Fighting Metamorphism Using Deep Learning With Fourier

Oct, 2016

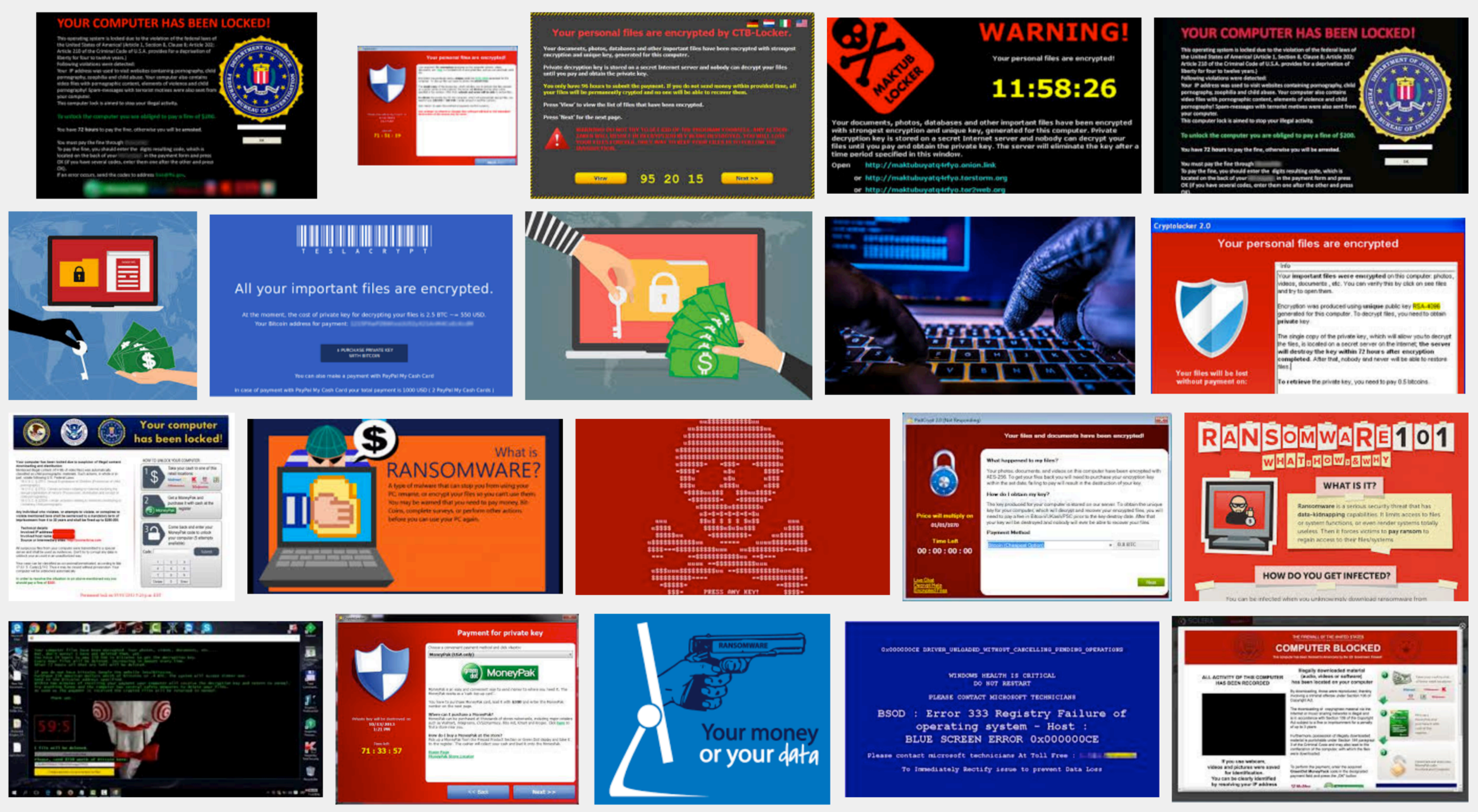
Sean Park

Senior Malware Scientist ,Trend Micro

spark@trendmicro.com

RUXCON

Ransomware

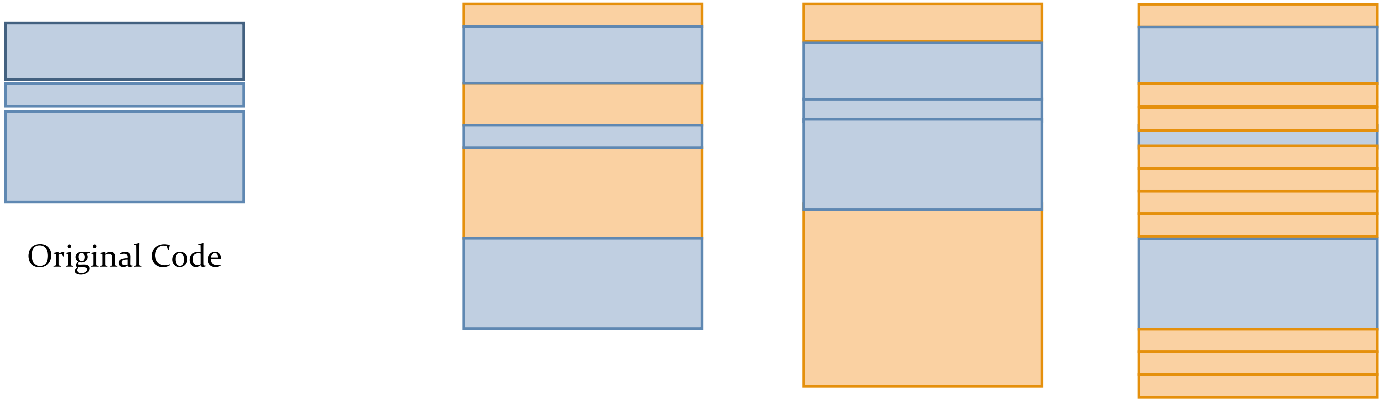

YOUR COMPUTER HAS BEEN LOCKED!
Your personal files are encrypted
Your personal files are encrypted by CTB-Locker.
View
95 20 15
Next >>
MAKTUB LOCKER
WARNING!
Your personal files are encrypted!
11:58:26
YOUR COMPUTER HAS BEEN LOCKED!
TESLACRYPT
All your important files are encrypted.
Cryptolocker 2.0
Your personal files are encrypted
Your files will be lost without payment on:
Your computer has been locked!
What is RANSOMWARE?
PRESS ANY KEY!
Your files and documents have been encrypted!
RANSOMWARE 101
WHAT, HOW & WHY
WHAT IS IT?
HOW DO YOU GET INFECTED?
Payment for private key
MoneyPak
71 : 33 : 57
<< Back
Next >>
RANSOMWARE
Your money or your data
BSOD : Error 333 Registry Failure of operating system - Host : BLUE SCREEN ERROR 0x000000CE
COMPUTER BLOCKED

Courtesy of google images

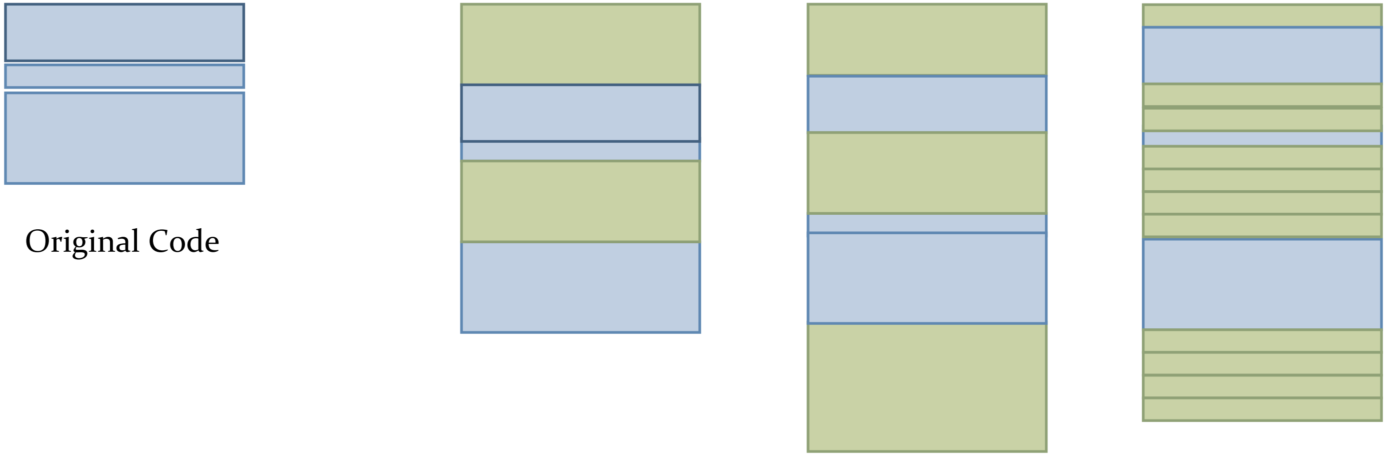
Problems

Metamorphism

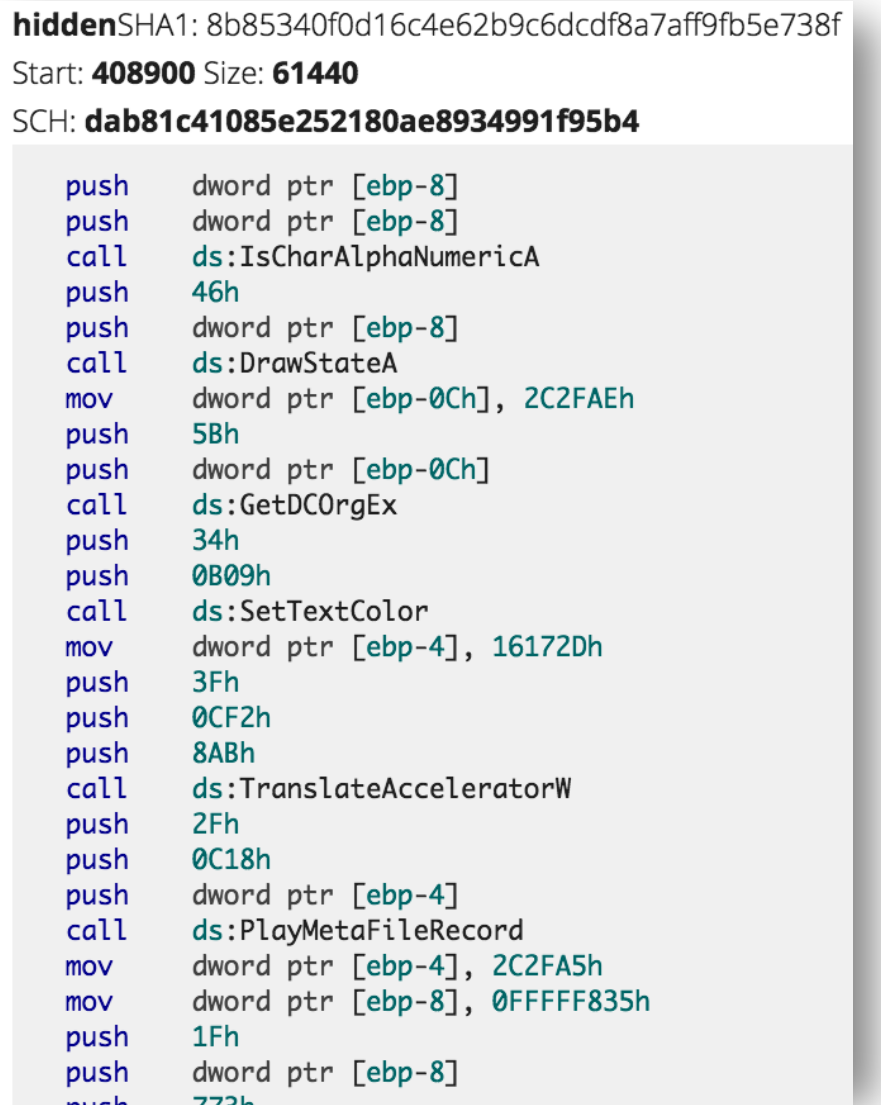
- Metamorphic template with parameters #1



- Metamorphic template with parameters #2



Metamorphism: push push call

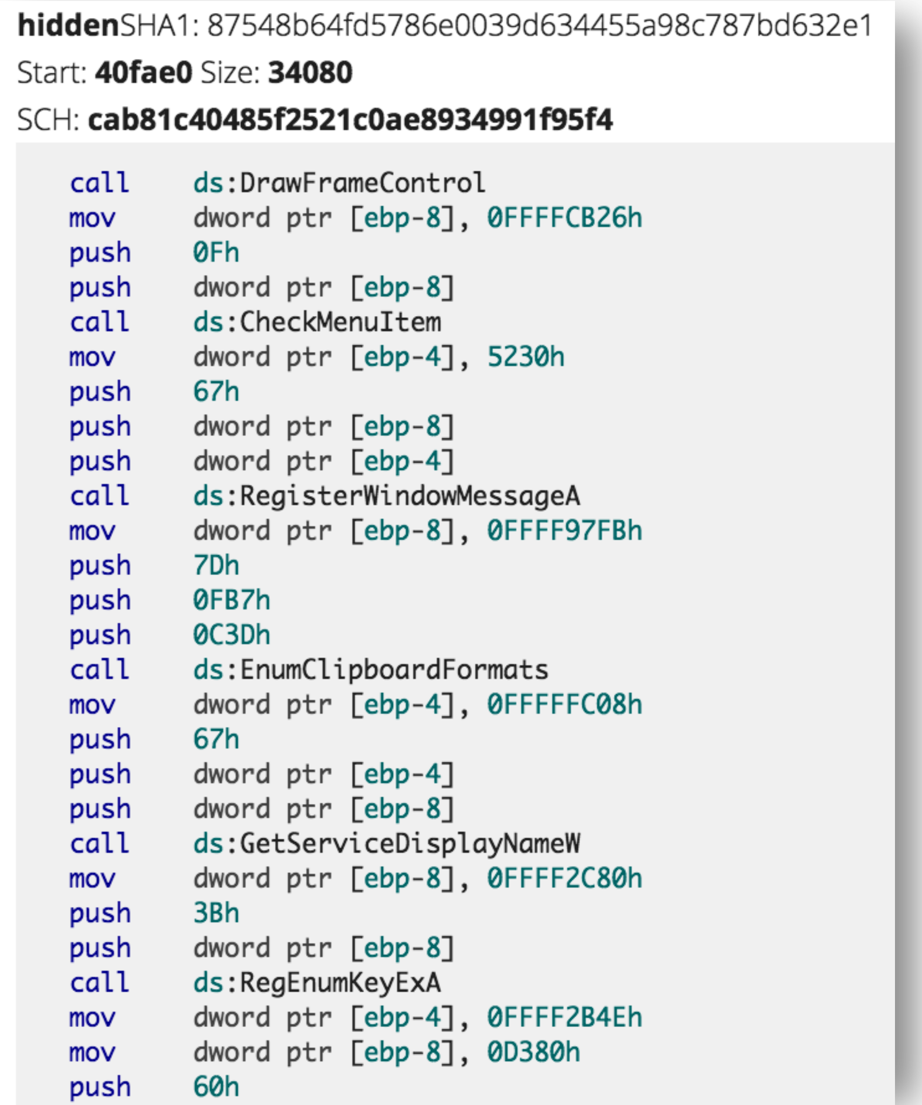


hiddenSHA1: 8b85340f0d16c4e62b9c6dcdf8a7aff9fb5e738f

Start: **408900** Size: **61440**

SCH: **dab81c41085e252180ae8934991f95b4**

```
push    dword ptr [ebp-8]
push    dword ptr [ebp-8]
call    ds:IsCharAlphaNumericA
push    46h
push    dword ptr [ebp-8]
call    ds:DrawStateA
mov     dword ptr [ebp-0Ch], 2C2FAEh
push    5Bh
push    dword ptr [ebp-0Ch]
call    ds:GetDCOrgEx
push    34h
push    0B09h
call    ds:SetTextColor
mov     dword ptr [ebp-4], 16172Dh
push    3Fh
push    0CF2h
push    8ABh
call    ds:TranslateAcceleratorW
push    2Fh
push    0C18h
push    dword ptr [ebp-4]
call    ds:PlayMetaFileRecord
mov     dword ptr [ebp-4], 2C2FA5h
mov     dword ptr [ebp-8], 0FFFFF835h
push    1Fh
push    dword ptr [ebp-8]
push    773h
```



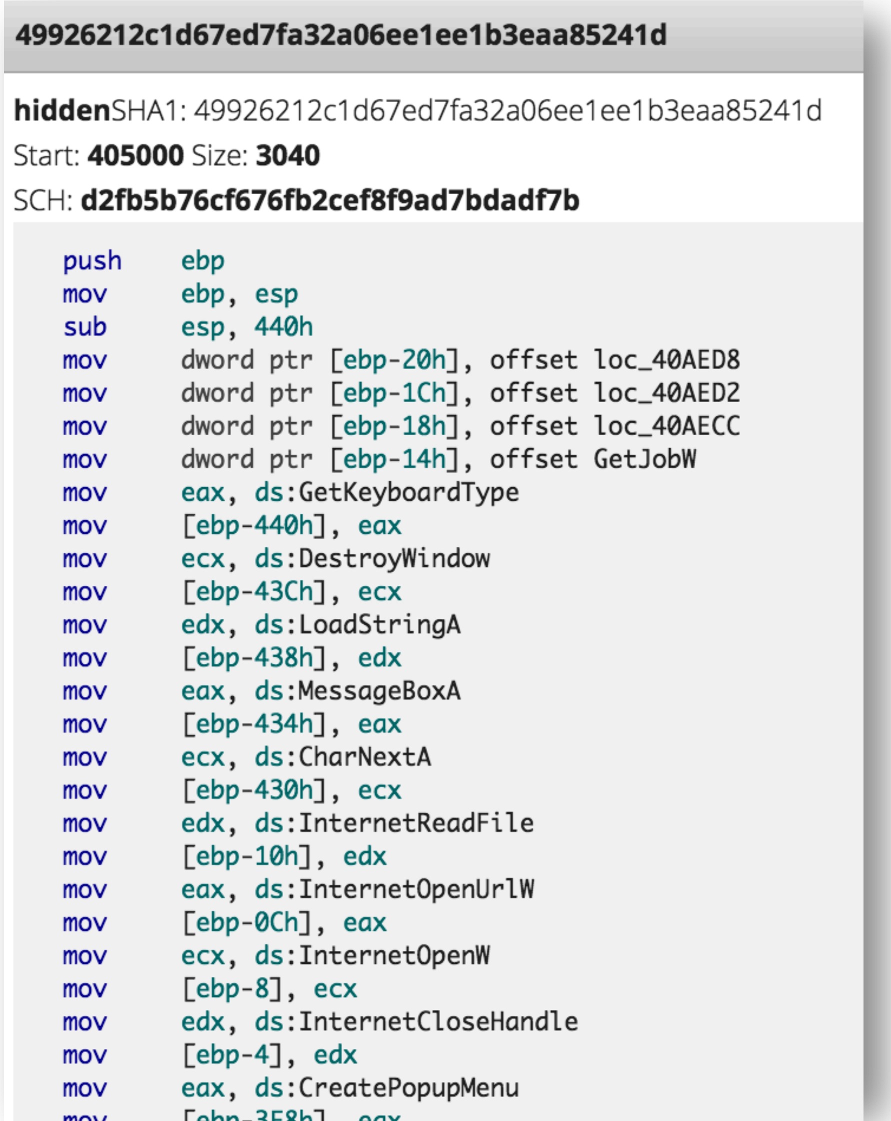
hiddenSHA1: 87548b64fd5786e0039d634455a98c787bd632e1

Start: **40fae0** Size: **34080**

SCH: **cab81c40485f2521c0ae8934991f95f4**

```
call    ds:DrawFrameControl
mov     dword ptr [ebp-8], 0FFFFCB26h
push    0Fh
push    dword ptr [ebp-8]
call    ds:CheckMenuItem
mov     dword ptr [ebp-4], 5230h
push    67h
push    dword ptr [ebp-8]
push    dword ptr [ebp-4]
call    ds:RegisterWindowMessageA
mov     dword ptr [ebp-8], 0FFFF97FBh
push    7Dh
push    0FB7h
push    0C3Dh
call    ds:EnumClipboardFormats
mov     dword ptr [ebp-4], 0FFFFFC08h
push    67h
push    dword ptr [ebp-4]
push    dword ptr [ebp-8]
call    ds:GetServiceDisplayNameW
mov     dword ptr [ebp-8], 0FFFF2C80h
push    3Bh
push    dword ptr [ebp-8]
call    ds:RegEnumKeyExA
mov     dword ptr [ebp-4], 0FFFF2B4Eh
mov     dword ptr [ebp-8], 0D380h
push    60h
```

Metamorphism: mov sequence



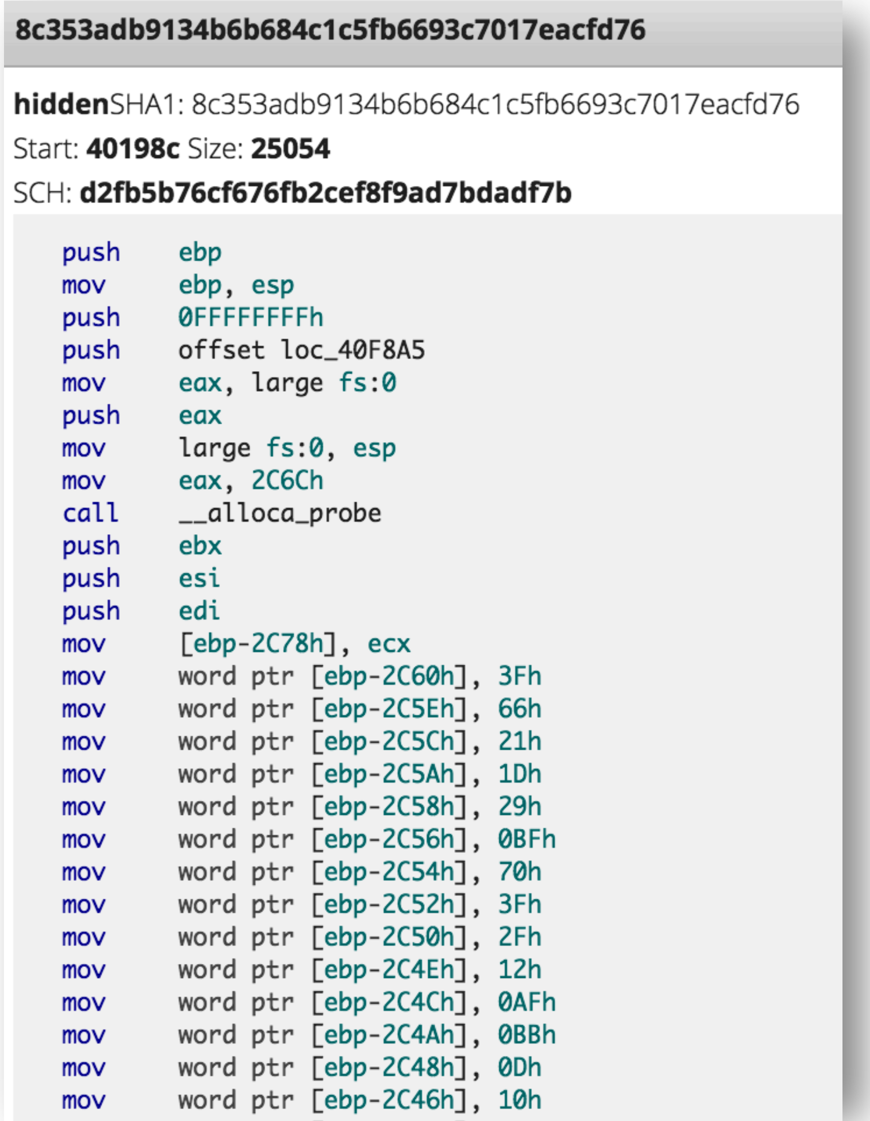
49926212c1d67ed7fa32a06ee1ee1b3eaa85241d

hiddenSHA1: 49926212c1d67ed7fa32a06ee1ee1b3eaa85241d

Start: **405000** Size: **3040**

SCH: **d2fb5b76cf676fb2cef8f9ad7bdadf7b**

```
push    ebp
mov     ebp, esp
sub     esp, 440h
mov     dword ptr [ebp-20h], offset loc_40AED8
mov     dword ptr [ebp-1Ch], offset loc_40AED2
mov     dword ptr [ebp-18h], offset loc_40AECC
mov     dword ptr [ebp-14h], offset GetJobW
mov     eax, ds:GetKeyboardType
mov     [ebp-440h], eax
mov     ecx, ds:DestroyWindow
mov     [ebp-43Ch], ecx
mov     edx, ds:LoadStringA
mov     [ebp-438h], edx
mov     eax, ds:MessageBoxA
mov     [ebp-434h], eax
mov     ecx, ds:CharNextA
mov     [ebp-430h], ecx
mov     edx, ds:InternetReadFile
mov     [ebp-10h], edx
mov     eax, ds:InternetOpenUrlW
mov     [ebp-0Ch], eax
mov     ecx, ds:InternetOpenW
mov     [ebp-8], ecx
mov     edx, ds:InternetCloseHandle
mov     [ebp-4], edx
mov     eax, ds:CreatePopupMenu
mov     [ebp-358h], eax
```



8c353adb9134b6b684c1c5fb6693c7017eacfd76

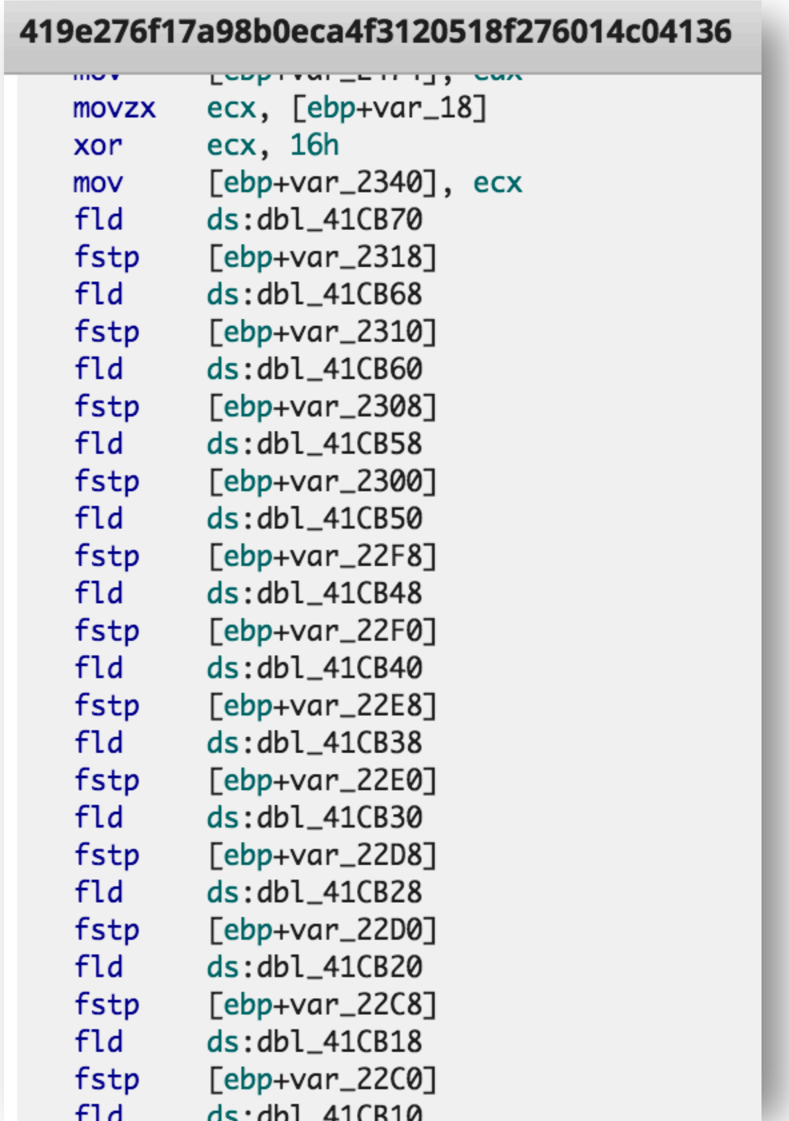
hiddenSHA1: 8c353adb9134b6b684c1c5fb6693c7017eacfd76

Start: **40198c** Size: **25054**

SCH: **d2fb5b76cf676fb2cef8f9ad7bdadf7b**

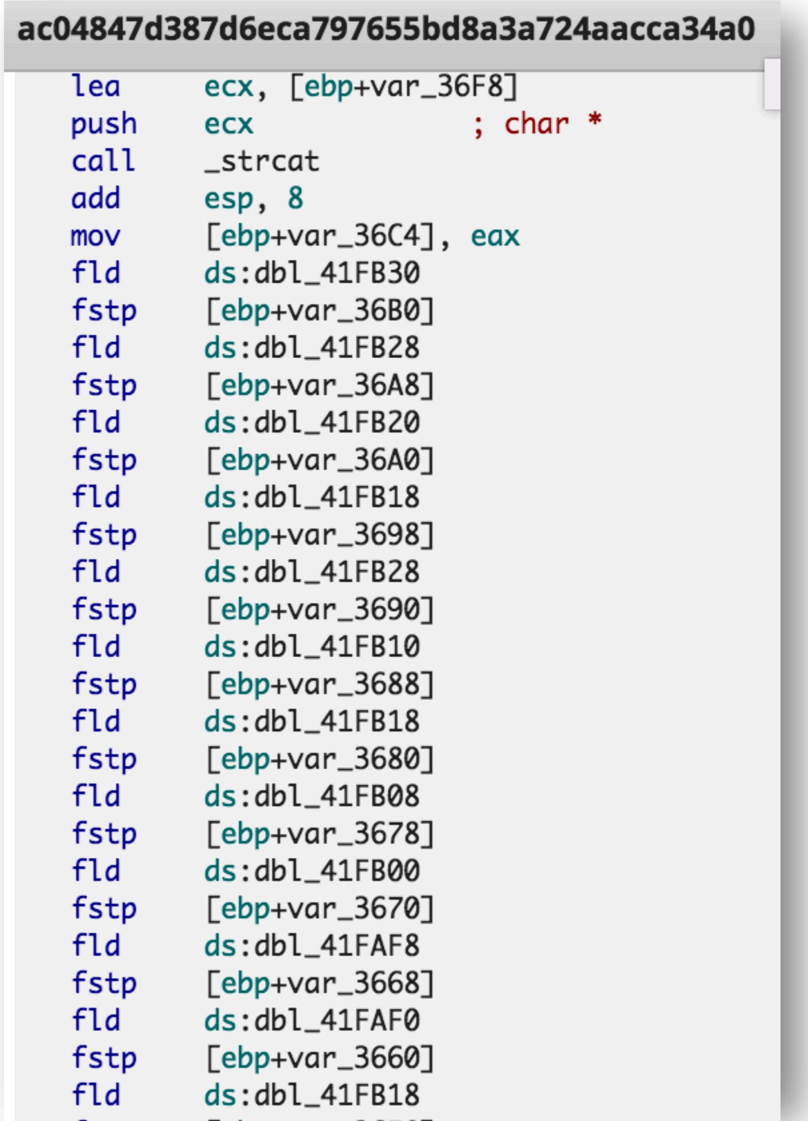
```
push    ebp
mov     ebp, esp
push    0FFFFFFFFh
push    offset loc_40F8A5
mov     eax, large fs:0
push    eax
mov     large fs:0, esp
mov     eax, 2C6Ch
call    __alloca_probe
push    ebx
push    esi
push    edi
mov     [ebp-2C78h], ecx
mov     word ptr [ebp-2C60h], 3Fh
mov     word ptr [ebp-2C5Eh], 66h
mov     word ptr [ebp-2C5Ch], 21h
mov     word ptr [ebp-2C5Ah], 1Dh
mov     word ptr [ebp-2C58h], 29h
mov     word ptr [ebp-2C56h], 0BFh
mov     word ptr [ebp-2C54h], 70h
mov     word ptr [ebp-2C52h], 3Fh
mov     word ptr [ebp-2C50h], 2Fh
mov     word ptr [ebp-2C4Eh], 12h
mov     word ptr [ebp-2C4Ch], 0AFh
mov     word ptr [ebp-2C4Ah], 0BBh
mov     word ptr [ebp-2C48h], 0Dh
mov     word ptr [ebp-2C46h], 10h
```

Metamorphism: fld/fstp sequence



419e276f17a98b0eca4f3120518f276014c04136

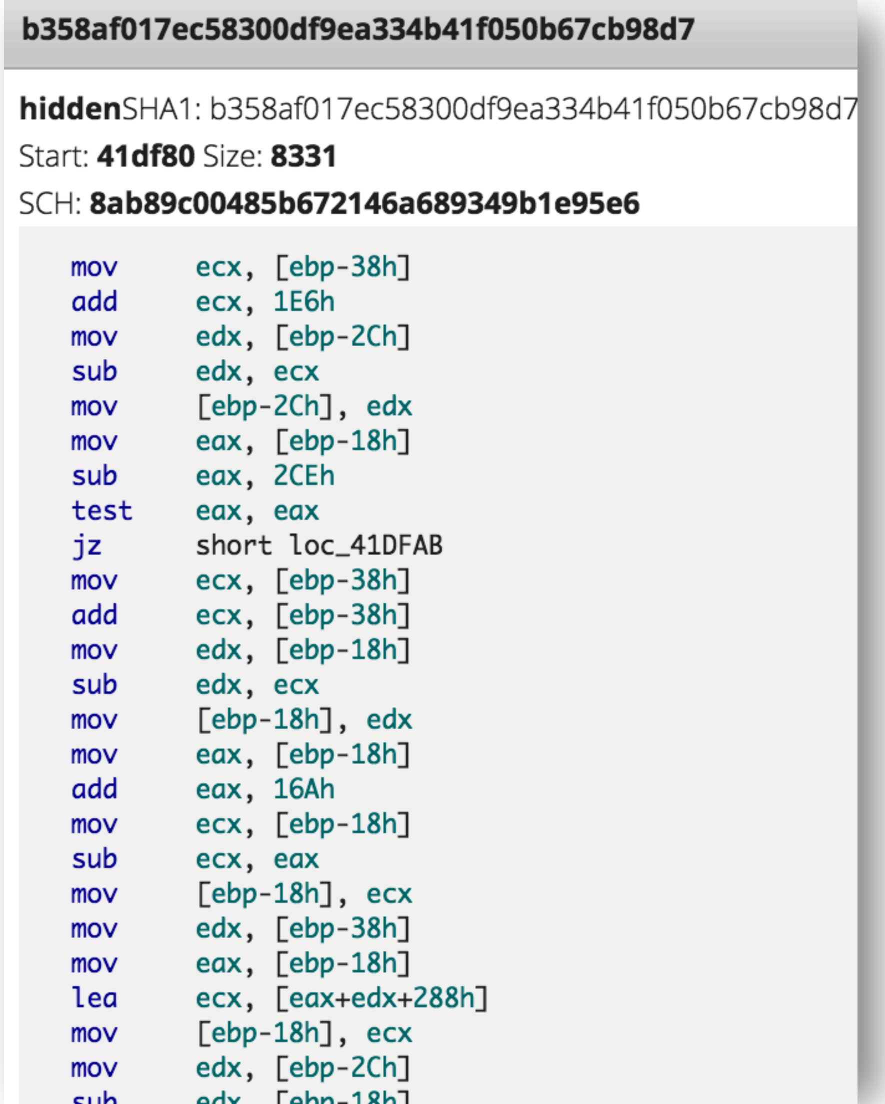
```
mov     [ebp+var_2174], eax
movzx   ecx, [ebp+var_18]
xor     ecx, 16h
mov     [ebp+var_2340], ecx
fld     ds:dbl_41CB70
fstp    [ebp+var_2318]
fld     ds:dbl_41CB68
fstp    [ebp+var_2310]
fld     ds:dbl_41CB60
fstp    [ebp+var_2308]
fld     ds:dbl_41CB58
fstp    [ebp+var_2300]
fld     ds:dbl_41CB50
fstp    [ebp+var_22F8]
fld     ds:dbl_41CB48
fstp    [ebp+var_22F0]
fld     ds:dbl_41CB40
fstp    [ebp+var_22E8]
fld     ds:dbl_41CB38
fstp    [ebp+var_22E0]
fld     ds:dbl_41CB30
fstp    [ebp+var_22D8]
fld     ds:dbl_41CB28
fstp    [ebp+var_22D0]
fld     ds:dbl_41CB20
fstp    [ebp+var_22C8]
fld     ds:dbl_41CB18
fstp    [ebp+var_22C0]
fld     ds:dbl_41CB10
```



ac04847d387d6eca797655bd8a3a724aacca34a0

```
lea     ecx, [ebp+var_36F8]
push    ecx             ; char *
call    _strcat
add     esp, 8
mov     [ebp+var_36C4], eax
fld     ds:dbl_41FB30
fstp    [ebp+var_36B0]
fld     ds:dbl_41FB28
fstp    [ebp+var_36A8]
fld     ds:dbl_41FB20
fstp    [ebp+var_36A0]
fld     ds:dbl_41FB18
fstp    [ebp+var_3698]
fld     ds:dbl_41FB28
fstp    [ebp+var_3690]
fld     ds:dbl_41FB10
fstp    [ebp+var_3688]
fld     ds:dbl_41FB18
fstp    [ebp+var_3680]
fld     ds:dbl_41FB08
fstp    [ebp+var_3678]
fld     ds:dbl_41FB00
fstp    [ebp+var_3670]
fld     ds:dbl_41FAF8
fstp    [ebp+var_3668]
fld     ds:dbl_41FAF0
fstp    [ebp+var_3660]
fld     ds:dbl_41FB18
```

Metamorphism: add/sub mov



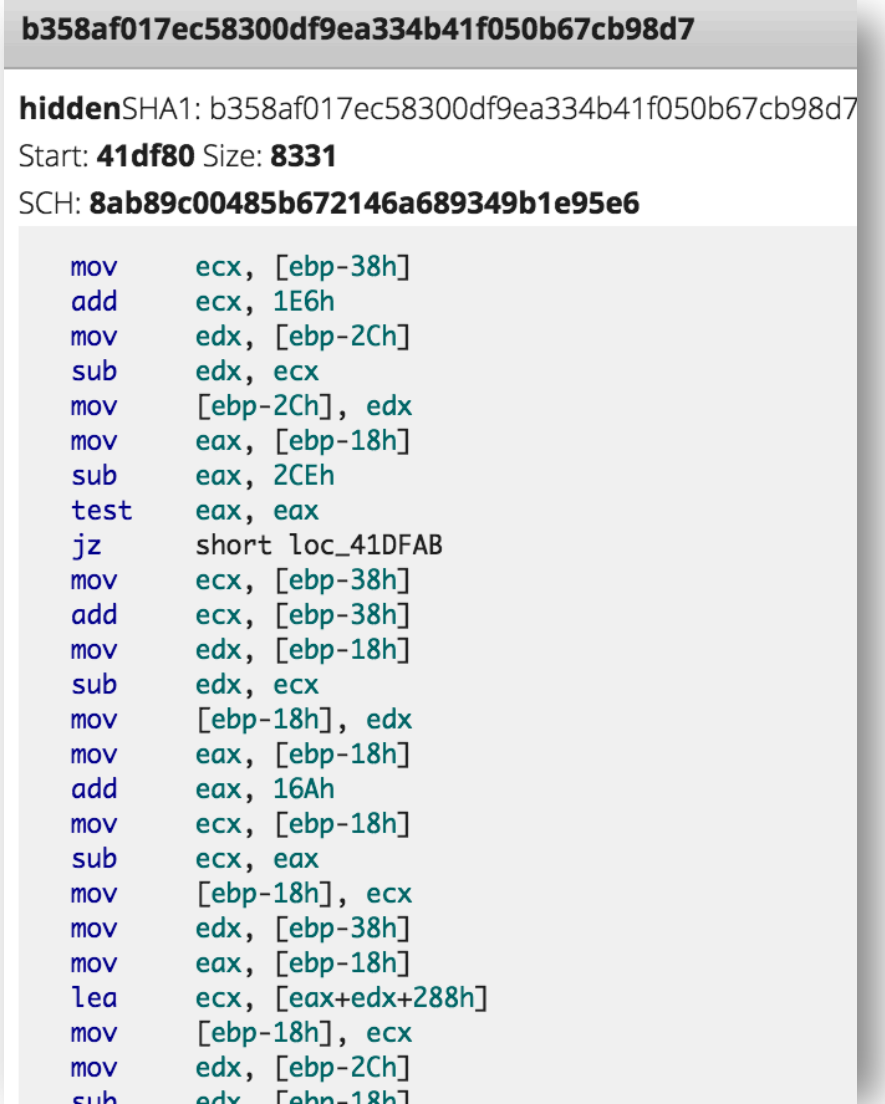
b358af017ec58300df9ea334b41f050b67cb98d7

hiddenSHA1: b358af017ec58300df9ea334b41f050b67cb98d7

Start: **41df80** Size: **8331**

SCH: **8ab89c00485b672146a689349b1e95e6**

```
mov     ecx, [ebp-38h]
add     ecx, 1E6h
mov     edx, [ebp-2Ch]
sub     edx, ecx
mov     [ebp-2Ch], edx
mov     eax, [ebp-18h]
sub     eax, 2CEh
test    eax, eax
jz      short loc_41DFAB
mov     ecx, [ebp-38h]
add     ecx, [ebp-38h]
mov     edx, [ebp-18h]
sub     edx, ecx
mov     [ebp-18h], edx
mov     eax, [ebp-18h]
add     eax, 16Ah
mov     ecx, [ebp-18h]
sub     ecx, eax
mov     [ebp-18h], ecx
mov     edx, [ebp-38h]
mov     eax, [ebp-18h]
lea     ecx, [eax+edx+288h]
mov     [ebp-18h], ecx
mov     edx, [ebp-2Ch]
sub     edx, [ebp-18h]
```



b358af017ec58300df9ea334b41f050b67cb98d7

hiddenSHA1: b358af017ec58300df9ea334b41f050b67cb98d7

Start: **41df80** Size: **8331**

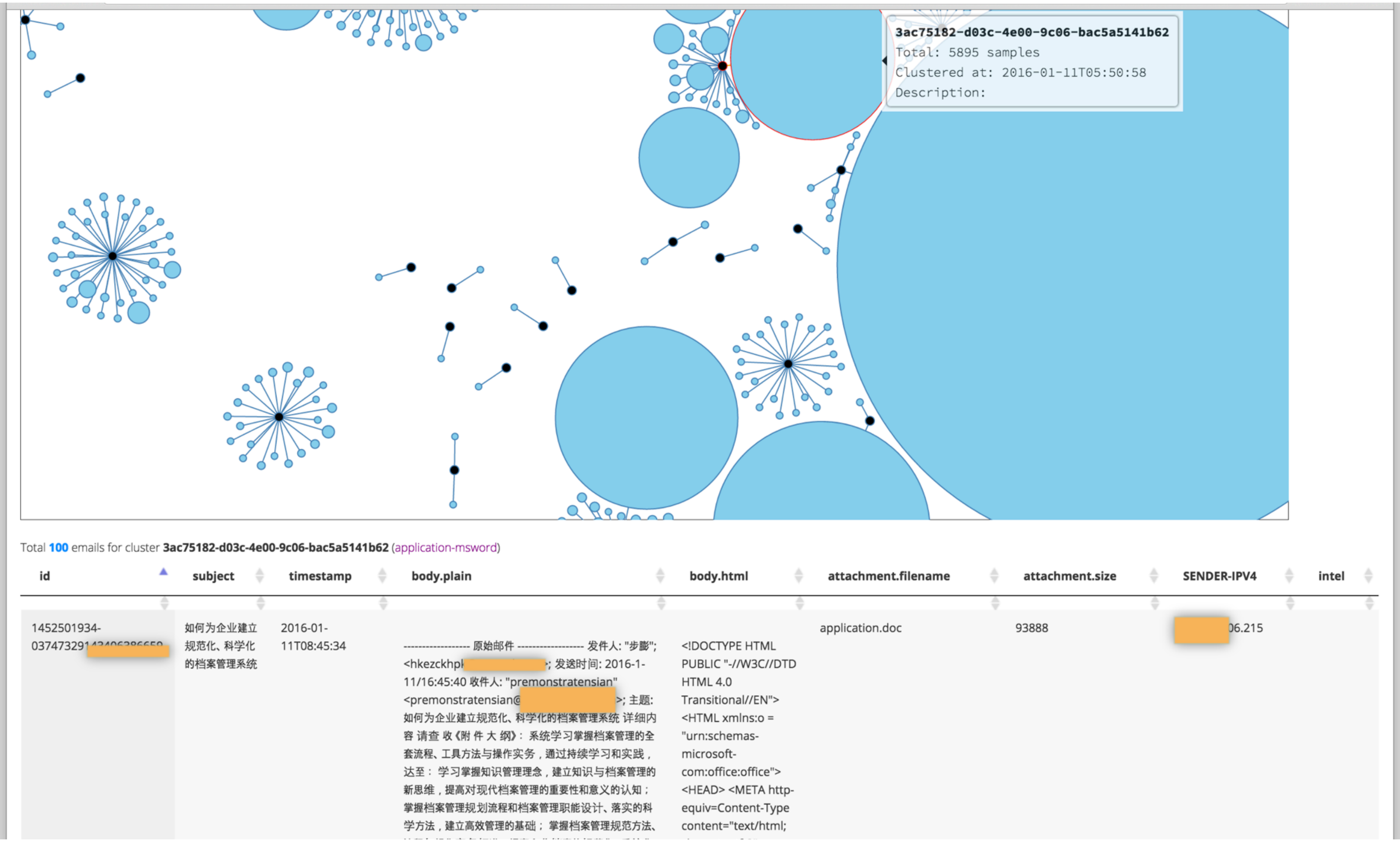
SCH: **8ab89c00485b672146a689349b1e95e6**

```
mov     ecx, [ebp-38h]
add     ecx, 1E6h
mov     edx, [ebp-2Ch]
sub     edx, ecx
mov     [ebp-2Ch], edx
mov     eax, [ebp-18h]
sub     eax, 2CEh
test    eax, eax
jz      short loc_41DFAB
mov     ecx, [ebp-38h]
add     ecx, [ebp-38h]
mov     edx, [ebp-18h]
sub     edx, ecx
mov     [ebp-18h], edx
mov     eax, [ebp-18h]
add     eax, 16Ah
mov     ecx, [ebp-18h]
sub     ecx, eax
mov     [ebp-18h], ecx
mov     edx, [ebp-38h]
mov     eax, [ebp-18h]
lea     ecx, [eax+edx+288h]
mov     [ebp-18h], ecx
mov     edx, [ebp-2Ch]
sub     edx, [ebp-18h]
```

Significance of Metamorphism Detection

- Malware mostly delivered through email outbreaks
  - An outbreak lasts days or a couple of weeks
- The same metamorphic template used during a campaign
  - Early deep learning will block entire campaign
- Sometimes the same metamorphic template used across several different campaigns due to the high dev cost
  - Early deep learning will block multiple campaigns

SLAM
: Unsupervised email clustering system



3ac75182-d03c-4e00-9c06-bac5a5141b62
Total: 5895 samples
Clustered at: 2016-01-11T05:50:58
Description:

Total 100 emails for cluster 3ac75182-d03c-4e00-9c06-bac5a5141b62 (application-msword)

id	subject	timestamp	body.plain	body.html	attachment.filename	attachment.size	SENDER-IPV4	intel
1452501934-0374732914[redacted]	如何为企业建立规范化、科学化的档案管理系统	2016-01-11T08:45:34	------------------ 原始邮件 ------------------ 发件人: "步膨"; <hkezckhpl[redacted]>; 发送时间: 2016-1-11/16:45:40 收件人: "premonstratensian" <premonstratensian@[redacted]>; 主题: 如何为企业建立规范化、科学化的档案管理系统 详细内容 请查 收《附 件 大 纲》: 系统学习掌握档案管理的全套流程、工具方法与操作实务，通过持续学习和实践，达至： 学习掌握知识管理理念，建立知识与档案管理的新思维，提高对现代档案管理的重要性和意义的认知； 掌握档案管理规划流程和档案管理职能设计、落实的科学方法，建立高效管理的基础； 掌握档案管理规范方法、	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"> <HTML xmlns:o = "urn:schemas-microsoft-com:office:office"> <HEAD> <META http-equiv=Content-Type content="text/html;	application.doc	93888	[redacted]06.215	

Difficulties

- Key challenges
  - Different SHA1 for each sample
  - Significantly different in lengths and locations
  - Easy to change template parameters resulting in superficially different patterns

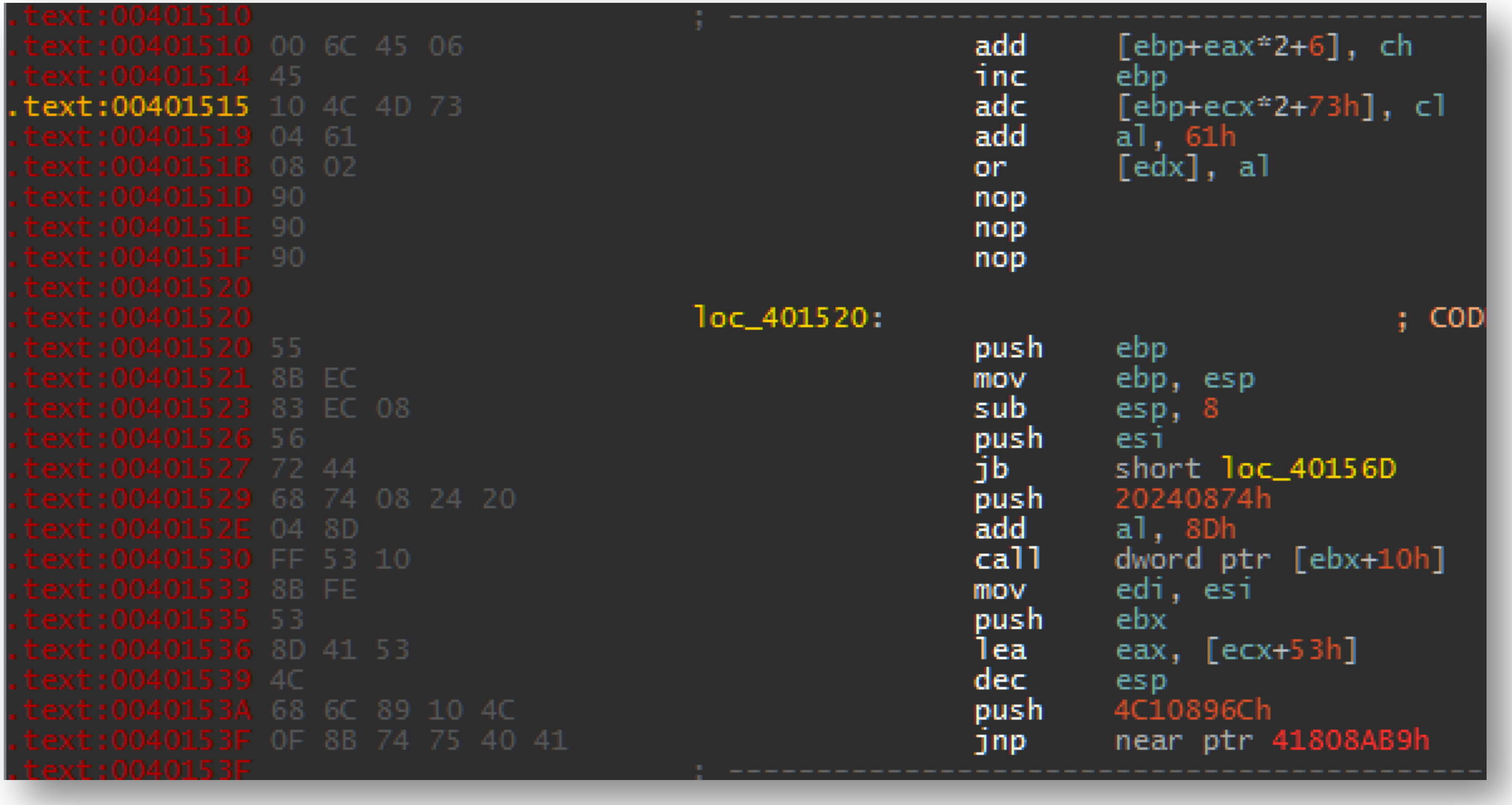
Failing Approaches

- Static signature
- Histograms/frequencies
- API call distribution
- Entropy
- Machine learning algorithms with binary classification

Solutions

Machine Instruction as Feature

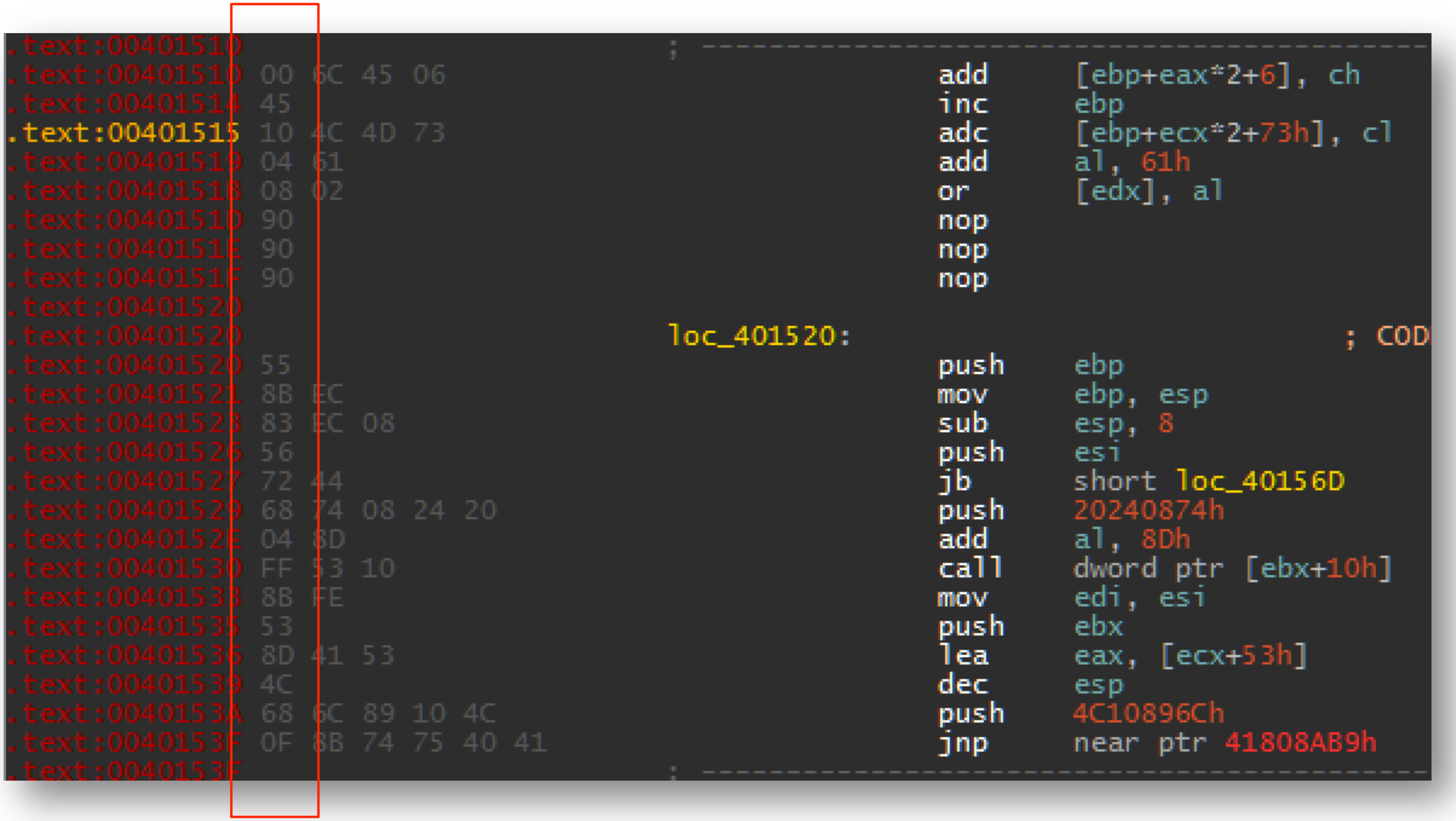
- All parsed functions and code blocks including those hidden



```
.text:00401510                                  ; ---------------------------------------
.text:00401510 00 6C 45 06                                      add     [ebp+eax*2+6], ch
.text:00401514 45                                               inc     ebp
.text:00401515 10 4C 4D 73                                      adc     [ebp+ecx*2+73h], cl
.text:00401519 04 61                                            add     al, 61h
.text:0040151B 08 02                                            or      [edx], al
.text:0040151D 90                                               nop
.text:0040151E 90                                               nop
.text:0040151F 90                                               nop
.text:00401520
.text:00401520                                  loc_401520:                                 ; COD
.text:00401520 55                                               push    ebp
.text:00401521 8B EC                                            mov     ebp, esp
.text:00401523 83 EC 08                                         sub     esp, 8
.text:00401526 56                                               push    esi
.text:00401527 72 44                                            jb      short loc_40156D
.text:00401529 68 74 08 24 20                                   push    20240874h
.text:0040152E 04 8D                                            add     al, 8Dh
.text:00401530 FF 53 10                                         call    dword ptr [ebx+10h]
.text:00401533 8B FE                                            mov     edi, esi
.text:00401535 53                                               push    ebx
.text:00401536 8D 41 53                                         lea     eax, [ecx+53h]
.text:00401539 4C                                               dec     esp
.text:0040153A 68 6C 89 10 4C                                   push    4C10896Ch
.text:0040153F 0F 8B 74 75 40 41                                jnp     near ptr 41808AB9h
.text:0040153F                                  ; ---------------------------------------
```

Machine Instruction as Feature

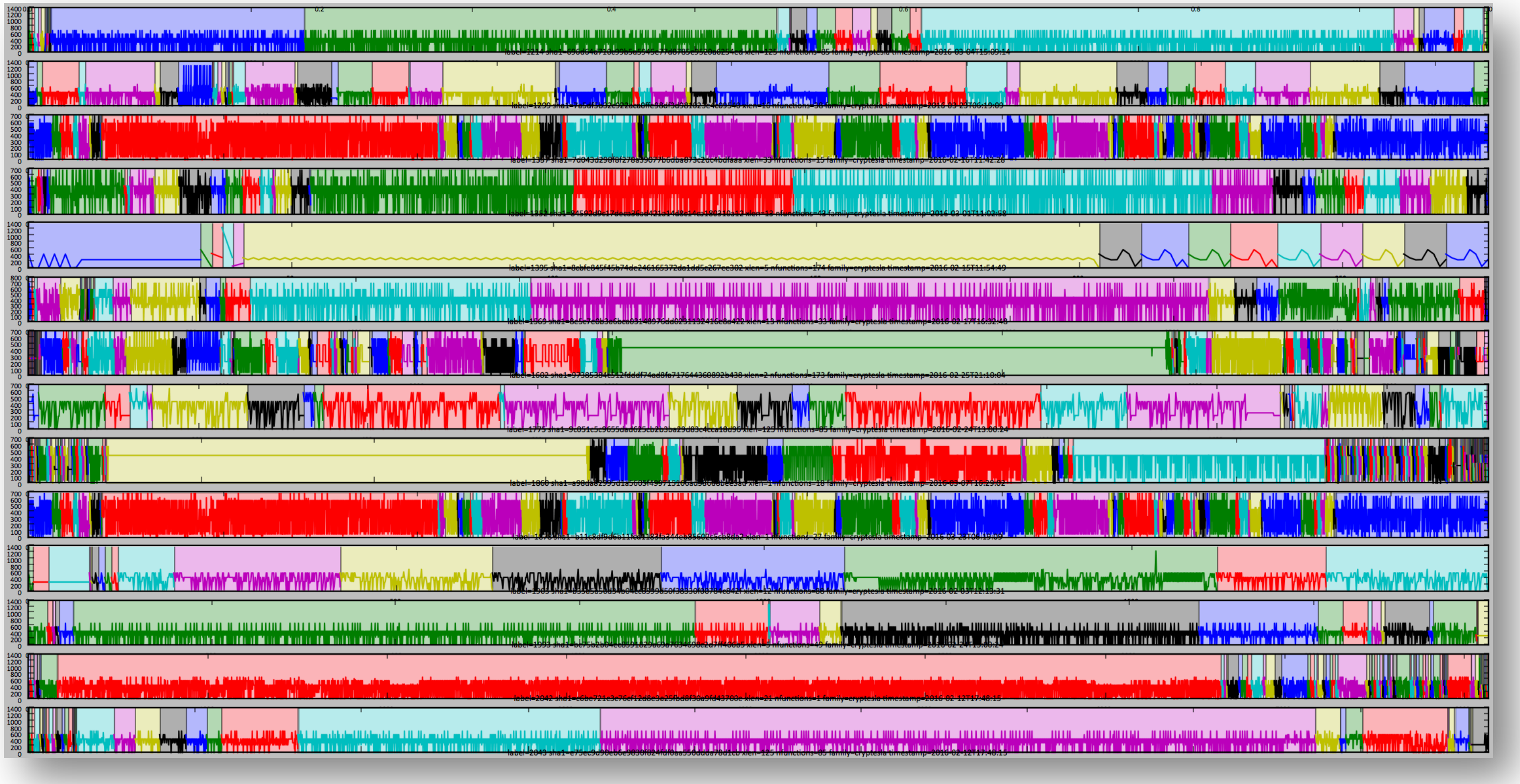
- Reduce noise by using opcode



```
.text:00401510                                  ; ----------------------------------------------
.text:00401510 00 6C 45 06                                add     [ebp+eax*2+6], ch
.text:00401514 45                                         inc     ebp
.text:00401515 10 4C 4D 73                                adc     [ebp+ecx*2+73h], cl
.text:00401519 04 61                                      add     al, 61h
.text:0040151B 08 02                                      or      [edx], al
.text:0040151D 90                                         nop
.text:0040151E 90                                         nop
.text:0040151F 90                                         nop
.text:00401520
.text:00401520                                  loc_401520:                           ; COD
.text:00401520 55                                         push    ebp
.text:00401521 8B EC                                      mov     ebp, esp
.text:00401523 83 EC 08                                   sub     esp, 8
.text:00401526 56                                         push    esi
.text:00401527 72 44                                      jb      short loc_40156D
.text:00401529 68 74 08 24 20                             push    20240874h
.text:0040152E 04 8D                                      add     al, 8Dh
.text:00401530 FF 53 10                                   call    dword ptr [ebx+10h]
.text:00401533 8B FE                                      mov     edi, esi
.text:00401535 53                                         push    ebx
.text:00401536 8D 41 53                                   lea     eax, [ecx+53h]
.text:00401539 4C                                         dec     esp
.text:0040153A 68 6C 89 10 4C                             push    4C10896Ch
.text:0040153F 0F 8B 74 75 40 41                          jnp     near ptr 41808AB9h
.text:0040153F                                  ; ----------------------------------------------
```

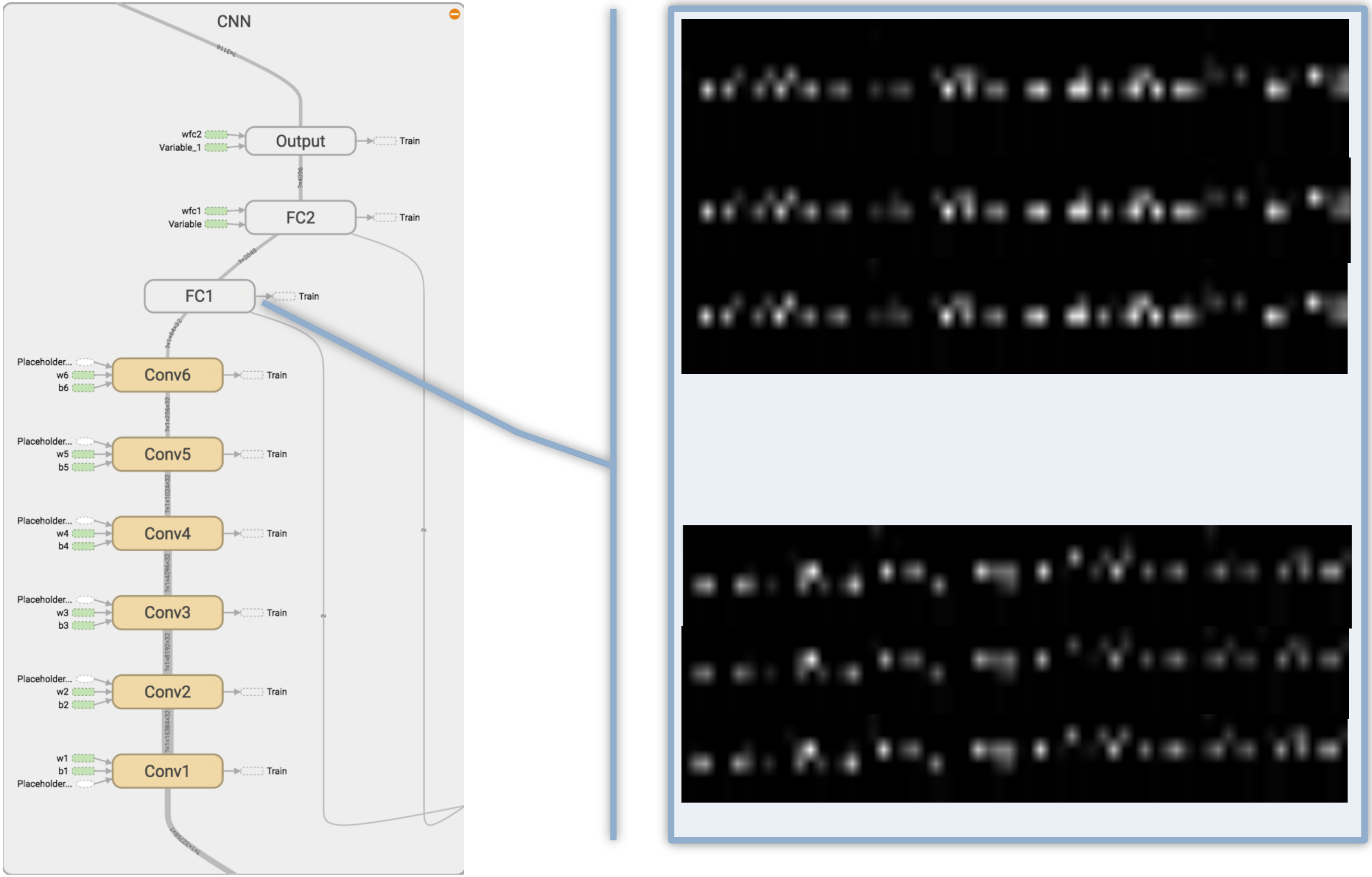
Recent Approaches Using Instructions

Spectrum of Instructions

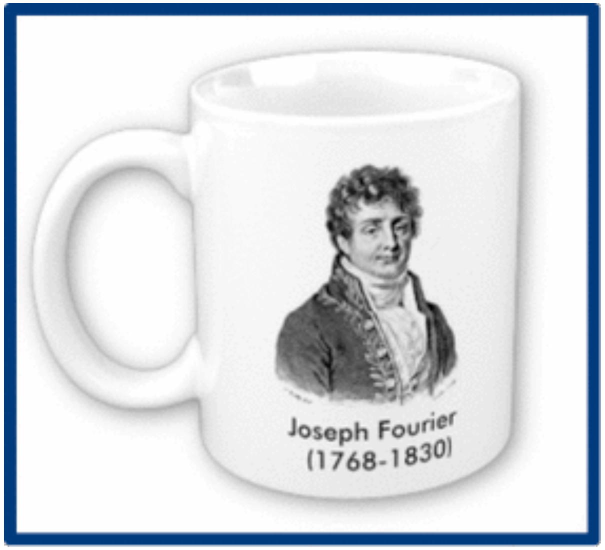


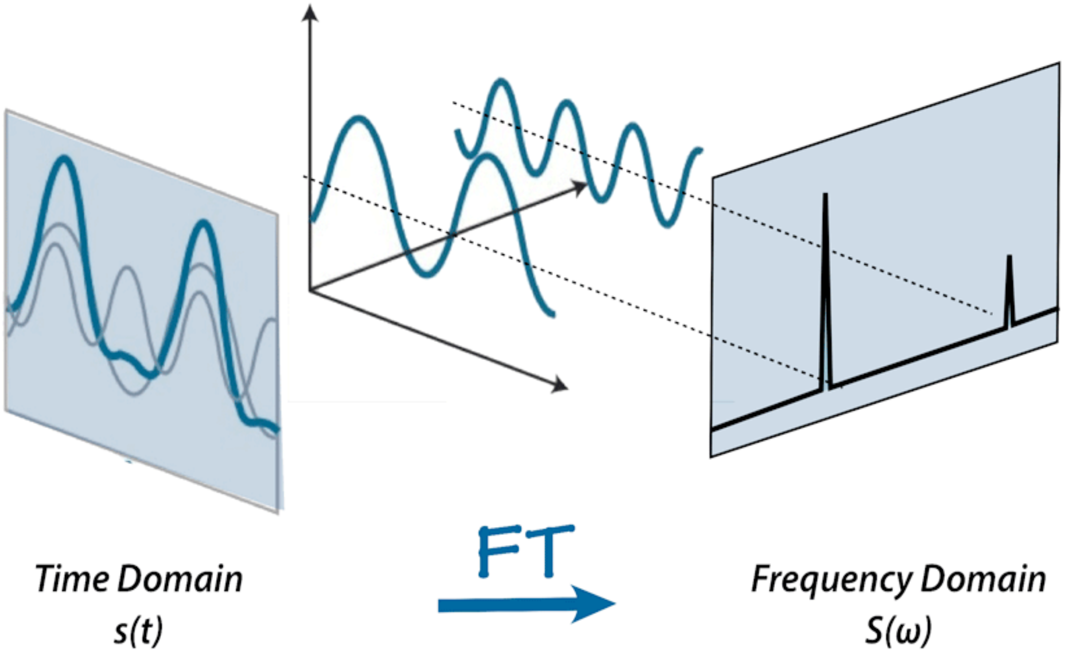
- Experiment shows it works to a degree.
- But, it is **unable** to characterise functions that possess the same metamorphism

CNN with instructions as feature



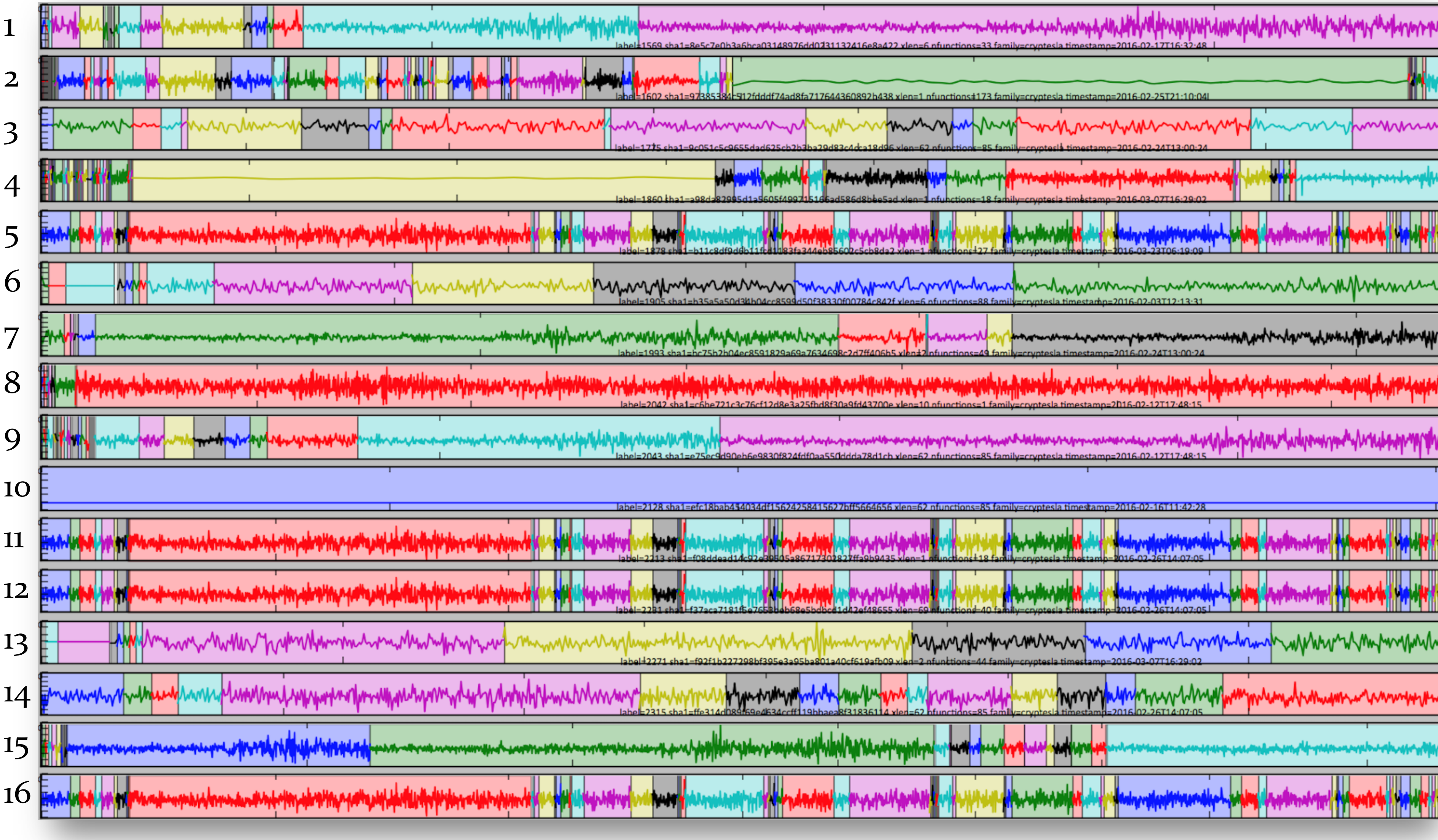
Fourier Transform



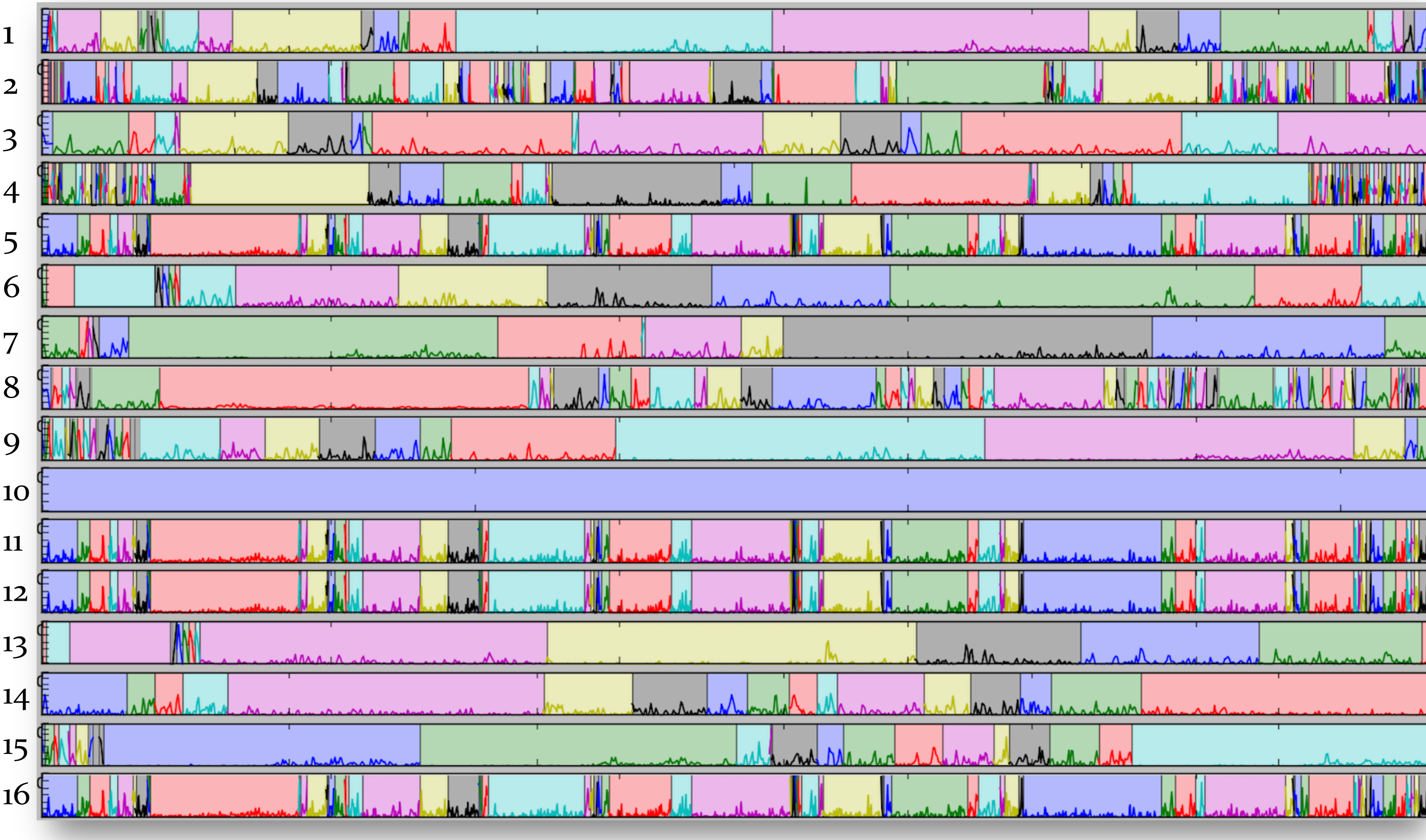


Courtesy of http://mriquestions.com/fourier-transform-ft.html

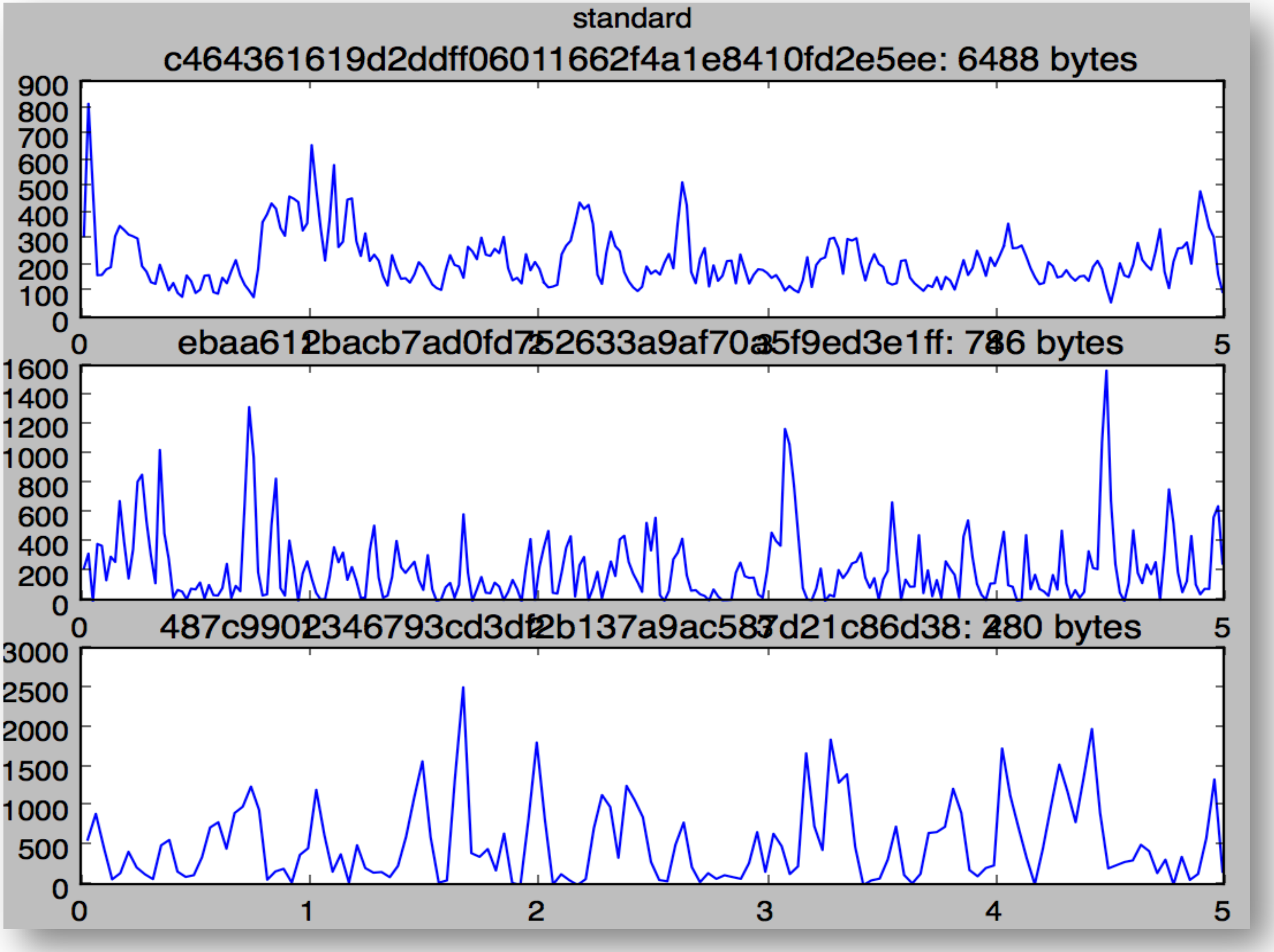
FFT As Feature – numpy.fft.fft



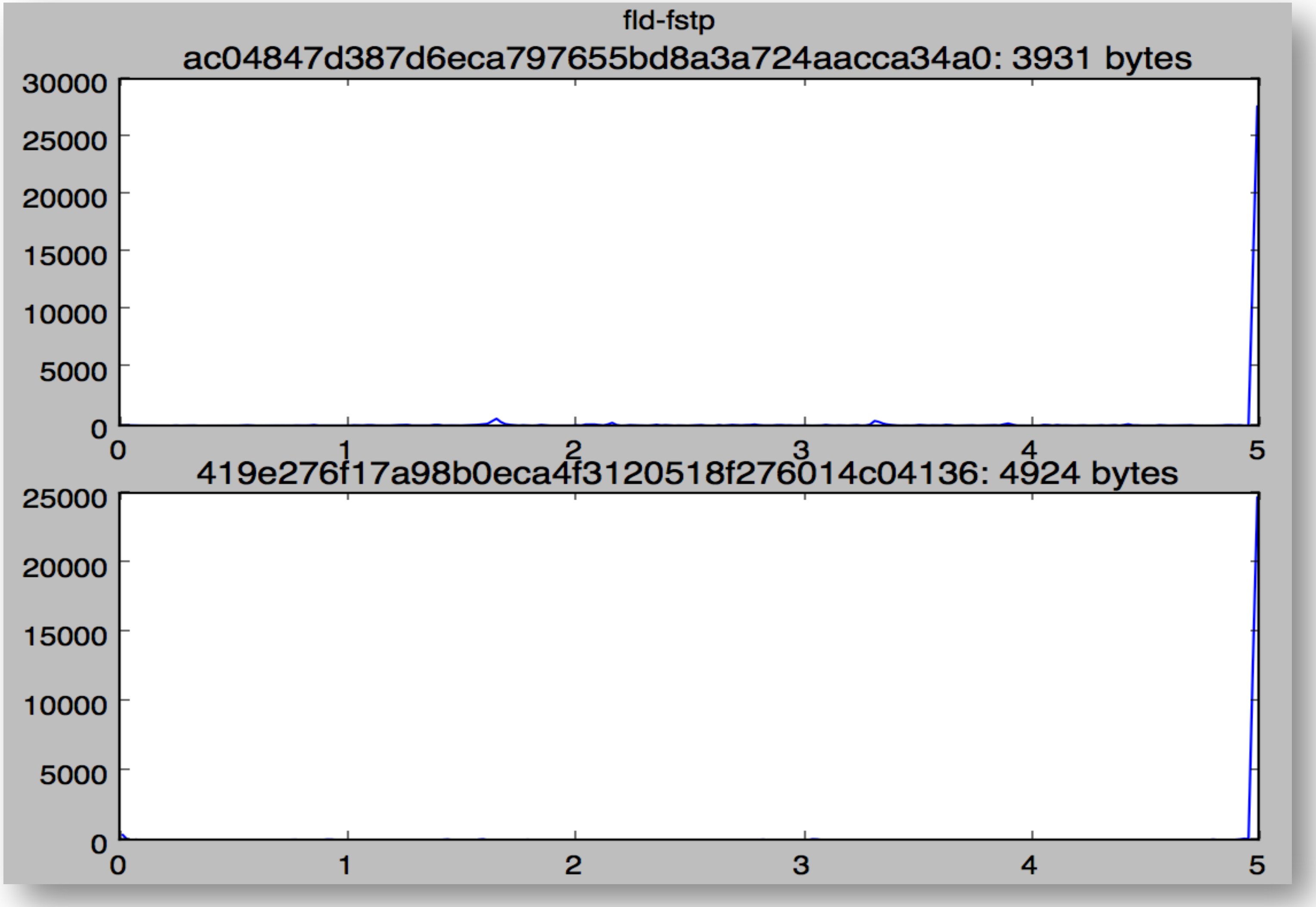
FFT As Feature – scipy.signal.welch



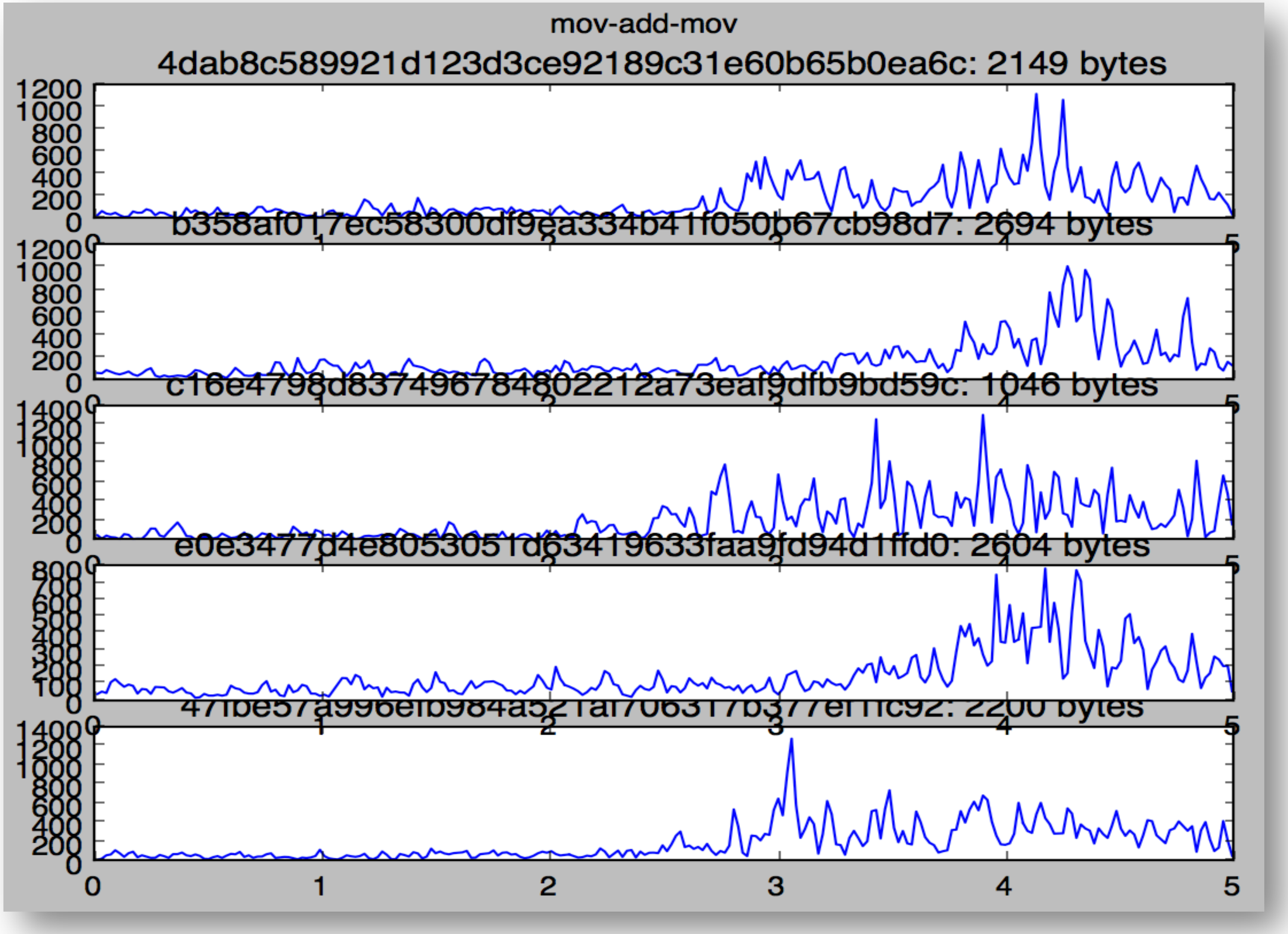
FFT
: Legitimate functions



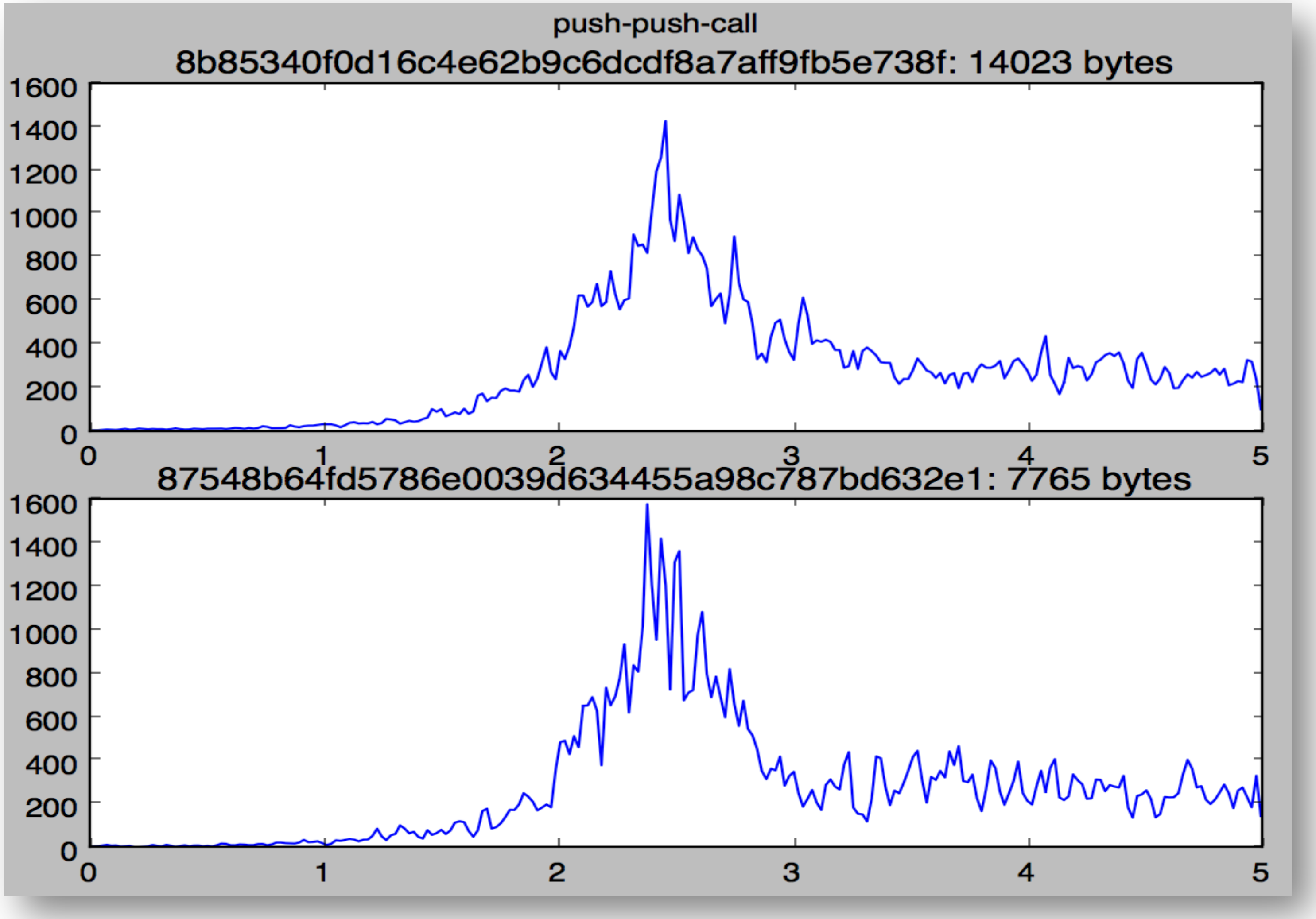
FFT
: fld-fstp metamorphism



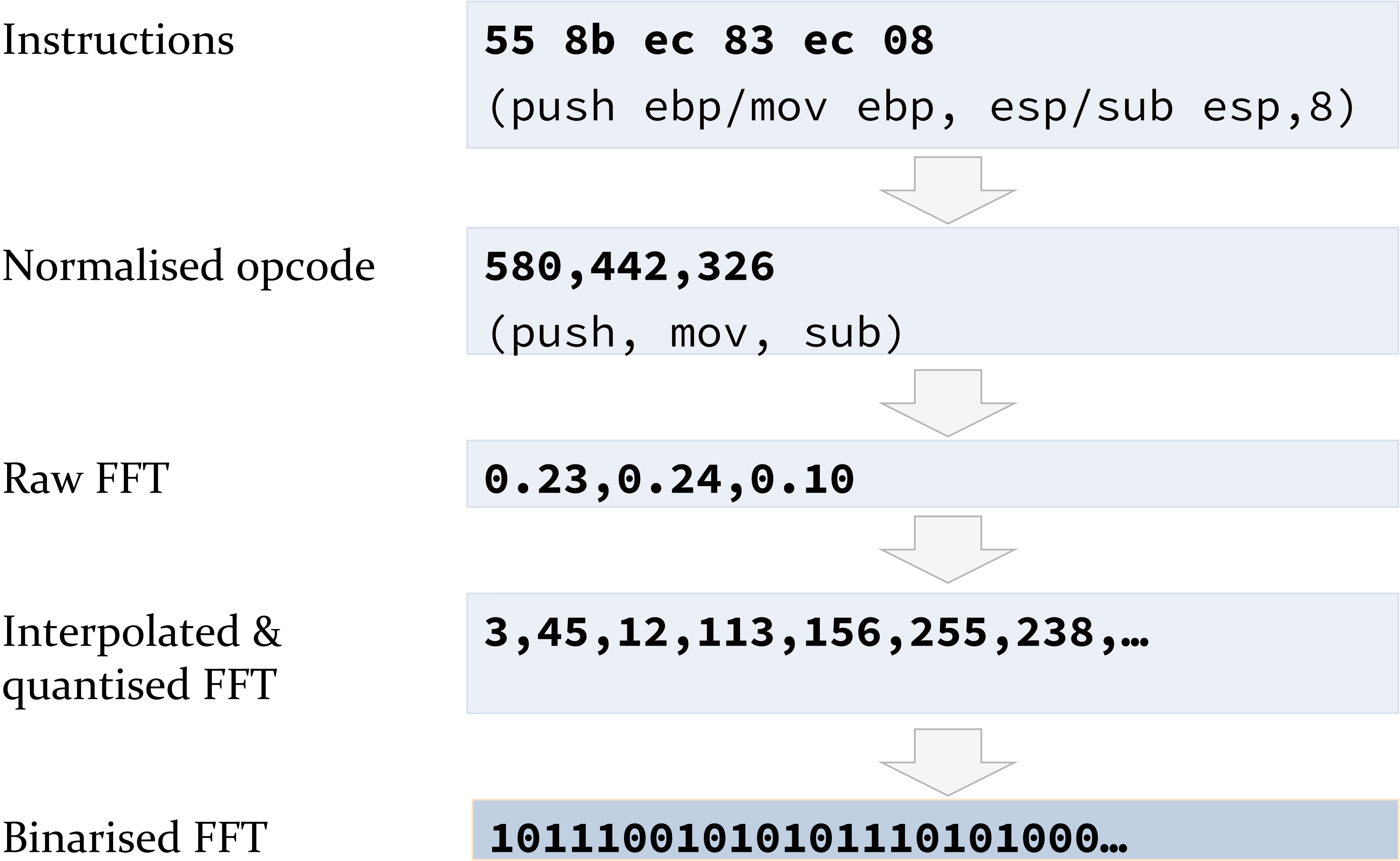
FFT
: mov-add-mov metamorphism



FFT
: push-push-call metamorphism



Dataset



Instructions

55 8b ec 83 ec 08

(push ebp/mov ebp, esp/sub esp,8)

Normalised opcode

580,442,326

(push, mov, sub)

Raw FFT

0.23,0.24,0.10

Interpolated & quantised FFT

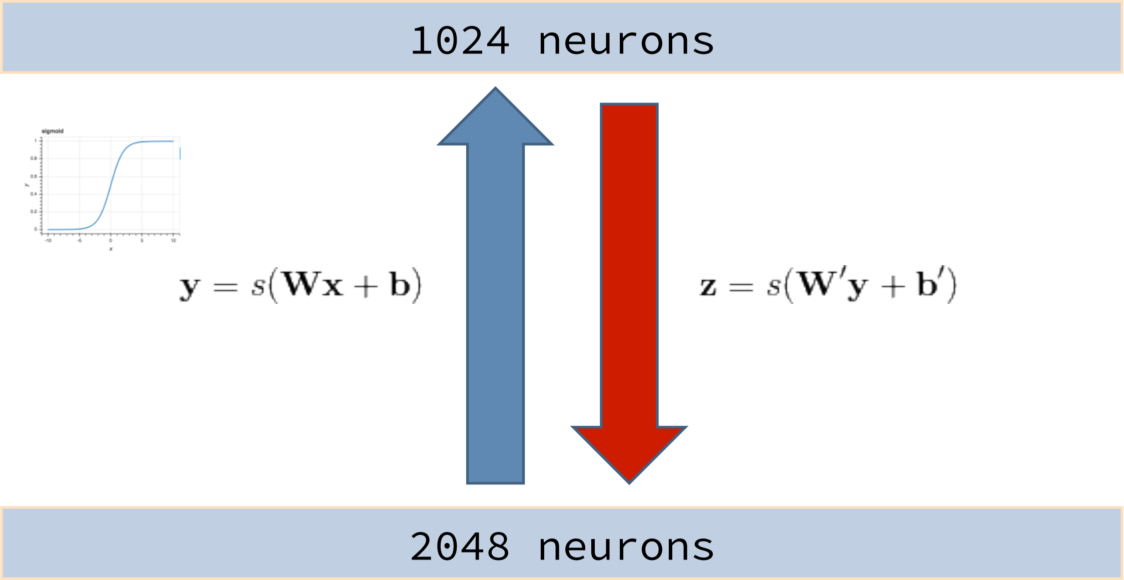
3,45,12,113,156,255,238,…

Binarised FFT

10111001010101110101000…

Neural Network
: Auto Encoder

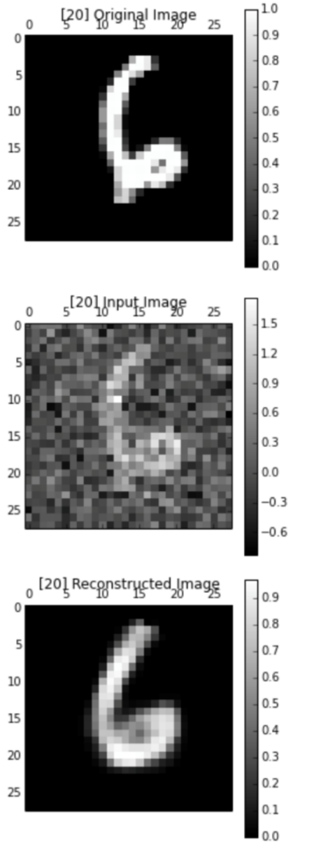
- Auto-Encoder
  - De-noising
  - Restricted Boltzmann Machine
  - Convolutional layer



$$\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b})$$

$$\mathbf{z} = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$$

$$L_H(\mathbf{x}, \mathbf{z}) = -\sum_{k=1}^{d} [\mathbf{x}_k \log \mathbf{z}_k + (1 - \mathbf{x}_k) \log(1 - \mathbf{z}_k)]$$

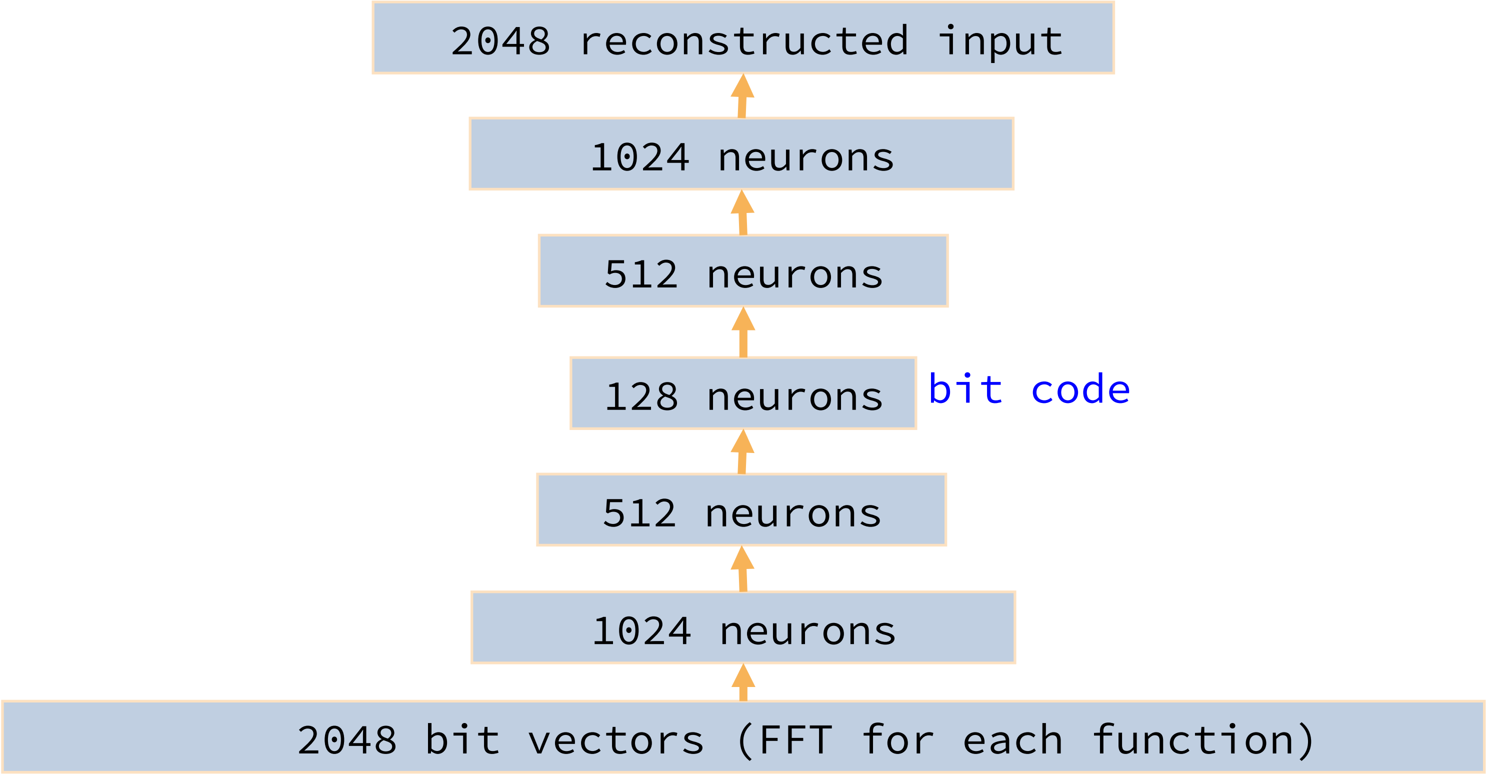


```
Epoch: 025/030 cost: 0.023201655
```

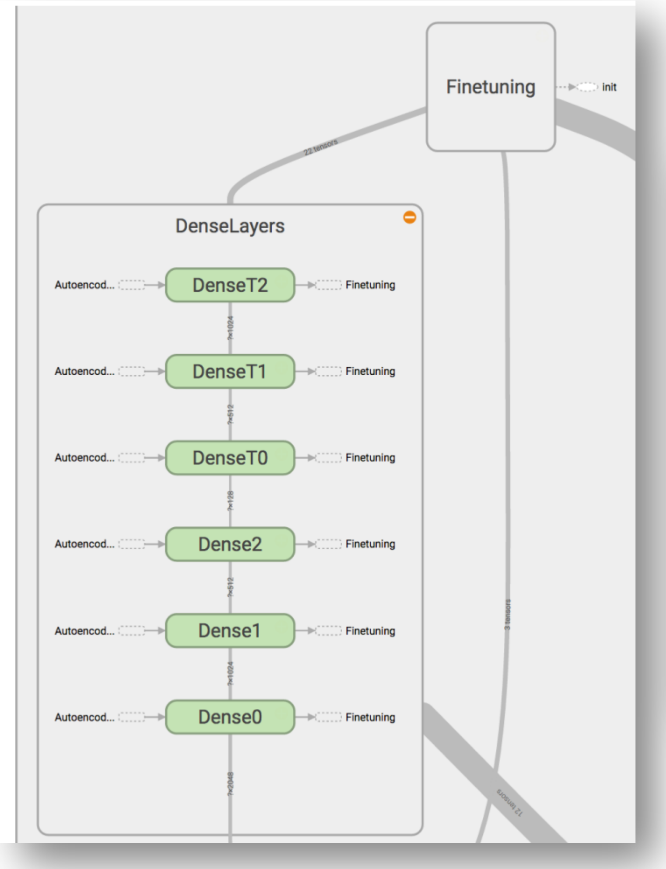
Courtesy of https://github.com/sjchoi86/Tensorflow-101, http://deeplearning.net/tutorial/dA.html#daa

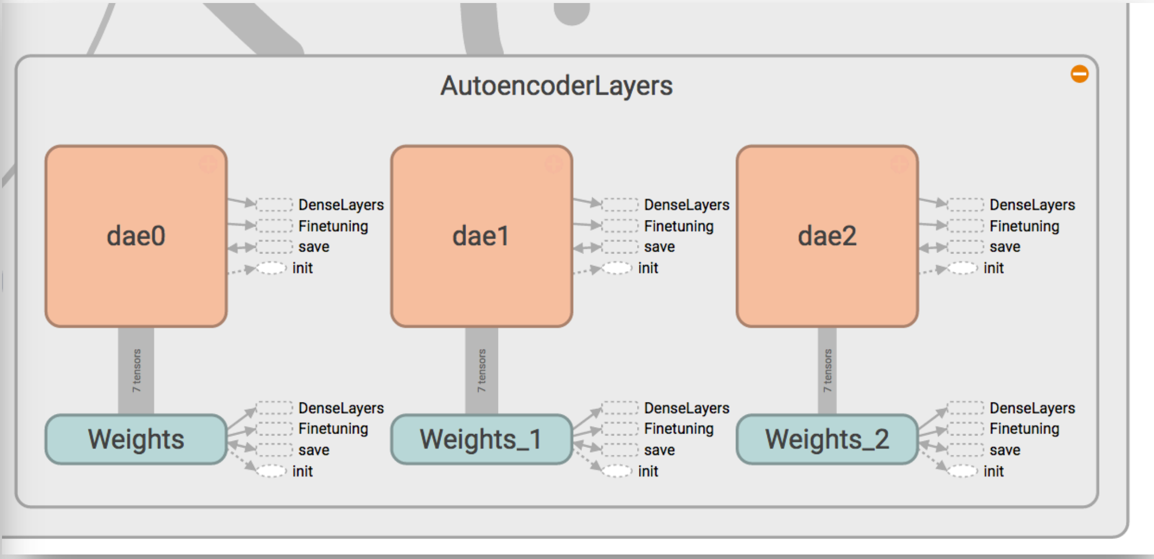
Semantic Hashing

- Deep Auto-Encoder
  - Dimensionality reduction → represented as a fixed size 'code'.
  - Deep auto-encoder performs non-linear mapping

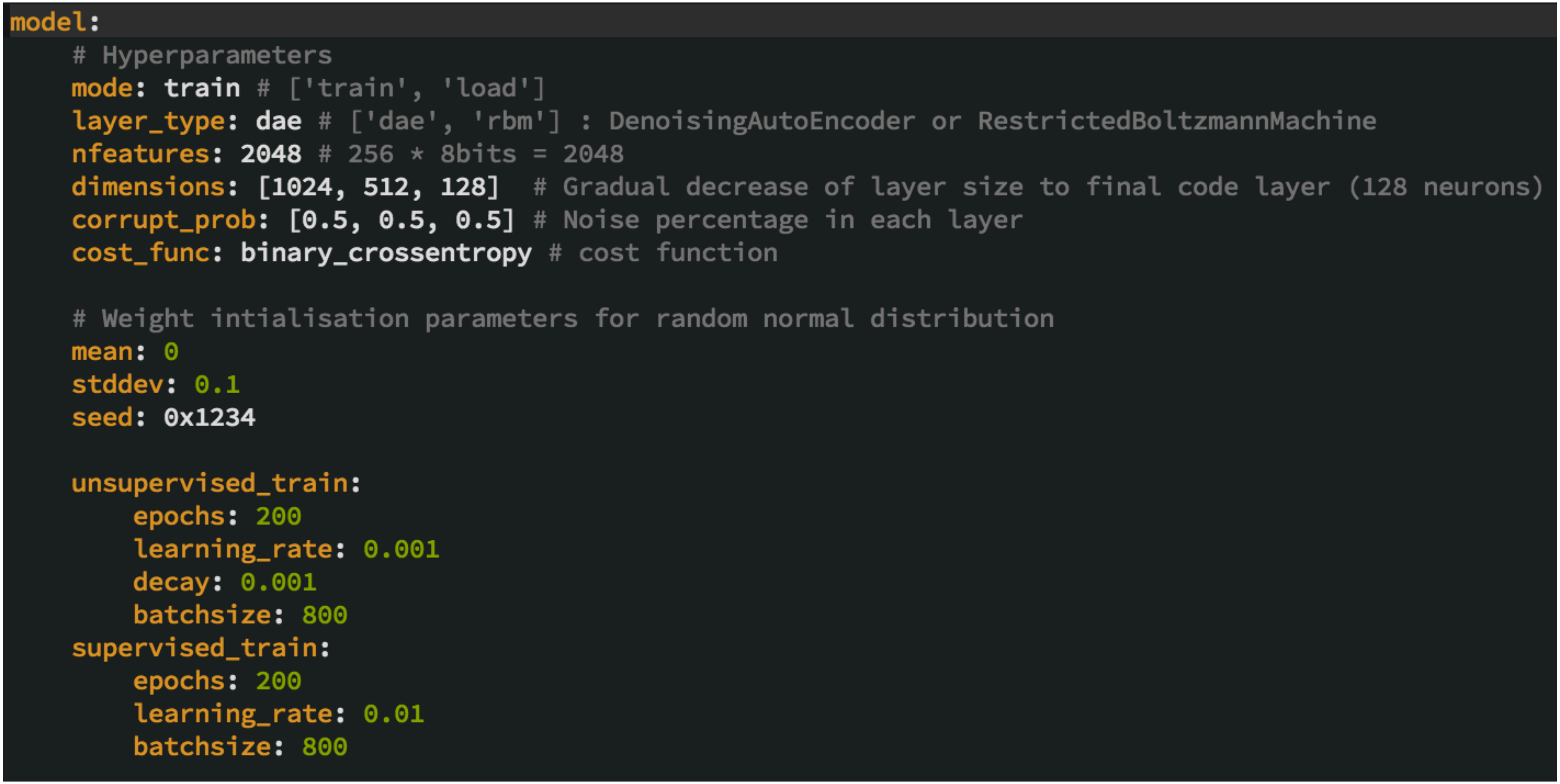


Network Architecture





Model Parameters



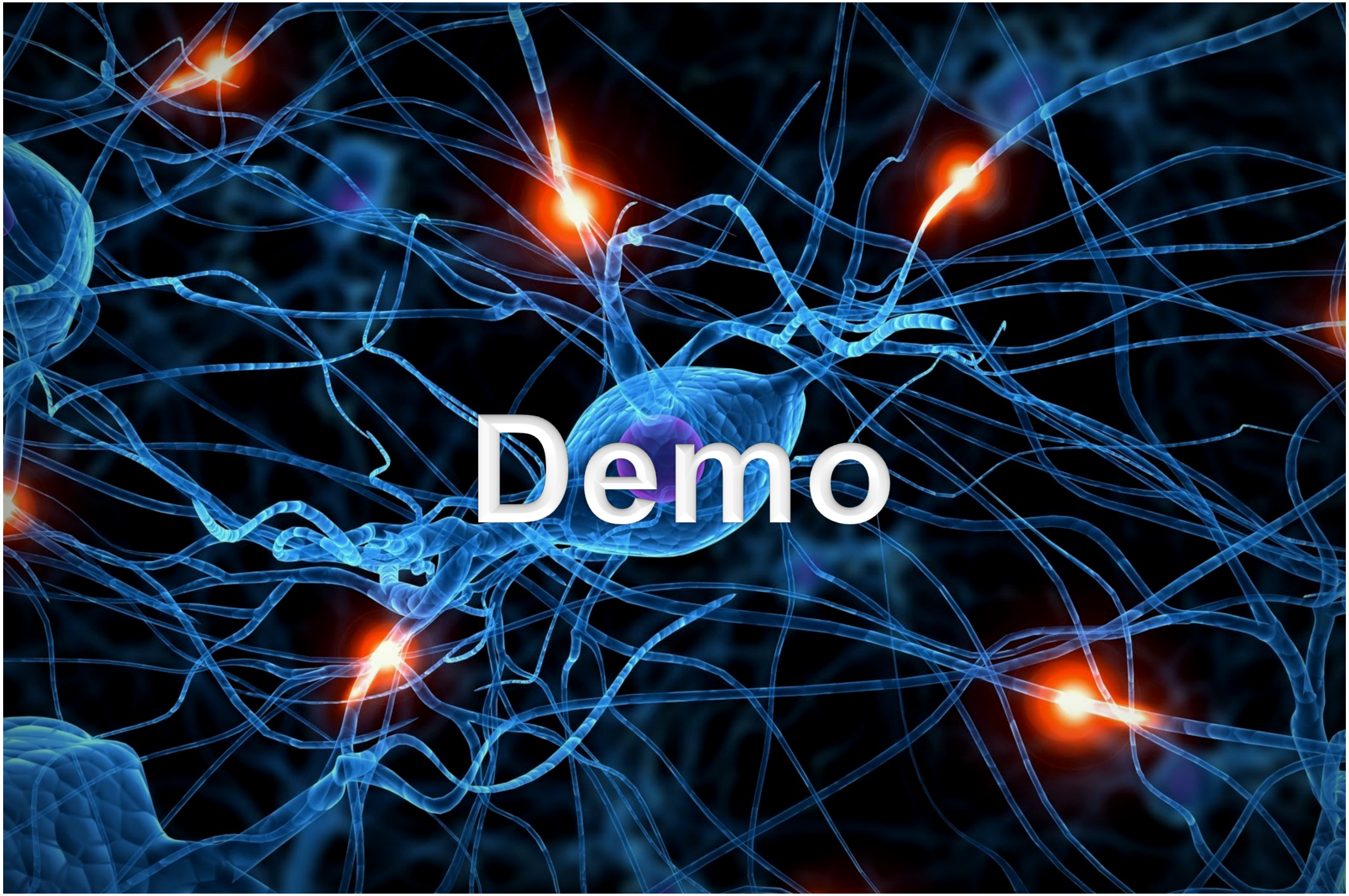
```
model:
    # Hyperparameters
    mode: train # ['train', 'load']
    layer_type: dae # ['dae', 'rbm'] : DenoisingAutoEncoder or RestrictedBoltzmannMachine
    nfeatures: 2048 # 256 * 8bits = 2048
    dimensions: [1024, 512, 128]  # Gradual decrease of layer size to final code layer (128 neurons)
    corrupt_prob: [0.5, 0.5, 0.5] # Noise percentage in each layer
    cost_func: binary_crossentropy # cost function

    # Weight intialisation parameters for random normal distribution
    mean: 0
    stddev: 0.1
    seed: 0x1234

    unsupervised_train:
        epochs: 200
        learning_rate: 0.001
        decay: 0.001
        batchsize: 800
    supervised_train:
        epochs: 200
        learning_rate: 0.01
        batchsize: 800
```

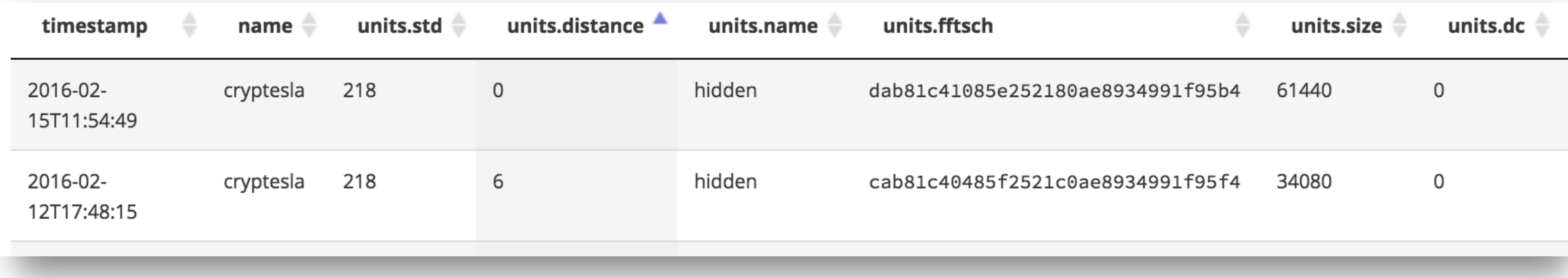
About Dataset & Semantic Hash

- Dataset
  - ~2000 unique ransomware binaries
    - Each binary was sampled from a unique outbreak
    - Each sampled binary can take millions of different forms within the outbreak
  - ~1000 exe/dll from windows/system32/
- Semantic Hash
  - Malware gets detected when semantic hash is identical.
    - An identical semantic hash detects samples with different size and function layouts
  - Malware gets detected when hamming distance of the semantic hash, DC, mean and STD are close.



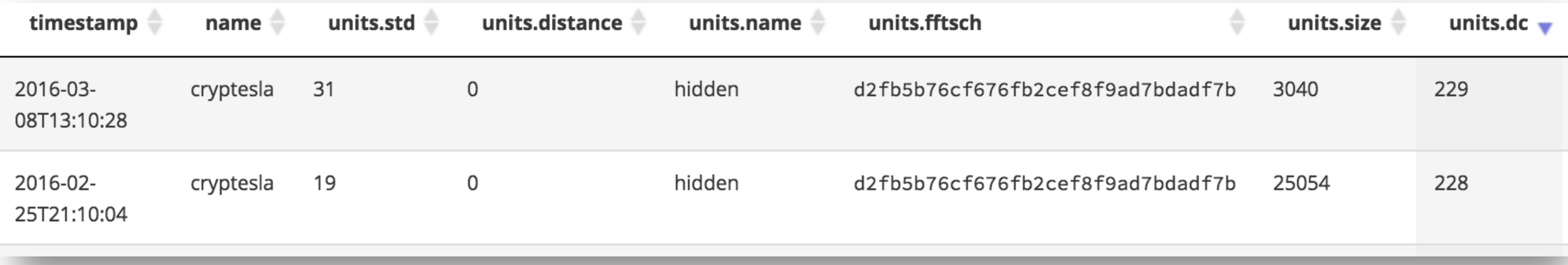
Demo

Metamorphism: push push call



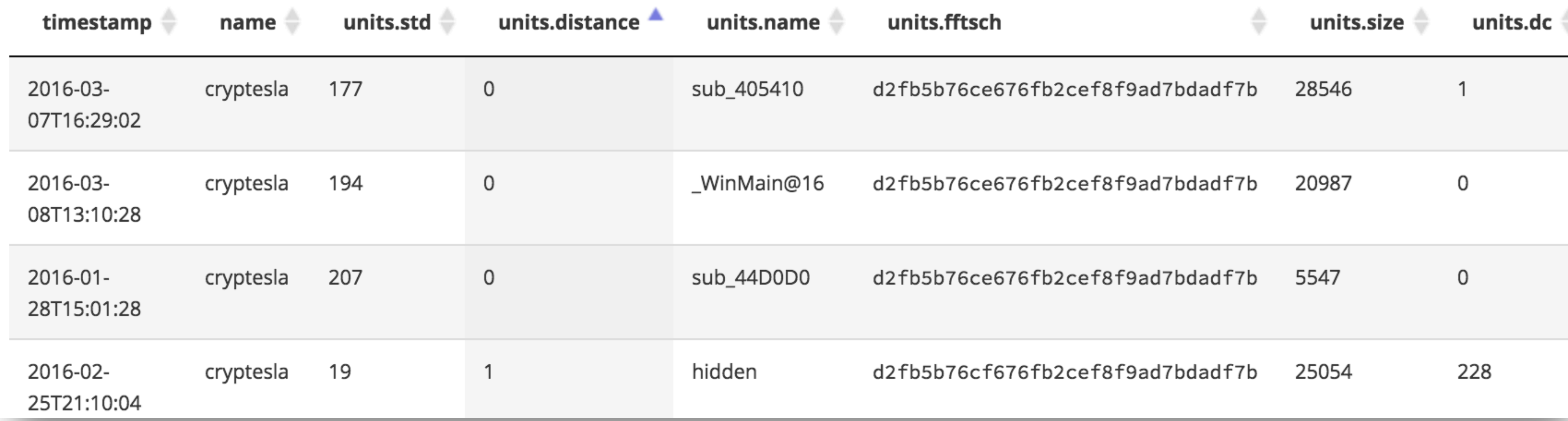
timestamp	name	units.std	units.distance	units.name	units.fftsch	units.size	units.dc
2016-02-15T11:54:49	cryptesla	218	0	hidden	dab81c41085e252180ae8934991f95b4	61440	0
2016-02-12T17:48:15	cryptesla	218	6	hidden	cab81c40485f2521c0ae8934991f95f4	34080	0

Metamorphism: mov sequence



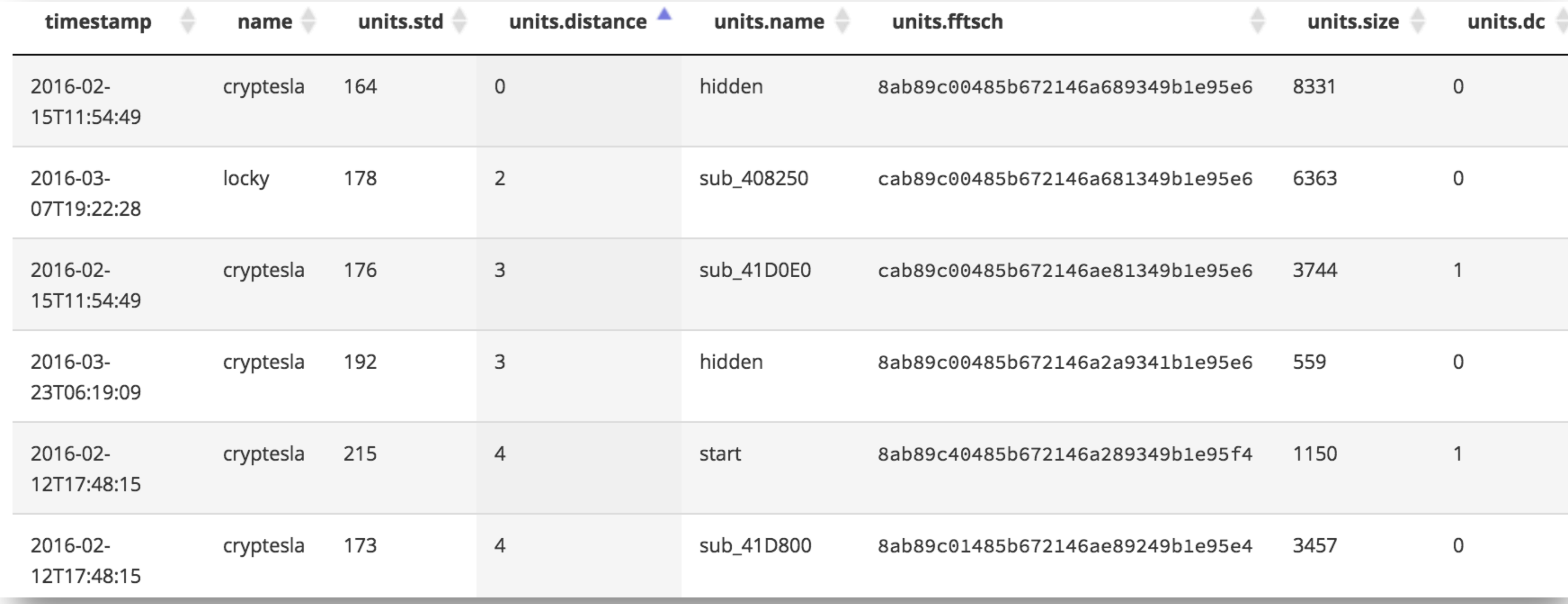
timestamp	name	units.std	units.distance	units.name	units.fftsch	units.size	units.dc
2016-03-08T13:10:28	cryptesla	31	0	hidden	d2fb5b76cf676fb2cef8f9ad7bdadf7b	3040	229
2016-02-25T21:10:04	cryptesla	19	0	hidden	d2fb5b76cf676fb2cef8f9ad7bdadf7b	25054	228

Metamorphism: fld/fstp sequence



timestamp	name	units.std	units.distance	units.name	units.fftsch	units.size	units.dc
2016-03-07T16:29:02	cryptesla	177	0	sub_405410	d2fb5b76ce676fb2cef8f9ad7bdadf7b	28546	1
2016-03-08T13:10:28	cryptesla	194	0	_WinMain@16	d2fb5b76ce676fb2cef8f9ad7bdadf7b	20987	0
2016-01-28T15:01:28	cryptesla	207	0	sub_44D0D0	d2fb5b76ce676fb2cef8f9ad7bdadf7b	5547	0
2016-02-25T21:10:04	cryptesla	19	1	hidden	d2fb5b76cf676fb2cef8f9ad7bdadf7b	25054	228

Metamorphism: add/sub mov

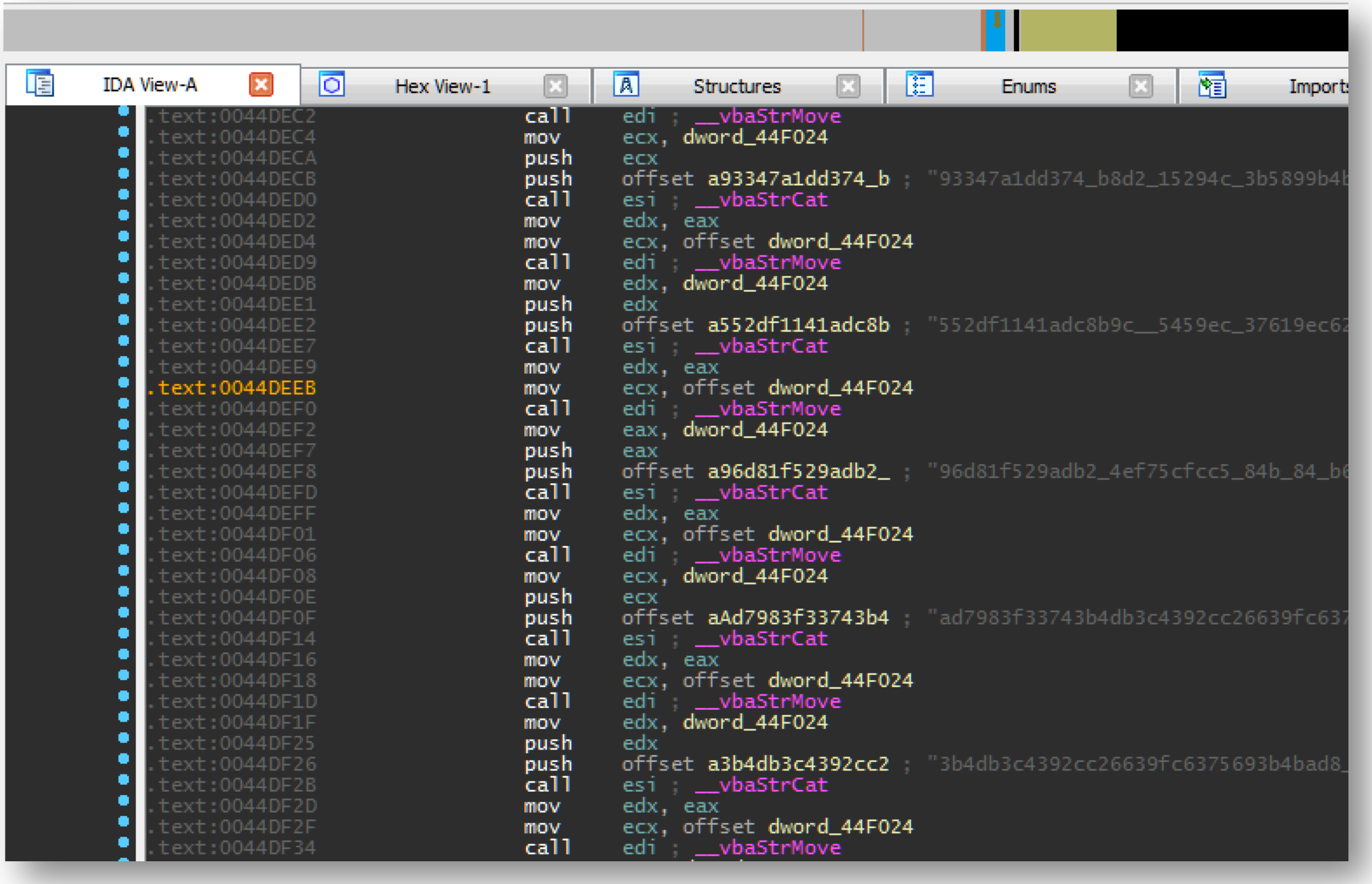


timestamp	name	units.std	units.distance	units.name	units.fftsch	units.size	units.dc
2016-02-15T11:54:49	cryptesla	164	0	hidden	8ab89c00485b672146a689349b1e95e6	8331	0
2016-03-07T19:22:28	locky	178	2	sub_408250	cab89c00485b672146a681349b1e95e6	6363	0
2016-02-15T11:54:49	cryptesla	176	3	sub_41D0E0	cab89c00485b672146ae81349b1e95e6	3744	1
2016-03-23T06:19:09	cryptesla	192	3	hidden	8ab89c00485b672146a2a9341b1e95e6	559	0
2016-02-12T17:48:15	cryptesla	215	4	start	8ab89c40485b672146a289349b1e95f4	1150	1
2016-02-12T17:48:15	cryptesla	173	4	sub_41D800	8ab89c01485b672146ae89249b1e95e4	3457	0

Limitations of The Approach

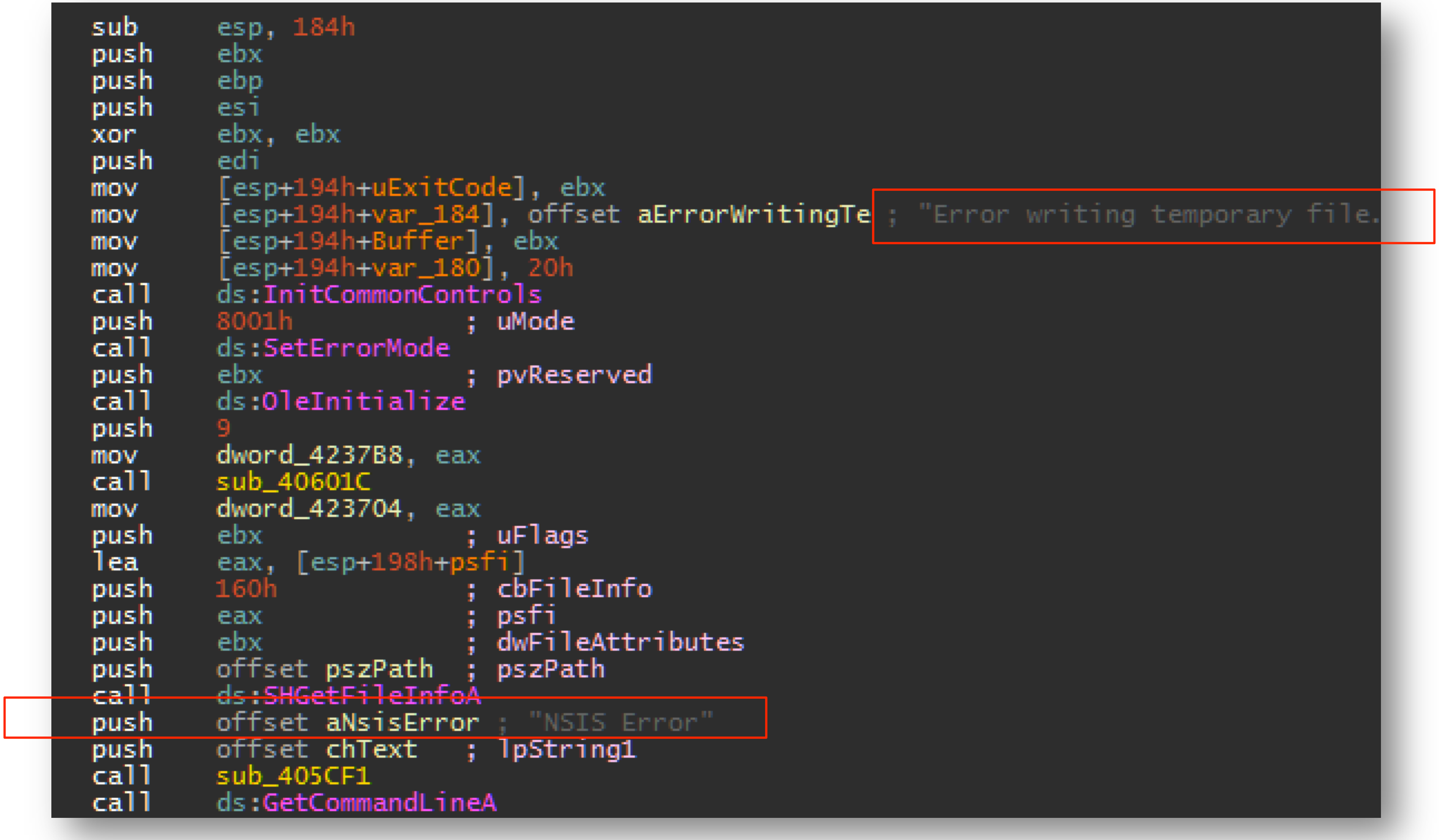
- Layered Executables
  - .NET
  - VBA
  - Self-extracting archives (INNO/NSIS, RAR, ZIP, …)
  - Py2Exe
  - +more
- Hide in plain sight
  - Exploit the assumptions deployed in my approach

Limitations
: Layered executable - VBA



```
.text:0044DEC2                 call    edi ; __vbaStrMove
.text:0044DEC4                 mov     ecx, dword_44F024
.text:0044DECA                 push    ecx
.text:0044DECB                 push    offset a93347a1dd374_b ; "93347a1dd374_b8d2_15294c_3b5899b4
.text:0044DED0                 call    esi ; __vbaStrCat
.text:0044DED2                 mov     edx, eax
.text:0044DED4                 mov     ecx, offset dword_44F024
.text:0044DED9                 call    edi ; __vbaStrMove
.text:0044DEDB                 mov     edx, dword_44F024
.text:0044DEE1                 push    edx
.text:0044DEE2                 push    offset a552df1141adc8b ; "552df1141adc8b9c__5459ec_37619ec6
.text:0044DEE7                 call    esi ; __vbaStrCat
.text:0044DEE9                 mov     edx, eax
.text:0044DEEB                 mov     ecx, offset dword_44F024
.text:0044DEF0                 call    edi ; __vbaStrMove
.text:0044DEF2                 mov     eax, dword_44F024
.text:0044DEF7                 push    eax
.text:0044DEF8                 push    offset a96d81f529adb2_ ; "96d81f529adb2_4ef75cfcc5_84b_84_b
.text:0044DEFD                 call    esi ; __vbaStrCat
.text:0044DEFF                 mov     edx, eax
.text:0044DF01                 mov     ecx, offset dword_44F024
.text:0044DF06                 call    edi ; __vbaStrMove
.text:0044DF08                 mov     ecx, dword_44F024
.text:0044DF0E                 push    ecx
.text:0044DF0F                 push    offset aAd7983f33743b4 ; "ad7983f33743b4db3c4392cc26639fc63
.text:0044DF14                 call    esi ; __vbaStrCat
.text:0044DF16                 mov     edx, eax
.text:0044DF18                 mov     ecx, offset dword_44F024
.text:0044DF1D                 call    edi ; __vbaStrMove
.text:0044DF1F                 mov     edx, dword_44F024
.text:0044DF25                 push    edx
.text:0044DF26                 push    offset a3b4db3c4392cc2 ; "3b4db3c4392cc26639fc6375693b4bad8_
.text:0044DF2B                 call    esi ; __vbaStrCat
.text:0044DF2D                 mov     edx, eax
.text:0044DF2F                 mov     ecx, offset dword_44F024
.text:0044DF34                 call    edi ; __vbaStrMove
```

Limitations
: Layered executable – NSIS installer



```
sub     esp, 184h
push    ebx
push    ebp
push    esi
xor     ebx, ebx
push    edi
mov     [esp+194h+uExitCode], ebx
mov     [esp+194h+var_184], offset aErrorWritingTe ; "Error writing temporary file.
mov     [esp+194h+Buffer], ebx
mov     [esp+194h+var_180], 20h
call    ds:InitCommonControls
push    8001h           ; uMode
call    ds:SetErrorMode
push    ebx             ; pvReserved
call    ds:OleInitialize
push    9
mov     dword_4237B8, eax
call    sub_40601C
mov     dword_423704, eax
push    ebx             ; uFlags
lea     eax, [esp+198h+psfi]
push    160h            ; cbFileInfo
push    eax             ; psfi
push    ebx             ; dwFileAttributes
push    offset pszPath  ; pszPath
call    ds:SHGetFileInfoA
push    offset aNsisError ; "NSIS Error"
push    offset chText   ; lpString1
call    sub_405CF1
call    ds:GetCommandLineA
```

Dealing with large dataset

: What GPU out of memory error looks like

```
...
I tensorflow/core/common_runtime/bfc_allocator.cc:692] 7 Chunks of size 2097152 totalling 14.00MiB
I tensorflow/core/common_runtime/bfc_allocator.cc:692] 8 Chunks of size 8388608 totalling 64.00MiB
I tensorflow/core/common_runtime/bfc_allocator.cc:692] 2 Chunks of size 204800000 totalling 390.62MiB
I tensorflow/core/common_runtime/bfc_allocator.cc:692] 1 Chunks of size 409600000 totalling 390.62MiB
I tensorflow/core/common_runtime/bfc_allocator.cc:692] 5 Chunks of size 819200000 totalling 3.81GiB
I tensorflow/core/common_runtime/bfc_allocator.cc:692] 1 Chunks of size 820948992 totalling 782.92MiB
I tensorflow/core/common_runtime/bfc_allocator.cc:696] Sum Total of in-use chunks: 5.42GiB
I tensorflow/core/common_runtime/bfc_allocator.cc:698] Stats:
Limit:                  5828558848
InUse:                  5819909888
MaxInUse:               5819909888
NumAllocs:                      99
MaxAllocSize:            820948992
```

Dealing with large dataset

- Too many functions in dataset
  - Even for a small dataset (3000 samples), total function count exceeds 1million!
- GPU memory exhaustion
  - Batch processing (reconstruct/evaluate)
  - Even predictions shouldn't be defined as an array
- System memory
  - Do your math between pickled dataset file size and your system memory
  - Consider reading 'Reading Data' section of tensorflow

Fourier Transform As Feature

- Transform arbitrary signal into frequency domain
- Why is it effective for code pattern similarity detection?
  - Each code uniquely identifiable
  - Transformed frequency spectrum retains original data information (We have inverse Fourier transform)
  - Fourier transform of the code is resilient to noise
    - Slight distortion in original code won't affect the characteristics of frequency spectrum much.
  - It is difficult to create a code sequence that has different semantics but has the same frequency spectrum.

Thank You

Sean Park
Senior Malware Scientist , Trend Micro
spark@trendmicro.com