                 ADC ESP,DWORD PTR DS:[EAX+39]
00401011    2B78 A3                 SUB EDI,DWORD PTR DS:[EAX-5D]
00401014    F685 E4D87354 C9        TEST BYTE PTR SS:[EBP+5473D8E4],C9
```

Metamorphic Algorithm (comparison)

first DWORD

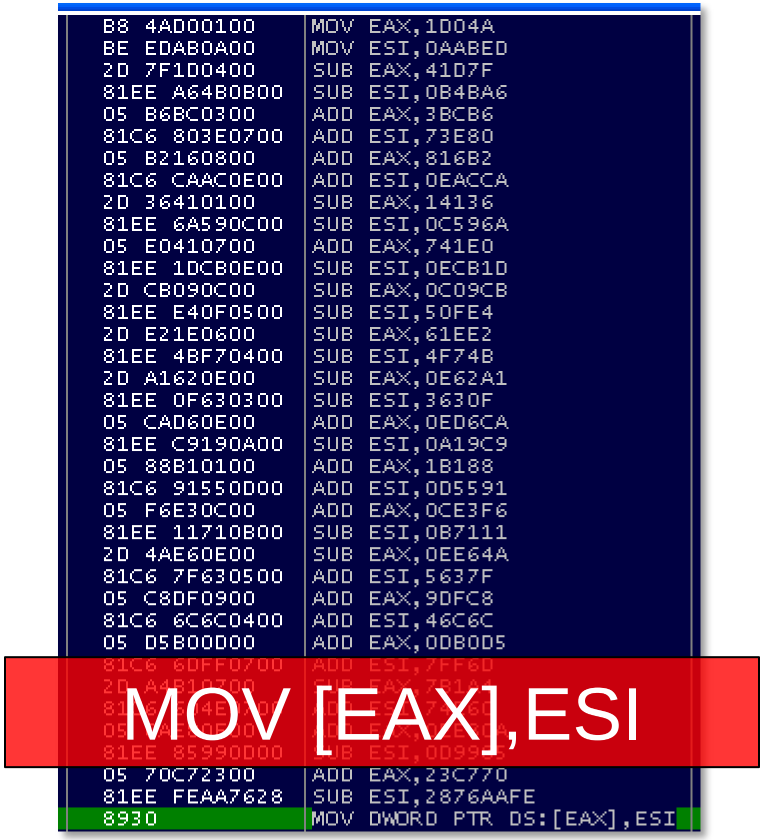


```
BF A9EB0000       MOV EDI,0EBA9
B9 7A390600       MOV ECX,6397A
81C7 F6080700     ADD EDI,708F6
81E9 344B0D00     SUB ECX,0D4B34
81EF E3780000     SUB EDI,78E3
81E9 6EC30D00     SUB ECX,0DC36E
81EF 9C8D0B00     SUB EDI,0B8D9C
81E9 4AA10A00     SUB ECX,0AA14A
81C7 73180300     ADD EDI,31873
81C1 83A00E00     ADD ECX,0EA083
81C7 0B880600     ADD EDI,6880B
81E9 40B40C00     SUB ECX,0CB440
81C7 EF340900     ADD EDI,934EF
81C1 5FBB0D00     ADD ECX,0DBB5F
81EF 8AD30700     SUB EDI,7D38A
81E9 D24E0600     SUB ECX,64ED2
81EF CFB90D00     SUB EDI,0DB9CF
81E9 CD3D0B00     SUB ECX,0B3DCD
81EF 52C10A00     SUB EDI,0AC152
81E9 1E040D00     SUB ECX,0D041E
81EF 89360000     SUB EDI,3689
81C1 0B840600     ADD ECX,6840B
81C7 40370B00     ADD EDI,0B3740
81C1 C3AB0700     ADD ECX,7ABC3
81EF 3FFD0C00     SUB EDI,0CFD3F
81C1 05730100     ADD ECX,17305
81C7 83AD0700     ADD EDI,7AD83
81E9 149A0A00     SUB ECX,0A9A14
81EF 85700C00     SUB EDI,0C7085
81C1 8A0D0B00     ADD ECX,0B0D8A
81EF B4D00A00     SUB EDI,0AD0B4
81E9 0DBA0200     SUB ECX,2BA0D
81C7 25FF0E00     ADD EDI,0EFF25
81C1 0A400E00     ADD ECX,0E400A
81C7 501D0F00     ADD EDI,0F1D50
```

MOV [EDI],ECX

```
81EF FCB8A4FF     SUB EDI,FFA4B8FC
81E9 568EBF5A     SUB ECX,5ABF8E56
890F              MOV DWORD PTR DS:[EDI],ECX
```

Sample 1



```
B8 4AD00100       MOV EAX,1D04A
BE EDAB0A00       MOV ESI,0AABED
2D 7F1D0400       SUB EAX,41D7F
81EE A64B0B00     SUB ESI,0B4BA6
05 B6BC0300       ADD EAX,3BCB6
81C6 803E0700     ADD ESI,73E80
05 B2160800       ADD EAX,816B2
81C6 CAAC0E00     ADD ESI,0EACCA
2D 36410100       SUB EAX,14136
81EE 6A590C00     SUB ESI,0C596A
05 E0410700       ADD EAX,741E0
81EE 1DCB0E00     SUB ESI,0ECB1D
2D CB090C00       SUB EAX,0C09CB
81EE E40F0500     SUB ESI,50FE4
2D E21E0600       SUB EAX,61EE2
81EE 4BF70400     SUB ESI,4F74B
2D A1620E00       SUB EAX,0E62A1
81EE 0F630300     SUB ESI,3630F
05 CAD60E00       ADD EAX,0ED6CA
81EE C9190A00     SUB ESI,0A19C9
05 88B10100       ADD EAX,1B188
81C6 91550D00     ADD ESI,0D5591
05 F6E30C00       ADD EAX,0CE3F6
81EE 11710B00     SUB ESI,0B7111
2D 4AE60E00       SUB EAX,0EE64A
81C6 7F630500     ADD ESI,5637F
05 C8DF0900       ADD EAX,9DFC8
81C6 6C6C0400     ADD ESI,46C6C
05 D5B00D00       ADD EAX,0DB0D5
```

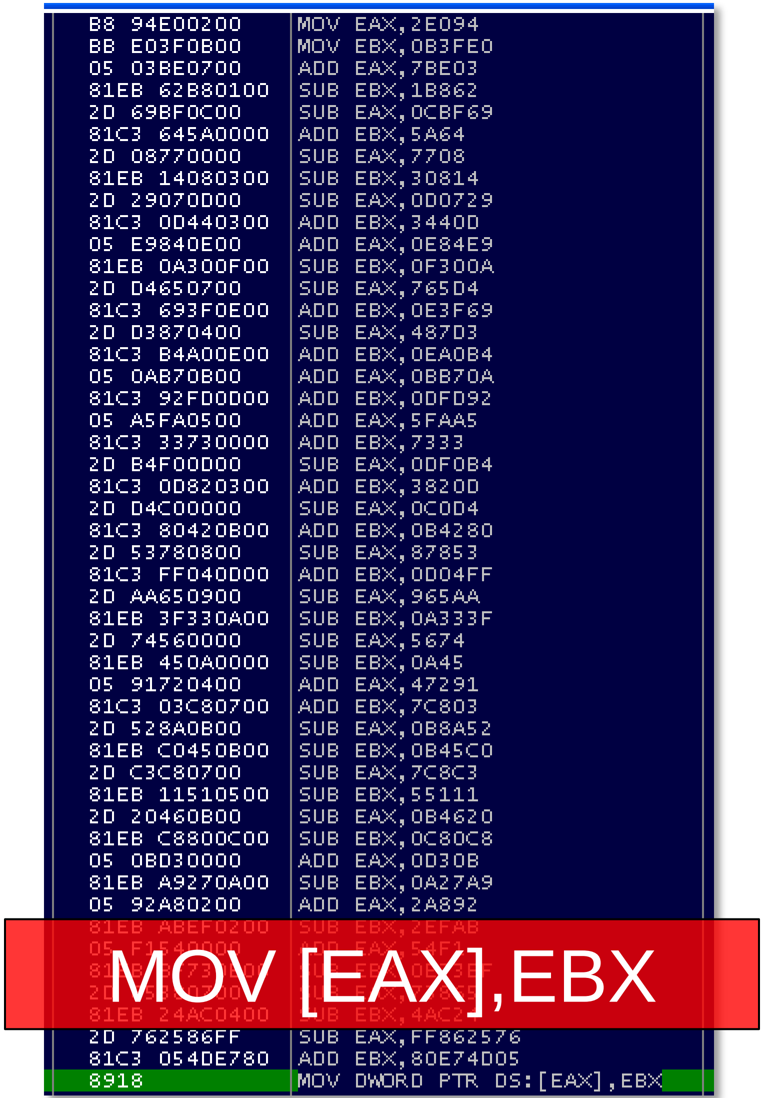
MOV [EAX],ESI

```
05 70C72300       ADD EAX,23C770
81EE FEAA7628     SUB ESI,2876AAFE
8930              MOV DWORD PTR DS:[EAX],ESI
```

Sample 2

Metamorphic Algorithm (comparison)

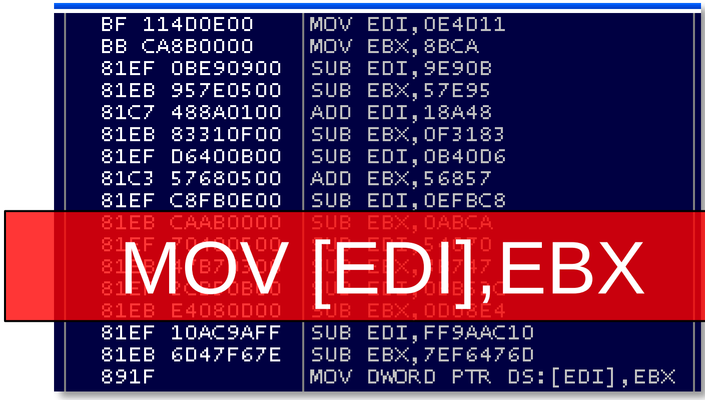
second DWORD



```
B8 94E00200       MOV EAX,2E094
BB E03F0B00       MOV EBX,0B3FE0
05 03BE0700       ADD EAX,7BE03
81EB 62B80100     SUB EBX,1B862
2D 69BF0C00       SUB EAX,0CBF69
81C3 645A0000     ADD EBX,5A64
2D 08770000       SUB EAX,7708
81EB 14080300     SUB EBX,30814
2D 29070D00       SUB EAX,0D0729
81C3 0D440300     ADD EBX,3440D
05 E9840E00       ADD EAX,0E84E9
81EB 0A300F00     SUB EBX,0F300A
2D D4650700       SUB EAX,765D4
81C3 693F0E00     ADD EBX,0E3F69
2D D3870400       SUB EAX,487D3
81C3 B4A00E00     ADD EBX,0EA0B4
05 0AB70B00       ADD EAX,0BB70A
81C3 92FD0D00     ADD EBX,0DFD92
05 A5FA0500       ADD EAX,5FAA5
81C3 33730000     ADD EBX,7333
2D B4F00D00       SUB EAX,0DF0B4
81C3 0D820300     ADD EBX,3820D
2D D4C00000       SUB EAX,0C0D4
81C3 80420B00     ADD EBX,0B4280
2D 53780800       SUB EAX,87853
81C3 FF040D00     ADD EBX,0D04FF
2D AA650900       SUB EAX,965AA
81EB 3F330A00     SUB EBX,0A333F
2D 74560000       SUB EAX,5674
81EB 450A0000     SUB EBX,0A45
05 91720400       ADD EAX,47291
81C3 03C80700     ADD EBX,7C803
2D 528A0B00       SUB EAX,0B8A52
81EB C0450B00     SUB EBX,0B45C0
2D C3C80700       SUB EAX,7C8C3
81EB 11510500     SUB EBX,55111
2D 20460B00       SUB EAX,0B4620
81EB C8800C00     SUB EBX,0C80C8
05 0BD30000       ADD EAX,0D30B
81EB A9270A00     SUB EBX,0A27A9
05 92A80200       ADD EAX,2A892
81EB ABEF0200     SUB EBX,2EFAB
05 F1540000       ADD EAX,54F1
[illegible]
[illegible]
81EB 24AC0400     SUB EBX,4AC24
2D 762586FF       SUB EAX,FF862576
81C3 054DE780     ADD EBX,80E74D05
8918              MOV DWORD PTR DS:[EAX],EBX
```

MOV [EAX],EBX

Sample 1



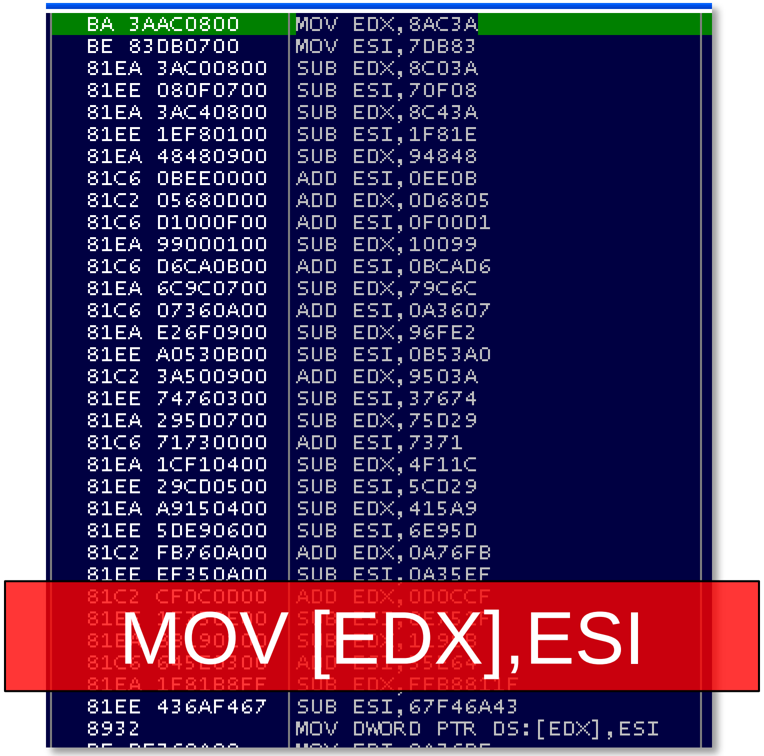
```
BF 114D0E00       MOV EDI,0E4D11
BB CA8B0000       MOV EBX,8BCA
81EF 0BE90900     SUB EDI,9E90B
81EB 957E0500     SUB EBX,57E95
81C7 488A0100     ADD EDI,18A48
81EB 83310F00     SUB EBX,0F3183
81EF D6400B00     SUB EDI,0B40D6
81C3 57680500     ADD EBX,56857
81EF C8FB0E00     SUB EDI,0EFBC8
81EB CAAB0000     SUB EBX,0ABCA
[illegible]
[illegible]
[illegible]
81EB E4080D00     SUB EBX,0D08E4
81EF 10AC9AFF     SUB EDI,FF9AAC10
81EB 6D47F67E     SUB EBX,7EF6476D
891F              MOV DWORD PTR DS:[EDI],EBX
```

MOV [EDI],EBX

Sample 2

Metamorphic Algorithm (comparison)

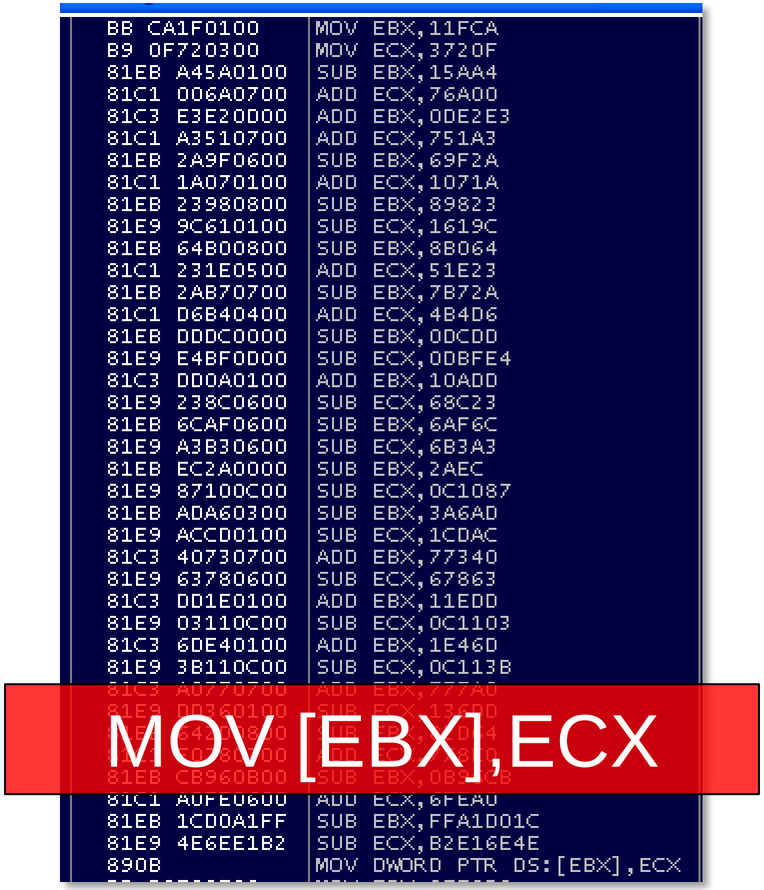
third DWORD



```
BA 3AAC0800         MOV EDX,8AC3A
BE 83DB0700         MOV ESI,7DB83
81EA 3AC00800       SUB EDX,8C03A
81EE 080F0700       SUB ESI,70F08
81EA 3AC40800       SUB EDX,8C43A
81EE 1EF80100       SUB ESI,1F81E
81EA 48480900       SUB EDX,94848
81C6 0BEE0000       ADD ESI,0EE0B
81C2 05680D00       ADD EDX,0D6805
81C6 D1000F00       ADD ESI,0F00D1
81EA 99000100       SUB EDX,10099
81C6 D6CA0B00       ADD ESI,0BCAD6
81EA 6C9C0700       SUB EDX,79C6C
81C6 07360A00       ADD ESI,0A3607
81EA E26F0900       SUB EDX,96FE2
81EE A0530B00       SUB ESI,0B53A0
81C2 3A500900       ADD EDX,9503A
81EE 74760300       SUB ESI,37674
81EA 295D0700       SUB EDX,75D29
81C6 71730000       ADD ESI,7371
81EA 1CF10400       SUB EDX,4F11C
81EE 29CD0500       SUB ESI,5CD29
81EA A9150400       SUB EDX,415A9
81EE 5DE90600       SUB ESI,6E95D
81C2 FB760A00       ADD EDX,0A76FB
81EE EF350A00       SUB ESI,0A35EF
81C2 CF0C0D00       ADD EDX,0D0CCF
81EE 436AF467       SUB ESI,67F46A43
8932                MOV DWORD PTR DS:[EDX],ESI
```

MOV [EDX],ESI

Sample 1

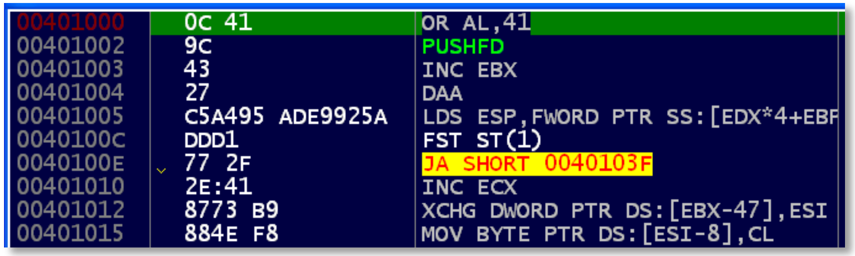


```
BB CA1F0100         MOV EBX,11FCA
B9 0F720300         MOV ECX,3720F
81EB A45A0100       SUB EBX,15AA4
81C1 006A0700       ADD ECX,76A00
81C3 E3E20D00       ADD EBX,0DE2E3
81C1 A3510700       ADD ECX,751A3
81EB 2A9F0600       SUB EBX,69F2A
81C1 1A070100       ADD ECX,1071A
81EB 23980800       SUB EBX,89823
81E9 9C610100       SUB ECX,1619C
81EB 64B00800       SUB EBX,8B064
81C1 231E0500       ADD ECX,51E23
81EB 2AB70700       SUB EBX,7B72A
81C1 D6B40400       ADD ECX,4B4D6
81EB DDDC0000       SUB EBX,0DCDD
81E9 E4BF0D00       SUB ECX,0DBFE4
81C3 DD0A0100       ADD EBX,10ADD
81E9 238C0600       SUB ECX,68C23
81EB 6CAF0600       SUB EBX,6AF6C
81E9 A3B30600       SUB ECX,6B3A3
81EB EC2A0000       SUB EBX,2AEC
81E9 87100C00       SUB ECX,0C1087
81EB ADA60300       SUB EBX,3A6AD
81E9 ACCD0100       SUB ECX,1CDAC
81C3 40730700       ADD EBX,77340
81E9 63780600       SUB ECX,67863
81C3 DD1E0100       ADD EBX,11EDD
81E9 03110C00       SUB ECX,0C1103
81C3 6DE40100       ADD EBX,1E46D
81E9 3B110C00       SUB ECX,0C113B
81EB 1CD0A1FF       SUB EBX,FFA1D01C
81E9 4E6EE1B2       SUB ECX,B2E16E4E
890B                MOV DWORD PTR DS:[EBX],ECX
```

MOV [EBX],ECX

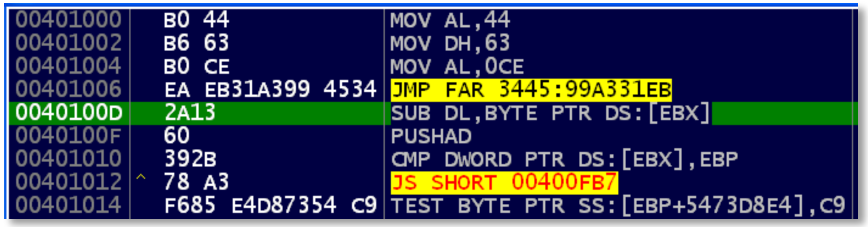
Sample 2

Metamorphic Algorithm (detection)



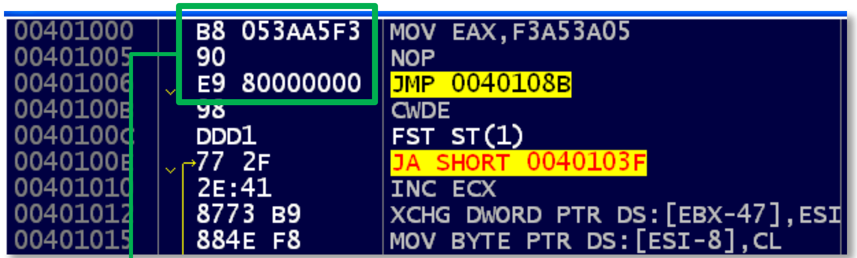
```
00401000   0C 41               OR AL,41
00401002   9C                  PUSHFD
00401003   43                  INC EBX
00401004   27                  DAA
00401005   C5A495 ADE9925A     LDS ESP,FWORD PTR SS:[EDX*4+EBF
0040100C   DDD1                FST ST(1)
0040100E   77 2F               JA SHORT 0040103F
00401010   2E:41               INC ECX
00401012   8773 B9             XCHG DWORD PTR DS:[EBX-47],ESI
00401015   884E F8             MOV BYTE PTR DS:[ESI-8],CL
```

Sample 1

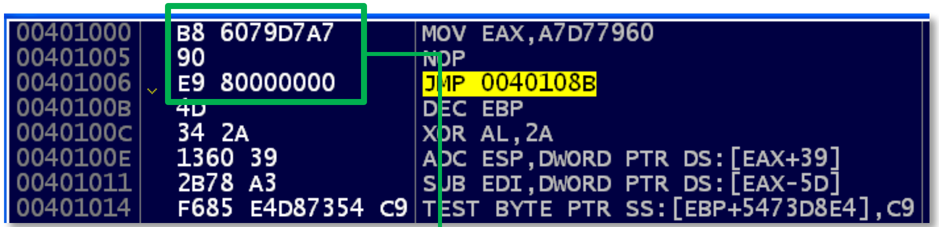


```
00401000   B0 44               MOV AL,44
00401002   B6 63               MOV DH,63
00401004   B0 CE               MOV AL,0CE
00401006   EA EB31A399 4534    JMP FAR 3445:99A331EB
0040100D   2A13                SUB DL,BYTE PTR DS:[EBX]
0040100F   60                  PUSHAD
00401010   392B                CMP DWORD PTR DS:[EBX],EBP
00401012   78 A3               JS SHORT 00400FB7
00401014   F685 E4D87354 C9    TEST BYTE PTR SS:[EBP+5473D8E4],C9
```

Sample 2



```
00401000   B8 053AA5F3         MOV EAX,F3A53A05
00401005   90                  NOP
00401006   E9 80000000         JMP 0040108B
0040100B   98                  CWDE
0040100C   DDD1                FST ST(1)
0040100E   77 2F               JA SHORT 0040103F
00401010   2E:41               INC ECX
00401012   8773 B9             XCHG DWORD PTR DS:[EBX-47],ESI
00401015   884E F8             MOV BYTE PTR DS:[ESI-8],CL
```



```
00401000   B8 6079D7A7         MOV EAX,A7D77960
00401005   90                  NOP
00401006   E9 80000000         JMP 0040108B
0040100B   4D                  DEC EBP
0040100C   34 2A               XOR AL,2A
0040100E   1360 39             ADC ESP,DWORD PTR DS:[EAX+39]
00401011   2B78 A3             SUB EDI,DWORD PTR DS:[EAX-5D]
00401014   F685 E4D87354 C9    TEST BYTE PTR SS:[EBP+5473D8E4],C9
```

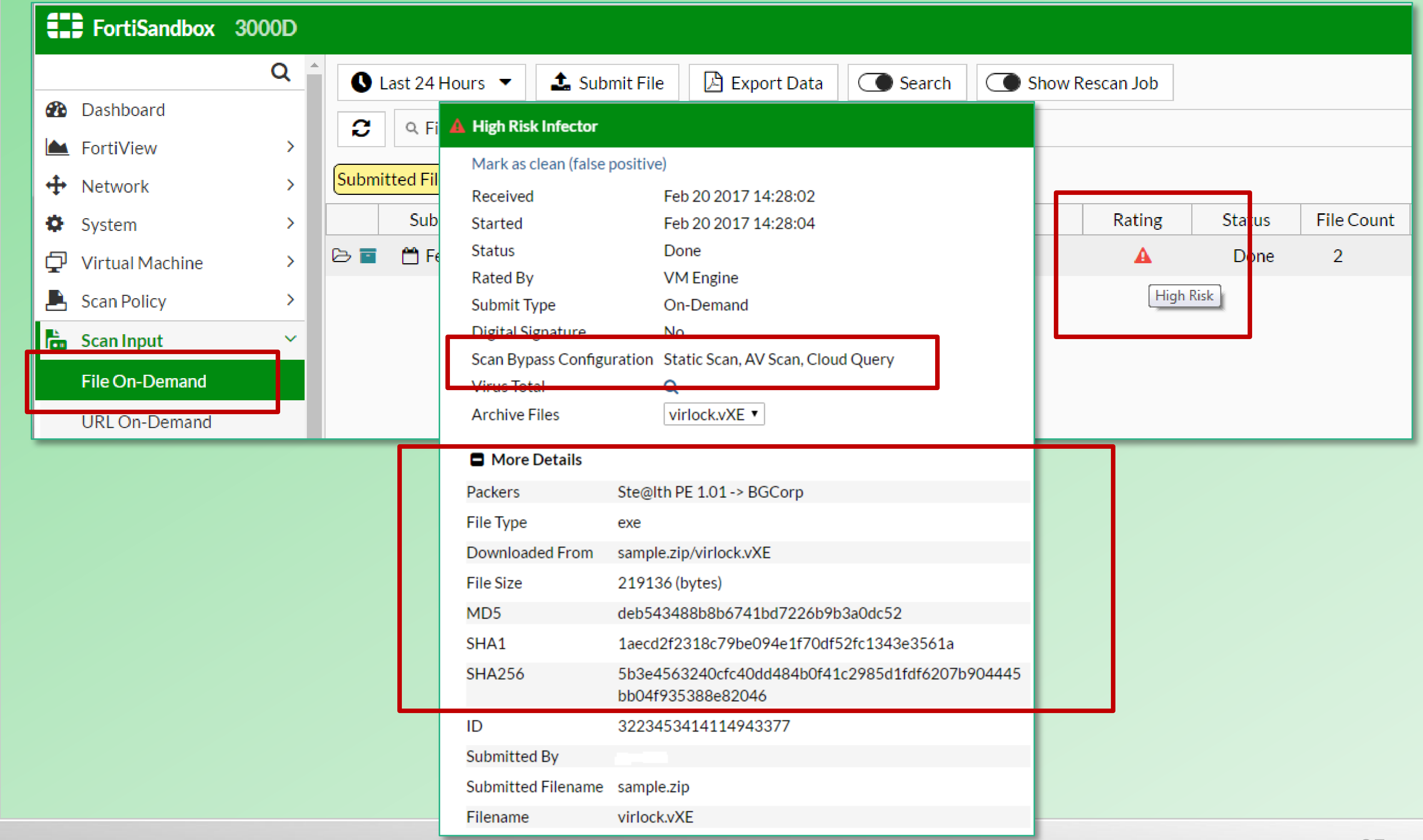
MOV EAX, --------

NOP

JMP 0040108B

Automated Detection

FortiSandbox



High Risk Infector

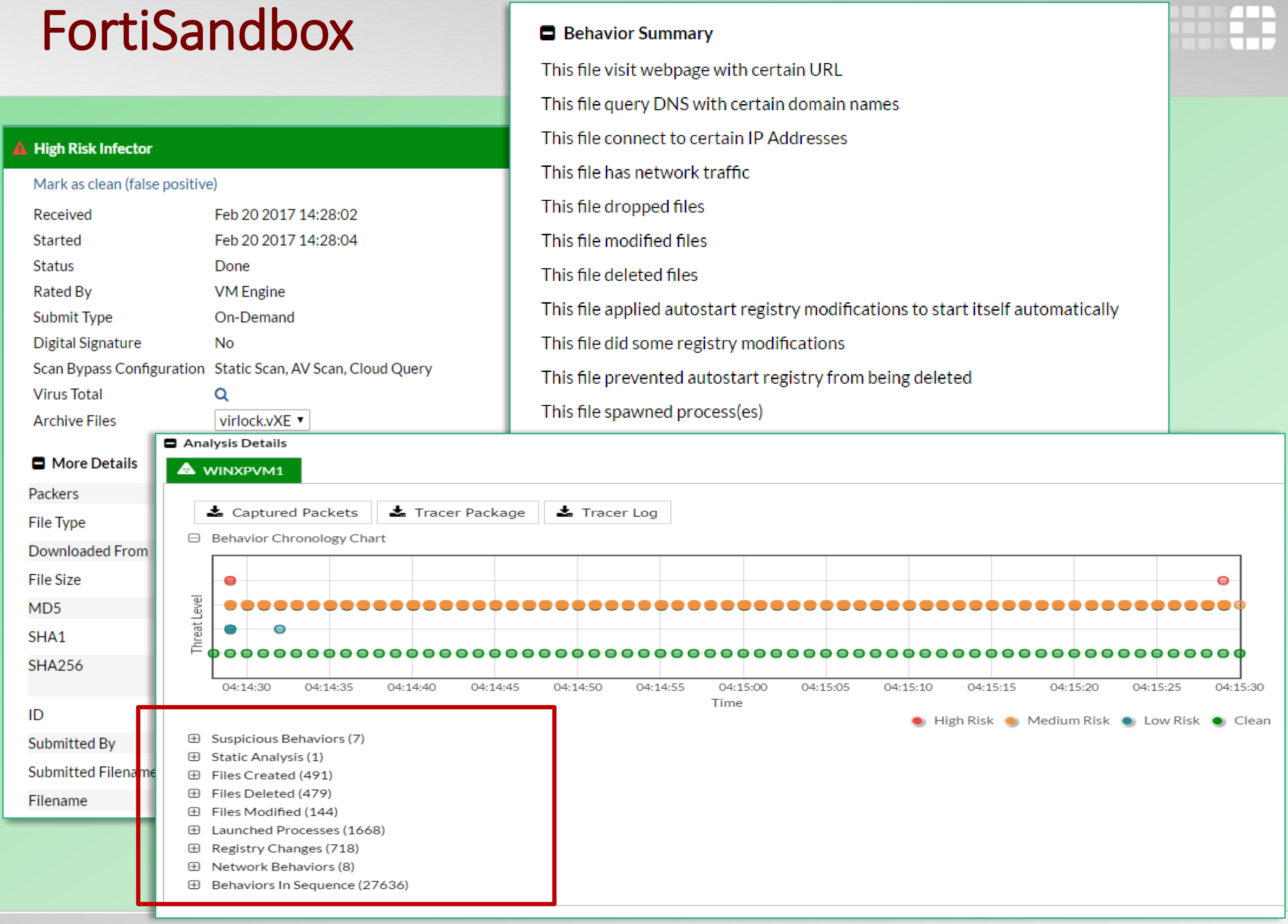
Mark as clean (false positive)

Received	Feb 20 2017 14:28:02
Started	Feb 20 2017 14:28:04
Status	Done
Rated By	VM Engine
Submit Type	On-Demand
Digital Signature	No
Scan Bypass Configuration	Static Scan, AV Scan, Cloud Query
Virus Total	🔍
Archive Files	virlock.vXE ▾

More Details

Packers	Ste@lth PE 1.01 -> BGCorp
File Type	exe
Downloaded From	sample.zip/virlock.vXE
File Size	219136 (bytes)
MD5	deb543488b8b6741bd7226b9b3a0dc52
SHA1	1aecd2f2318c79be094e1f70df52fc1343e3561a
SHA256	5b3e4563240cfc40dd484b0f41c2985d1fdf6207b904445bb04f935388e82046
ID	3223453414114943377
Submitted By	[redacted]
Submitted Filename	sample.zip
Filename	virlock.vXE

FortiSandbox



High Risk Infector

Mark as clean (false positive)

Received	Feb 20 2017 14:28:02
Started	Feb 20 2017 14:28:04
Status	Done
Rated By	VM Engine
Submit Type	On-Demand
Digital Signature	No
Scan Bypass Configuration	Static Scan, AV Scan, Cloud Query
Virus Total	🔍
Archive Files	virlock.vXE ▾

More Details

Packers

File Type

Downloaded From

File Size

MD5

SHA1

SHA256

ID

Submitted By

Submitted Filename

Filename

Behavior Summary

This file visit webpage with certain URL

This file query DNS with certain domain names

This file connect to certain IP Addresses

This file has network traffic

This file dropped files

This file modified files

This file deleted files

This file applied autostart registry modifications to start itself automatically

This file did some registry modifications

This file prevented autostart registry from being deleted

This file spawned process(es)

Analysis Details

WINXPVM1

Captured Packets | Tracer Package | Tracer Log

⊟ Behavior Chronology Chart

⊞ Suspicious Behaviors (7)

⊞ Static Analysis (1)

⊞ Files Created (491)

⊞ Files Deleted (479)

⊞ Files Modified (144)

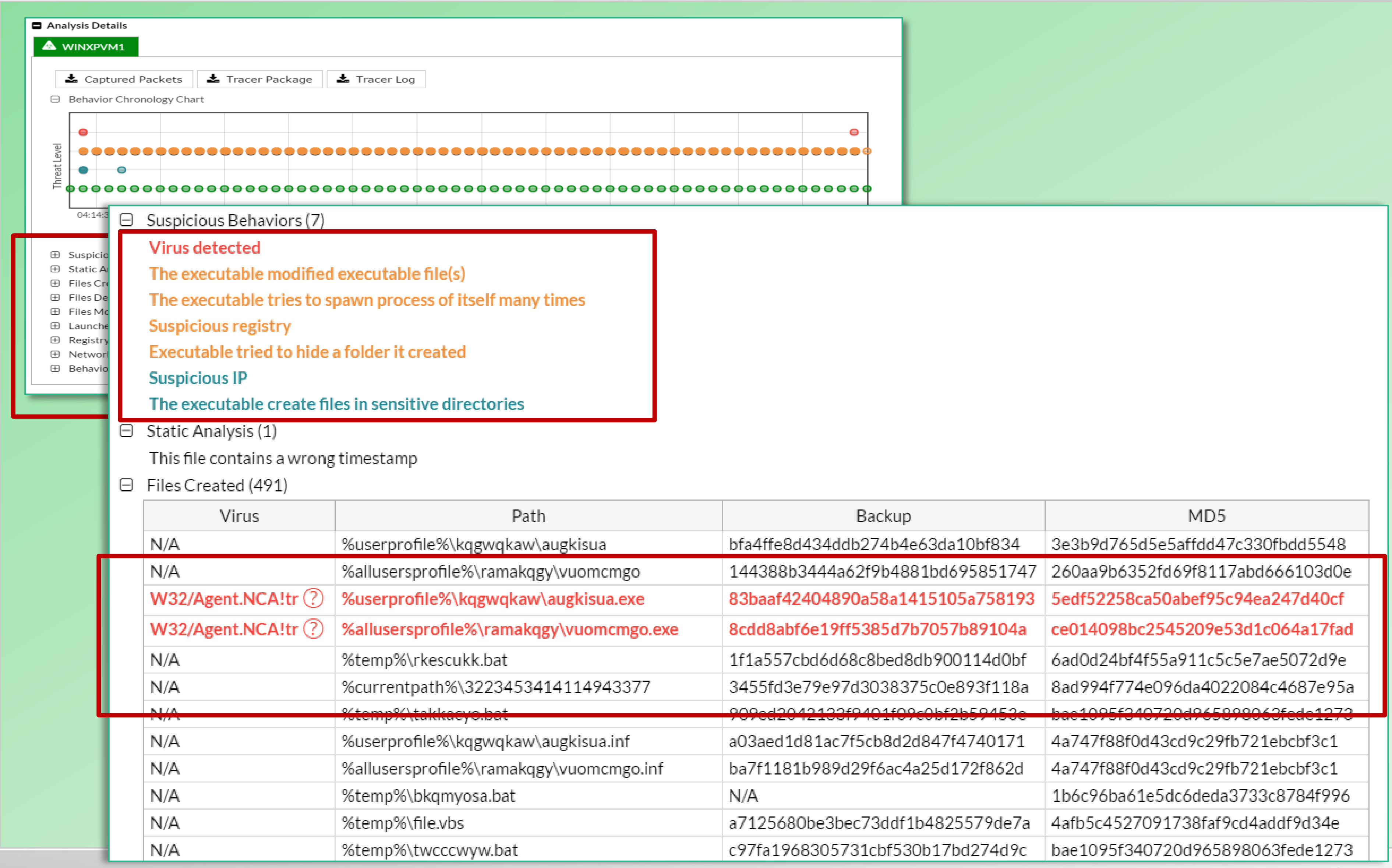
⊞ Launched Processes (1668)

⊞ Registry Changes (718)

⊞ Network Behaviors (8)

⊞ Behaviors In Sequence (27636)

FortiSandbox



⊟ Suspicious Behaviors (7)

Virus detected

The executable modified executable file(s)

The executable tries to spawn process of itself many times

Suspicious registry

Executable tried to hide a folder it created

Suspicious IP

The executable create files in sensitive directories

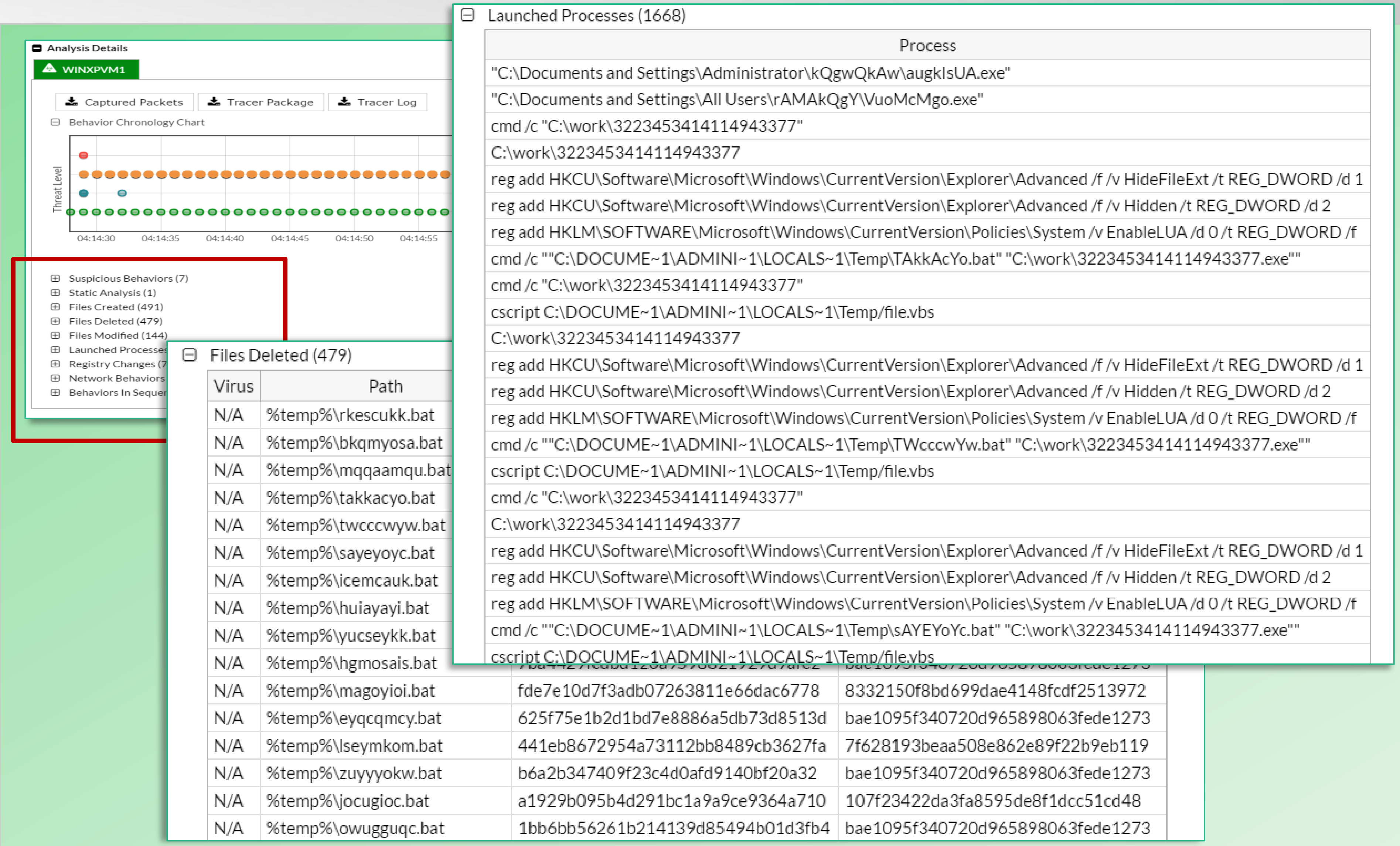
⊟ Static Analysis (1)

This file contains a wrong timestamp

⊟ Files Created (491)

Virus	Path	Backup	MD5
N/A	%userprofile%\kqgwqkaw\augkisua	bfa4ffe8d434ddb274b4e63da10bf834	3e3b9d765d5e5affdd47c330fbdd5548
N/A	%allusersprofile%\ramakqgy\vuomcmgo	144388b3444a62f9b4881bd695851747	260aa9b6352fd69f8117abd666103d0e
W32/Agent.NCA!tr ?	%userprofile%\kqgwqkaw\augkisua.exe	83baaf42404890a58a1415105a758193	5edf52258ca50abef95c94ea247d40cf
W32/Agent.NCA!tr ?	%allusersprofile%\ramakqgy\vuomcmgo.exe	8cdd8abf6e19ff5385d7b7057b89104a	ce014098bc2545209e53d1c064a17fad
N/A	%temp%\rkescukk.bat	1f1a557cbd6d68c8bed8db900114d0bf	6ad0d24bf4f55a911c5c5e7ae5072d9e
N/A	%currentpath%\3223453414114943377	3455fd3e79e97d3038375c0e893f118a	8ad994f774e096da4022084c4687e95a
N/A	%temp%\takkacyc.bat	909ed2042133f9401f09c0bf2b59453c	bae1095f340720d965898063fede1273
N/A	%userprofile%\kqgwqkaw\augkisua.inf	a03aed1d81ac7f5cb8d2d847f4740171	4a747f88f0d43cd9c29fb721ebcbf3c1
N/A	%allusersprofile%\ramakqgy\vuomcmgo.inf	ba7f1181b989d29f6ac4a25d172f862d	4a747f88f0d43cd9c29fb721ebcbf3c1
N/A	%temp%\bkqmyosa.bat	N/A	1b6c96ba61e5dc6deda3733c8784f996
N/A	%temp%\file.vbs	a7125680be3bec73ddf1b4825579de7a	4afb5c4527091738faf9cd4addf9d34e
N/A	%temp%\twcccwyw.bat	c97fa1968305731cbf530b17bd274d9c	bae1095f340720d965898063fede1273

FortiSandbox



Analysis Details

WINXPVM1

Captured Packets | Tracer Package | Tracer Log

Behavior Chronology Chart

- Suspicious Behaviors (7)
- Static Analysis (1)
- Files Created (491)
- Files Deleted (479)
- Files Modified (144)
- Launched Processes
- Registry Changes (7
- Network Behaviors
- Behaviors In Sequer

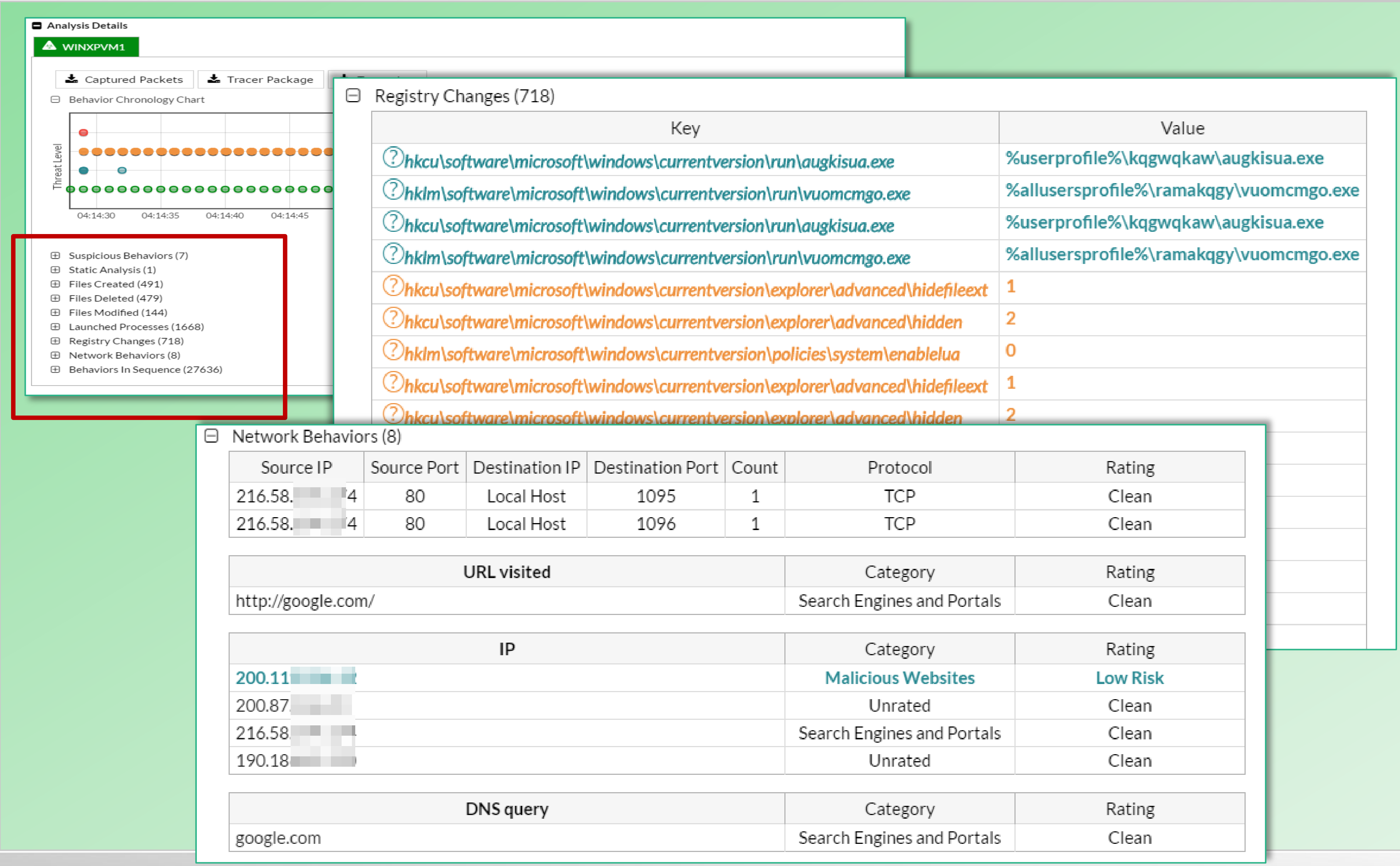
Files Deleted (479)

Virus	Path		
N/A	%temp%\rkescukk.bat		
N/A	%temp%\bkqmyosa.bat		
N/A	%temp%\mqqaamqu.bat		
N/A	%temp%\takkacyo.bat		
N/A	%temp%\twcccwyw.bat		
N/A	%temp%\sayeyoyc.bat		
N/A	%temp%\icemcauk.bat		
N/A	%temp%\huiayayi.bat		
N/A	%temp%\yucseykk.bat		
N/A	%temp%\hgmosais.bat	[illegible]	[illegible]
N/A	%temp%\magoyioi.bat	fde7e10d7f3adb07263811e66dac6778	8332150f8bd699dae4148fcdf2513972
N/A	%temp%\eyqcqmcy.bat	625f75e1b2d1bd7e8886a5db73d8513d	bae1095f340720d965898063fede1273
N/A	%temp%\lseymkom.bat	441eb8672954a73112bb8489cb3627fa	7f628193beaa508e862e89f22b9eb119
N/A	%temp%\zuyyyokw.bat	b6a2b347409f23c4d0afd9140bf20a32	bae1095f340720d965898063fede1273
N/A	%temp%\jocugioc.bat	a1929b095b4d291bc1a9a9ce9364a710	107f23422da3fa8595de8f1dcc51cd48
N/A	%temp%\owugguqc.bat	1bb6bb56261b214139d85494b01d3fb4	bae1095f340720d965898063fede1273

Launched Processes (1668)

Process
"C:\Documents and Settings\Administrator\kQgwQkAw\augkIsUA.exe"
"C:\Documents and Settings\All Users\rAMAkQgY\VuoMcMgo.exe"
cmd /c "C:\work\3223453414114943377"
C:\work\3223453414114943377
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /f /v HideFileExt /t REG_DWORD /d 1
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /f /v Hidden /t REG_DWORD /d 2
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLUA /d 0 /t REG_DWORD /f
cmd /c ""C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\TAkkAcYo.bat" "C:\work\3223453414114943377.exe""
cmd /c "C:\work\3223453414114943377"
cscript C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp/file.vbs
C:\work\3223453414114943377
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /f /v HideFileExt /t REG_DWORD /d 1
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /f /v Hidden /t REG_DWORD /d 2
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLUA /d 0 /t REG_DWORD /f
cmd /c ""C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\TWcccwYw.bat" "C:\work\3223453414114943377.exe""
cscript C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp/file.vbs
cmd /c "C:\work\3223453414114943377"
C:\work\3223453414114943377
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /f /v HideFileExt /t REG_DWORD /d 1
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /f /v Hidden /t REG_DWORD /d 2
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLUA /d 0 /t REG_DWORD /f
cmd /c ""C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\sAYEYoYc.bat" "C:\work\3223453414114943377.exe""
cscript C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp/file.vbs

FortiSandbox



Analysis Details

WINXPVM1

⬇ Captured Packets ⬇ Tracer Package

⊟ Behavior Chronology Chart

- ⊞ Suspicious Behaviors (7)
- ⊞ Static Analysis (1)
- ⊞ Files Created (491)
- ⊞ Files Deleted (479)
- ⊞ Files Modified (144)
- ⊞ Launched Processes (1668)
- ⊞ Registry Changes (718)
- ⊞ Network Behaviors (8)
- ⊞ Behaviors In Sequence (27636)

⊟ Registry Changes (718)

Key	Value
hkcu\software\microsoft\windows\currentversion\run\augkisua.exe	%userprofile%\kqgwqkaw\augkisua.exe
hklm\software\microsoft\windows\currentversion\run\vuomcmgo.exe	%allusersprofile%\ramakqgy\vuomcmgo.exe
hkcu\software\microsoft\windows\currentversion\run\augkisua.exe	%userprofile%\kqgwqkaw\augkisua.exe
hklm\software\microsoft\windows\currentversion\run\vuomcmgo.exe	%allusersprofile%\ramakqgy\vuomcmgo.exe
hkcu\software\microsoft\windows\currentversion\explorer\advanced\hidefileext	1
hkcu\software\microsoft\windows\currentversion\explorer\advanced\hidden	2
hklm\software\microsoft\windows\currentversion\policies\system\enablelua	0
hkcu\software\microsoft\windows\currentversion\explorer\advanced\hidefileext	1
hkcu\software\microsoft\windows\currentversion\explorer\advanced\hidden	2

⊟ Network Behaviors (8)

Source IP	Source Port	Destination IP	Destination Port	Count	Protocol	Rating
216.58.[redacted]4	80	Local Host	1095	1	TCP	Clean
216.58.[redacted]4	80	Local Host	1096	1	TCP	Clean

URL visited	Category	Rating
http://google.com/	Search Engines and Portals	Clean

IP	Category	Rating
200.11[redacted]	Malicious Websites	Low Risk
200.87[redacted]	Unrated	Clean
216.58[redacted]	Search Engines and Portals	Clean
190.18[redacted]	Unrated	Clean

DNS query	Category	Rating
google.com	Search Engines and Portals	Clean

Wrap Up

- **For reversing:**
  - ✓ Set a breakpoint at the end of metamorphic algorithm
  - ✓ Copy the decrypted code from memory

- **For detection:**
  - ✓ Get patterns from the decrypted code

- **For cleaning:**
  - ✓ Remove the entries from the registry keys
  - ✓ Extract the host file
  - ✓ Delete all malicious dropped files

Merci!

FORTINET®