TREND MICRO

Protect against tomorrow's threats | Machine Learning

基於機器學習的惡意軟體分類實作: Microsoft Malware Classification Challenge 經驗談

Trend Micro
ch0upi
miaoski
Kyle Chung
2 Dec 2016

Protect against tomorrow's threats | Machine Learning

ch0upi

- Staff engineer in Trend Micro
- Machine Learning + Data Analysis
- Threat intelligence services
- KDDCup 2014 + KDDCup 2016: Top10
- GoTrend: 6th in UEC Cup 2015

miaoski

- Senior threat researcher in Trend Micro
- Threat intelligence
- Smart City
- SDR
- Arduino + RPi makers
- Loves cats

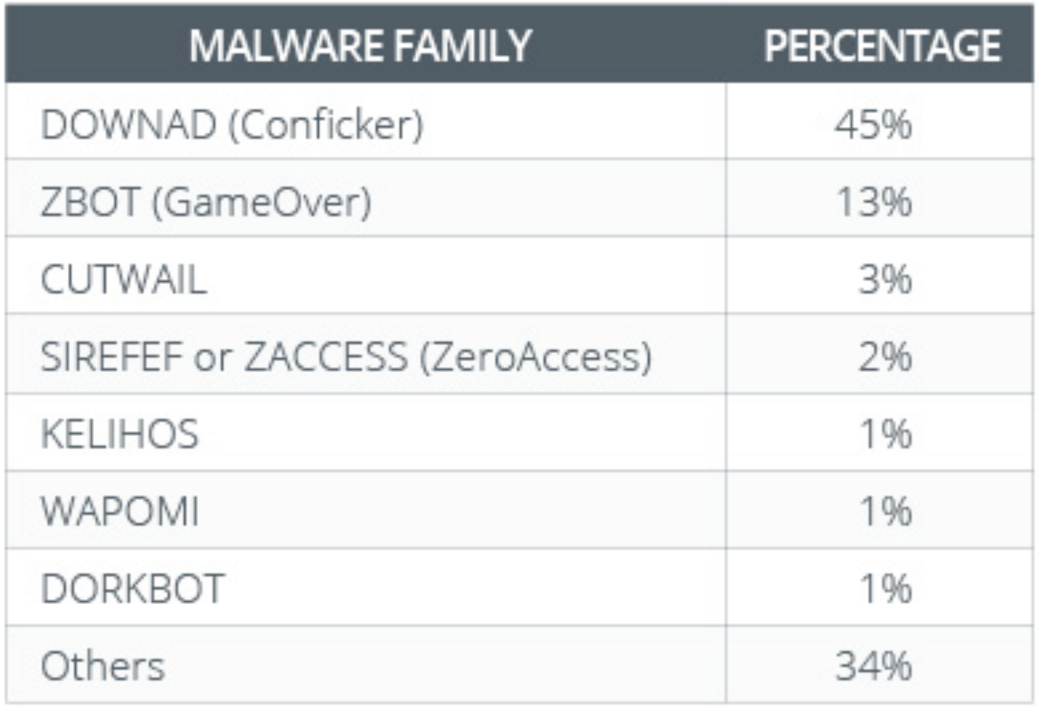
Outline

- Why Malware Classification?
- Machine Learning
- Microsoft Challenge
- How to Solve it?
- Conclusion

MALWARE CLASSIFICATION

What’s Malware Classification?

- Identify malware family

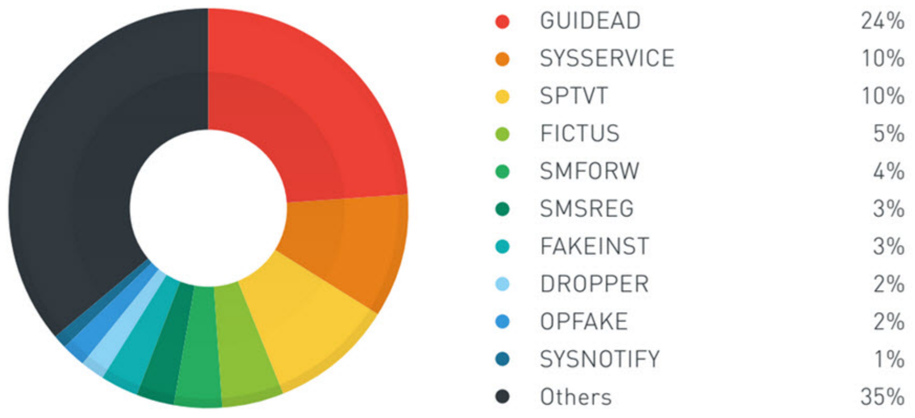


MALWARE FAMILY	PERCENTAGE
DOWNAD (Conficker)	45%
ZBOT (GameOver)	13%
CUTWAIL	3%
SIREFEF or ZACCESS (ZeroAccess)	2%
KELIHOS	1%
WAPOMI	1%
DORKBOT	1%
Others	34%

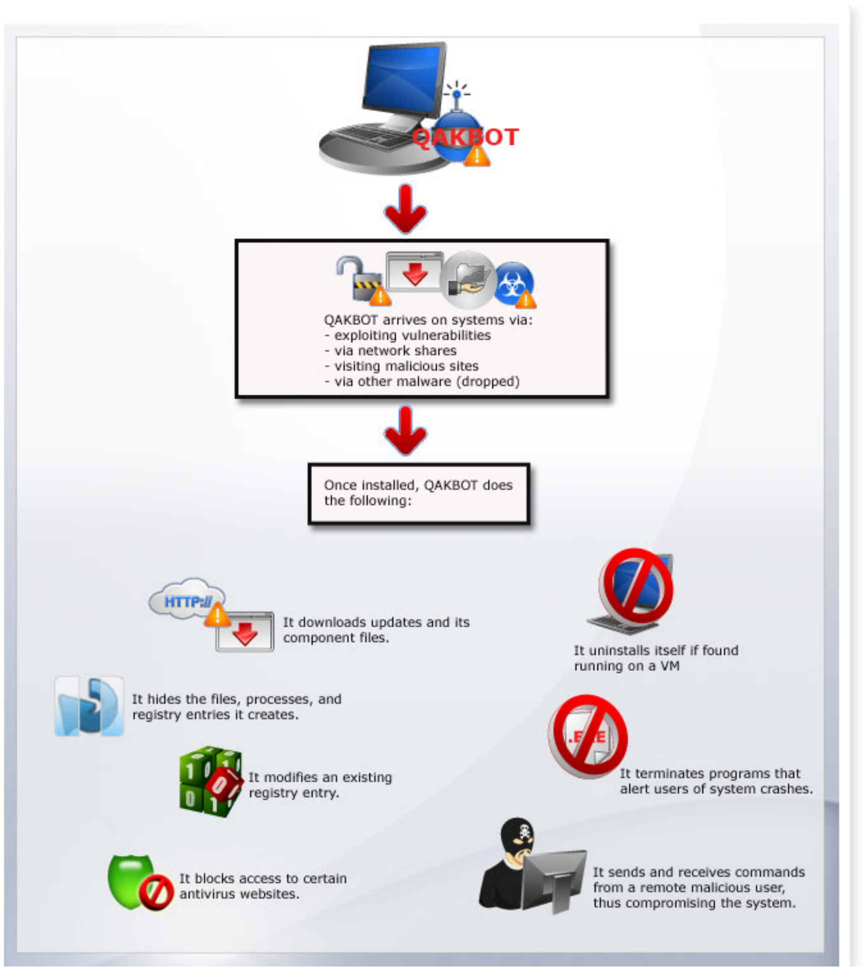
Why Malware Classification?

- Know how to clean
- Possible attribution
- Set proper priority

Top Android malware families (2Q 2015)



GUIDEAD variants don't have graphical user interfaces (GUIs) or icons. They just silently run in the background after installation.



QAKBOT arrives on systems via:
- exploiting vulnerabilities
- via network shares
- visiting malicious sites
- via other malware (dropped)

Once installed, QAKBOT does the following:

It downloads updates and its component files.

It uninstalls itself if found running on a VM

It hides the files, processes, and registry entries it creates.

It modifies an existing registry entry.

It terminates programs that alert users of system crashes.

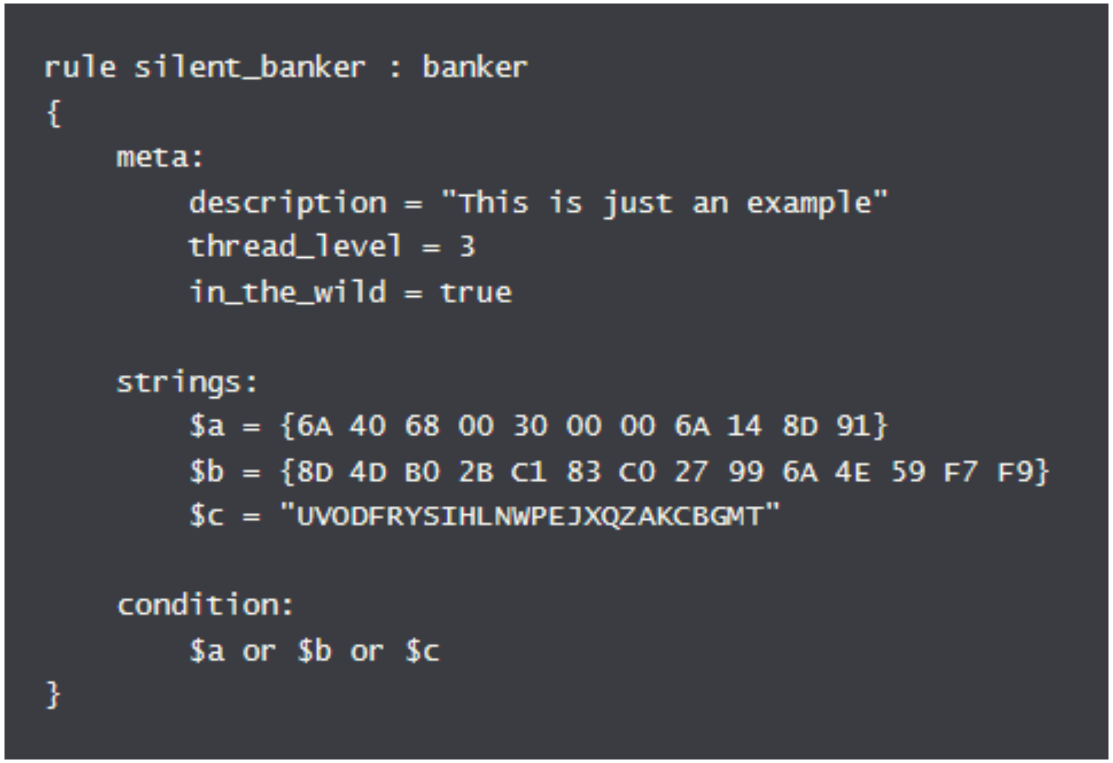
It blocks access to certain antivirus websites.

It sends and receives commands from a remote malicious user, thus compromising the system.

Current Malware Classification

Protect against tomorrow's threats | Machine Learning

- Manually generated by researchers
- Use signature to fingerprint malware
- YARA rules



```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        thread_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

Challenge

- Manual process → wrong family
  - more and more malware families
- Very large volume
  - daily 1M+ samples
- Increasing signatures
  - Slow in scanning + need more storage

BSidesLV 2016: VirusShare

- John Seymour, Labeling the VirusShare Corpus: Lessons Learned, BSidesLV 2016
- VirusShare Corpus: ~20M files

Machine Learning for

- Automation of malware family identification
- Save researcher's effort

MACHINE LEARNING

Machine Learning Steps

- Prepare Data
- Generate Feature
- Train Model
- Make Prediction
- Evaluate

Fruit Classification

- Apple

- Banana

Fruit Features

- Color
- Shape
- Size
- Weight

Learning

- Apple

- Banana

Learned Model of Fruit

- Apple
  - Color: Red
  - Shape: Round
- Banana
  - Color: Yellow
  - Shape: Long

New Fruit Coming…

- Apple? Banana?

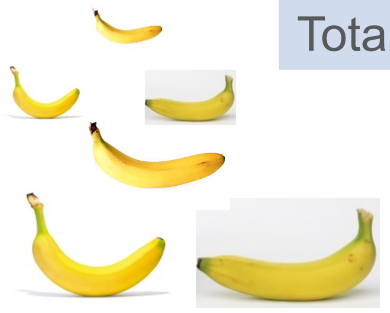
Prediction of Fruits

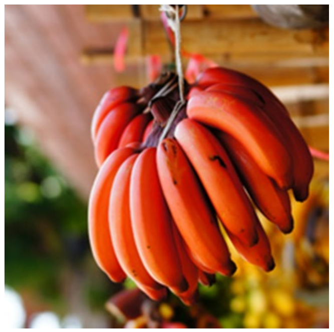
- Fruit 1
  - Color: Red => Apple
  - Shape: Round => Apple
- Fruit 2
  - Color: Yellow => Banana
  - Shape: Long => Banana

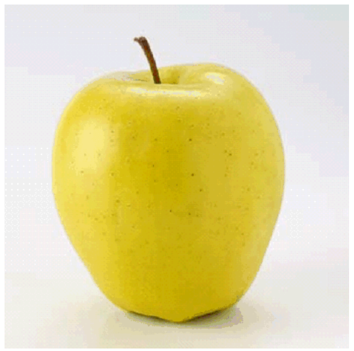
Evaluation of Fruit

- Accuracy: (9+9)/20 = 90%

	Apple	Banana
Apple	9	1
Banana	1	9
Total	10	10







Machine Learning Steps

- Prepare Data
- Generate Feature
- Train Model
- Make Prediction
- Evaluate

Machine Learning is ...

- Mathematical methods and algorithms
- From historical labelled data
- Find a separating hyperplane
- Apply it on future data

Feature is ...

- Measurable property of a phenomenon being observed
  - Use to describe entries
- Feature vector
  - input of machine learning algorithm
- Source of features
  - data exploring
  - domain knowledge

Model is …

- A mathematical description of how to classify the data
- Parameters tuned by certain algorithm
  - training
- Used to make prediction

Prediction

- Identify the class of new entities
- With trained model from training data

Evaluation

- Review model result by some measurements
- Cross validation
- Evaluation functions
  - Accuracy
  - logloss
  - AUC
  - precision, recall, F1

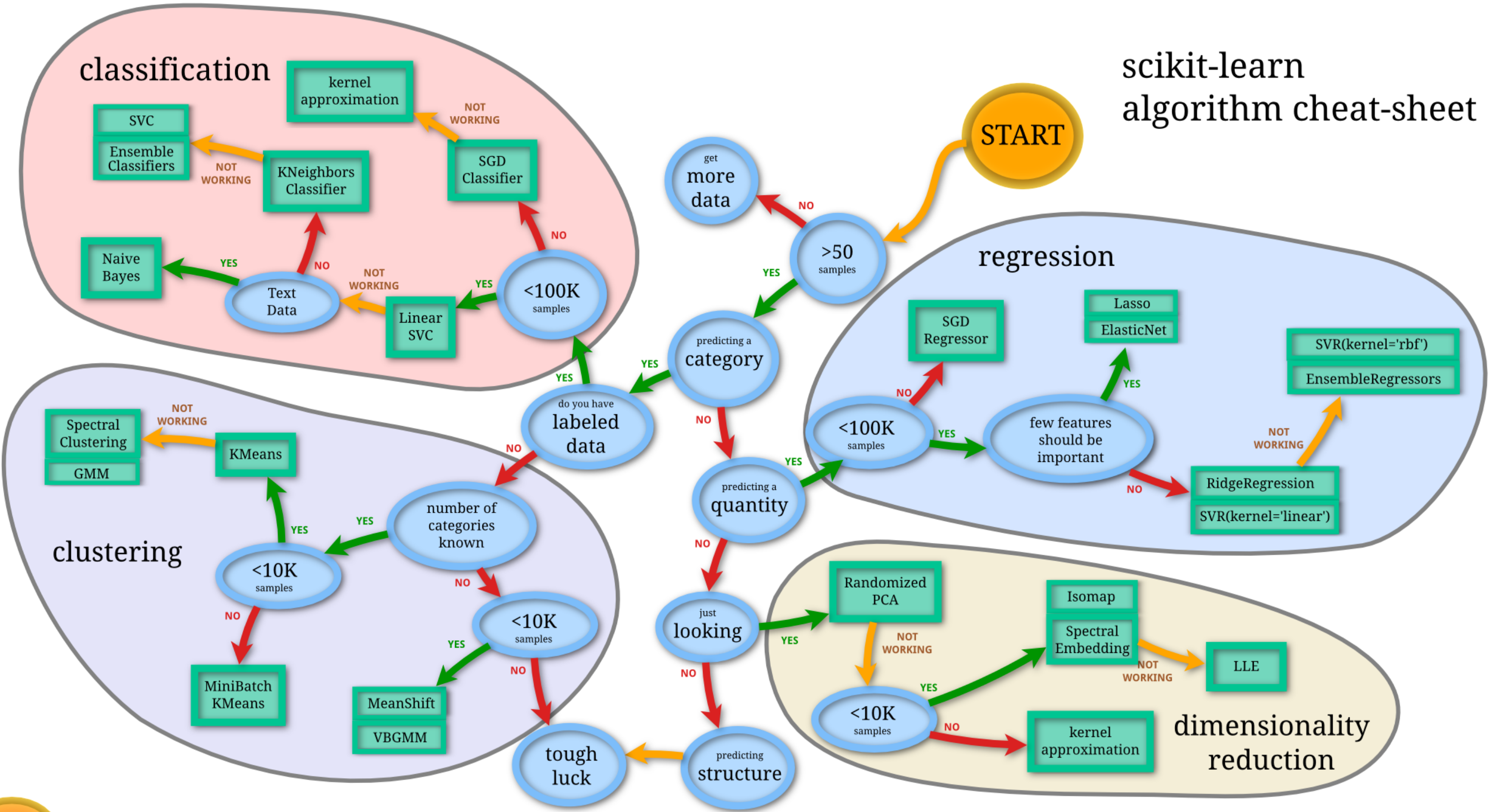
Glue Language

- Glue the steps of Machine Learning
- Batch running for large amount of data
- Integration with Hadoop, Spark
- Rich libraries/Algorithm support
- Easy to develop/learn

scikit learn

- Open source machine learning library for **python**
- Various classification, regression and clustering algorithms
- Interoperate with NumPy, SciPy, and underlying BLAS

scikit learn cheat-sheet



Back

Supported Algorithm in scikit learn

- Classification Algorithm
  - Logistic Regression: linear_model.LogisticRegression()
  - SVM: svm.SVC()
  - Random Forest: ensemble.RandomForestClassifier()
- Interface
  - fit(X, Y): train model
  - Yp=predict(X): make prediction

Evaluation Functions in scikit learn

- Evaluation functions
  - metrics.accuracy_score()
  - metrics.log_loss()
  - metrics.auc()
  - metrics.f1_score()

  - metrics.confusion_matrix()
  - metrics.classification_report()

Protect against tomorrow's threats | Machine Learning

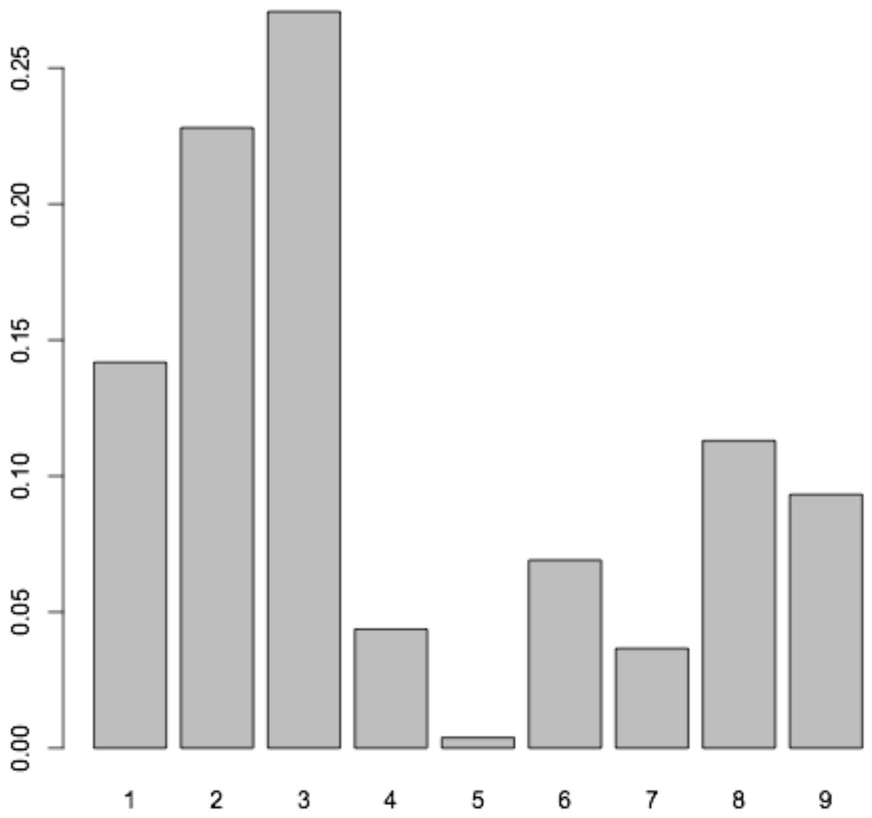
MICROSOFT MALWARE CLASSIFICATION CHALLENGE

Microsoft Malware Classification Challenge

- Hosted by WWW 2015 / BIG 2015
- Microsoft Malware Protection Center
  Microsoft Azure Machine Learning
  Microsoft Talent Management
- PE Hexdump & Disassembled
- Training: 10,868 (compressed: 17.5GB)
- Testing: 10,873 (compressed: 17.7GB)

9 Classes

	Category	Count
1	Ramnit	1541
2	Lollipop	2478
3	Kelihos_ver3	2942
4	Vundo	475
5	Simda	42
6	Tracur	751
7	Kelihos_ver1	398
8	Obfuscator.ACY	1228
9	Gatak	1013
	Total	10868

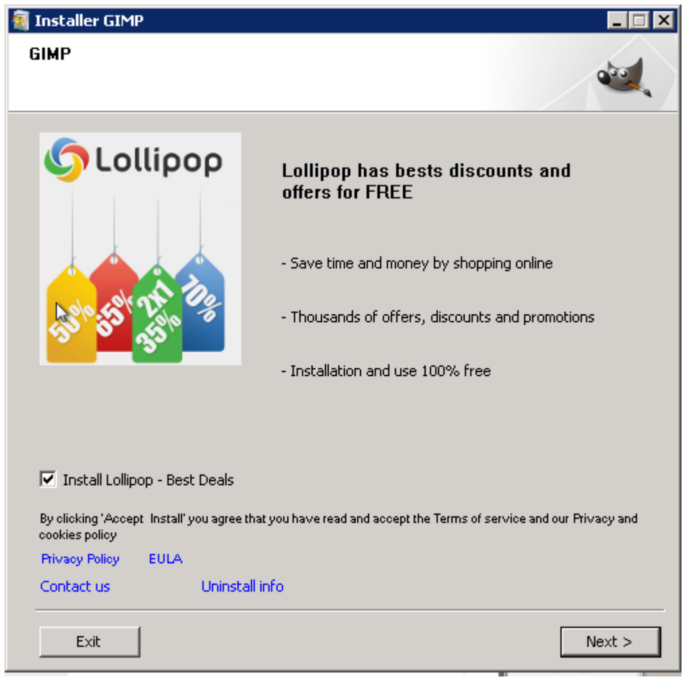


Class: Ramnit

- Steal sensitive personal information
- Infected through removable drivers
- Copy itself using a hard-coded name, or with a random file name to a random folder
- Inject codes into svchost.exe
- Infects DLL, EXE, HTML

Class: Lollipop

- An adware shows ads when browsing web
- Bundle with third-party software
- Auto run when Windows starting


Installer GIMP

GIMP

Lollipop

Lollipop has bests discounts and offers for FREE

- Save time and money by shopping online

- Thousands of offers, discounts and promotions

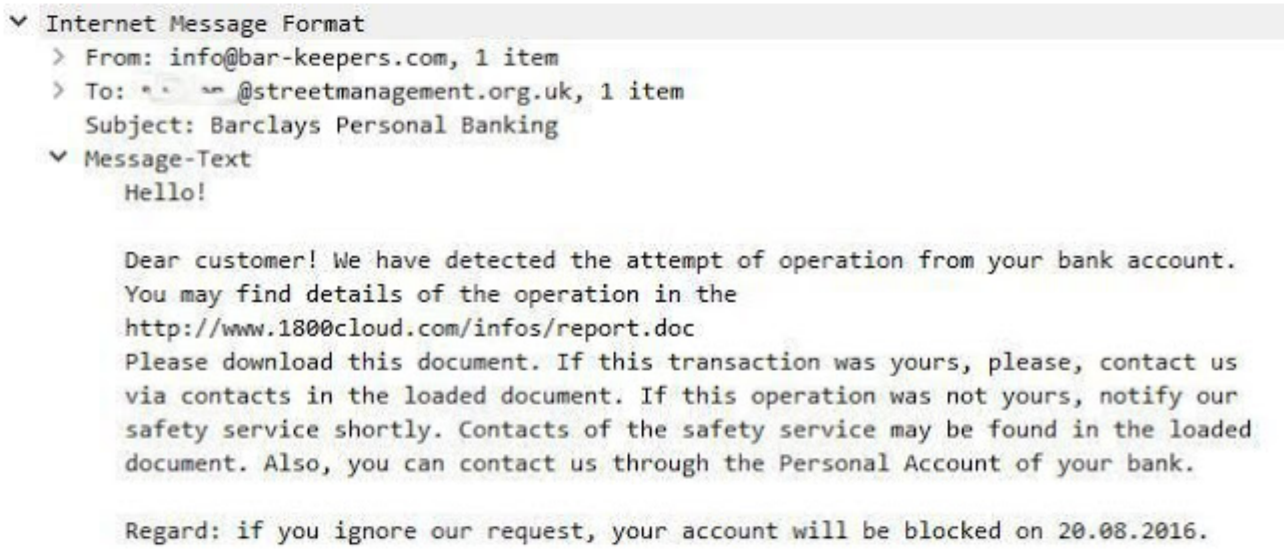
- Installation and use 100% free

☑ Install Lollipop - Best Deals

By clicking 'Accept Install' you agree that you have read and accept the Terms of service and our Privacy and cookies policy

Class: Kelihos

- A Trojan family distributes spam email with malware download link
- Communicate with C&C server
- Some variants install WinPcap to spy network activity

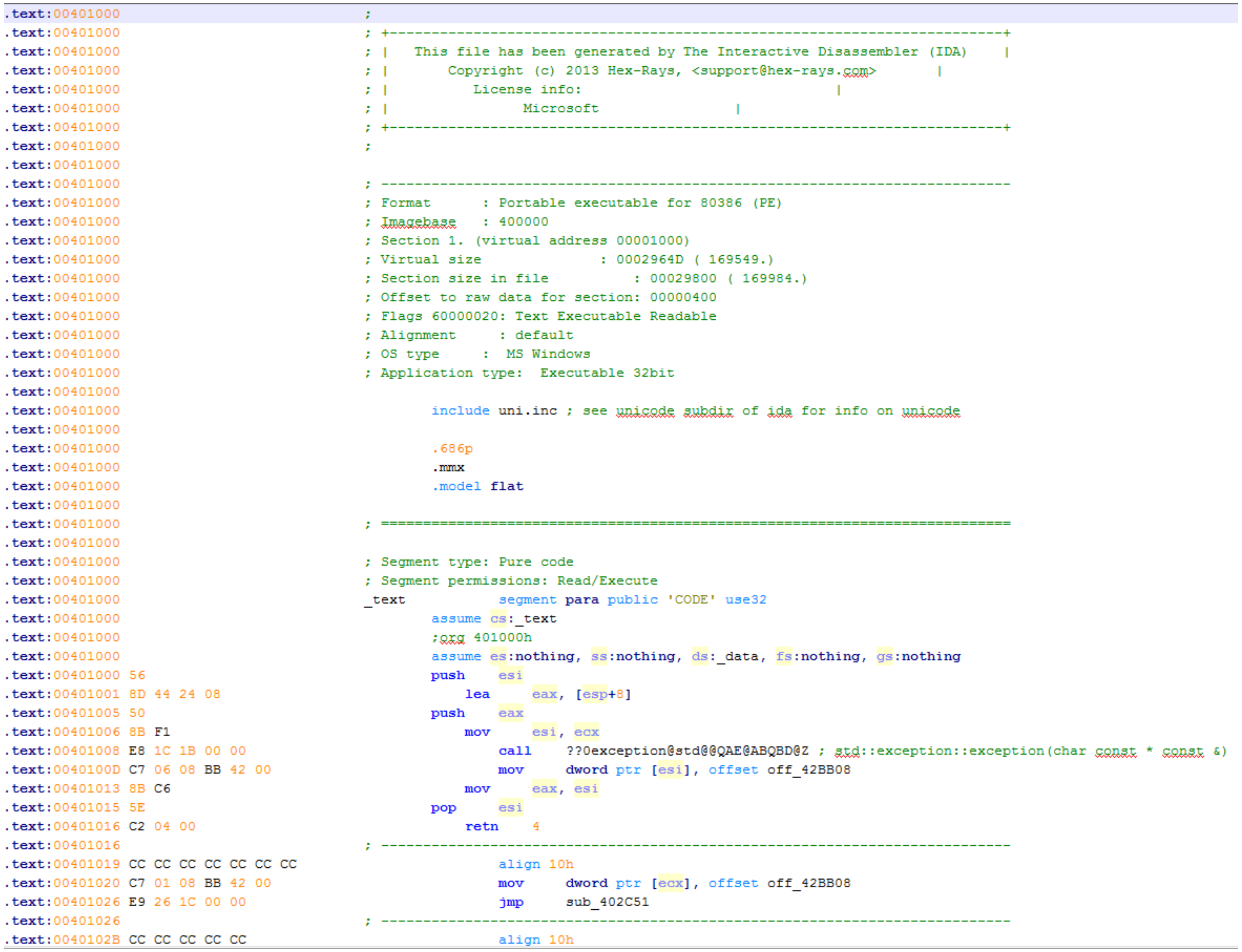


```
⌄ Internet Message Format
  › From: info@bar-keepers.com, 1 item
  › To: [redacted]@streetmanagement.org.uk, 1 item
    Subject: Barclays Personal Banking
  ⌄ Message-Text
      Hello!

      Dear customer! We have detected the attempt of operation from your bank account.
      You may find details of the operation in the
      http://www.1800cloud.com/infos/report.doc
      Please download this document. If this transaction was yours, please, contact us
      via contacts in the loaded document. If this operation was not yours, notify our
      safety service shortly. Contacts of the safety service may be found in the loaded
      document. Also, you can contact us through the Personal Account of your bank.

      Regard: if you ignore our request, your account will be blocked on 20.08.2016.
```

PE Hexdump w/o Header

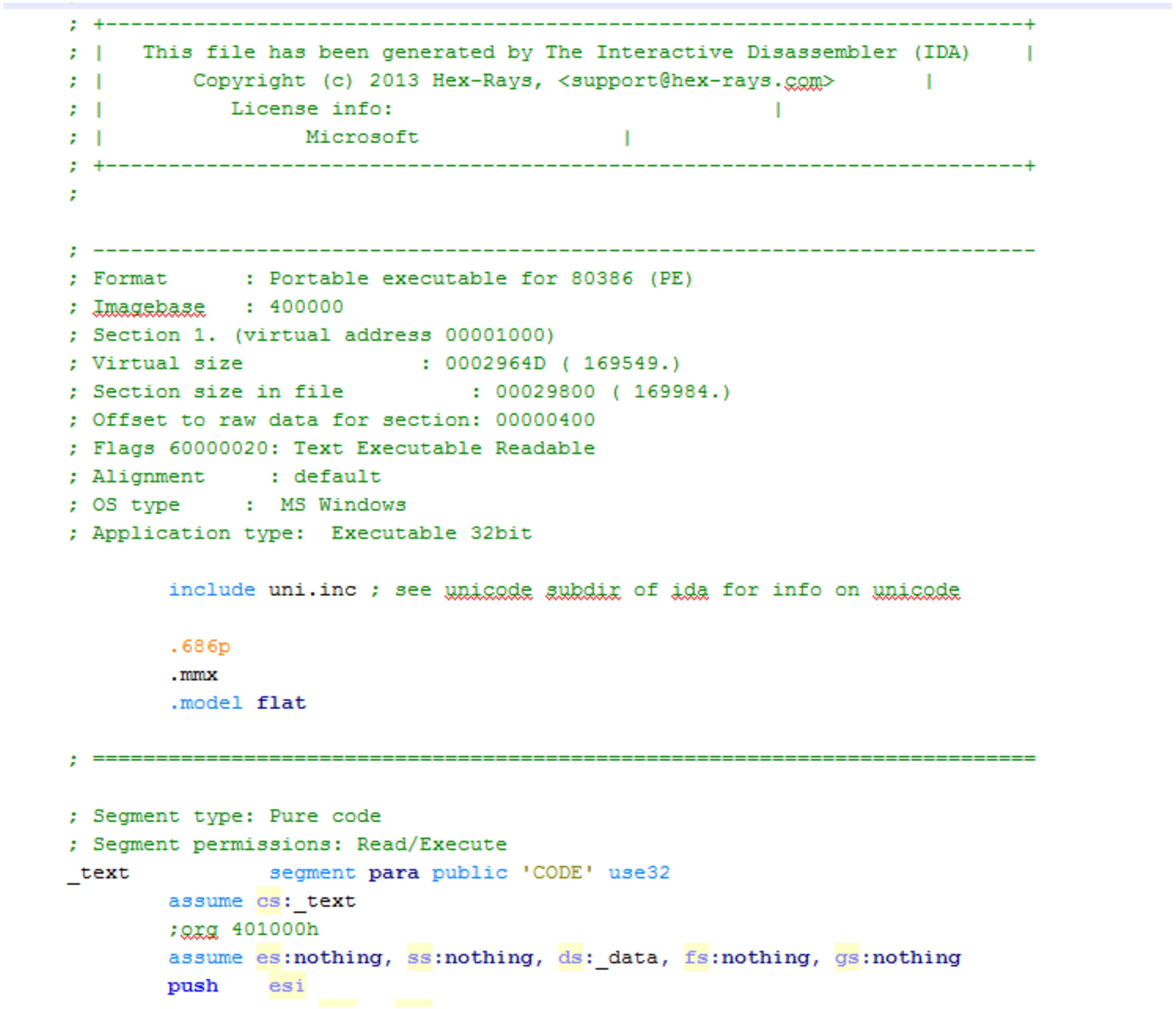
```
00401000 56 8D 44 24 08 50 8B F1 E8 1C 1B 00 00 C7 06 08
00401010 BB 42 00 8B C6 5E C2 04 00 CC CC CC CC CC CC CC
00401020 C7 01 08 BB 42 00 E9 26 1C 00 00 CC CC CC CC CC
00401030 56 8B F1 C7 06 08 BB 42 00 E8 13 1C 00 00 F6 44
00401040 24 08 01 74 09 56 E8 6C 1E 00 00 83 C4 04 8B C6
00401050 5E C2 04 00 CC CC CC CC CC CC CC CC CC CC CC CC
00401060 8B 44 24 08 8A 08 8B 54 24 04 88 0A C3 CC CC CC
00401070 8B 44 24 04 8D 50 01 8A 08 40 84 C9 75 F9 2B C2
00401080 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
00401090 8B 44 24 10 8B 4C 24 0C 8B 54 24 08 56 8B 74 24
004010A0 08 50 51 52 56 E8 18 1E 00 00 83 C4 10 8B C6 5E
004010B0 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
004010C0 8B 44 24 10 8B 4C 24 0C 8B 54 24 08 56 8B 74 24

00436FD0 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
00436FE0 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
00436FF0 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
00437000 38 BA 42 00 01 00 00 00 7C 1D 43 00 00 00 00 00
00437010 2E 3F 41 56 62 61 64 5F 61 6C 6C 6F 63 40 73 74
00437020 64 40 40 00 00 00 00 00 00 00 00 00 7C 1D 43 00
00437030 00 00 00 00 2E 3F 41 56 65 78 63 65 70 74 69 6F
00437040 6E 40 73 74 64 40 40 00 00 00 00 00 00 00 00 00
```

IDA Pro Dump



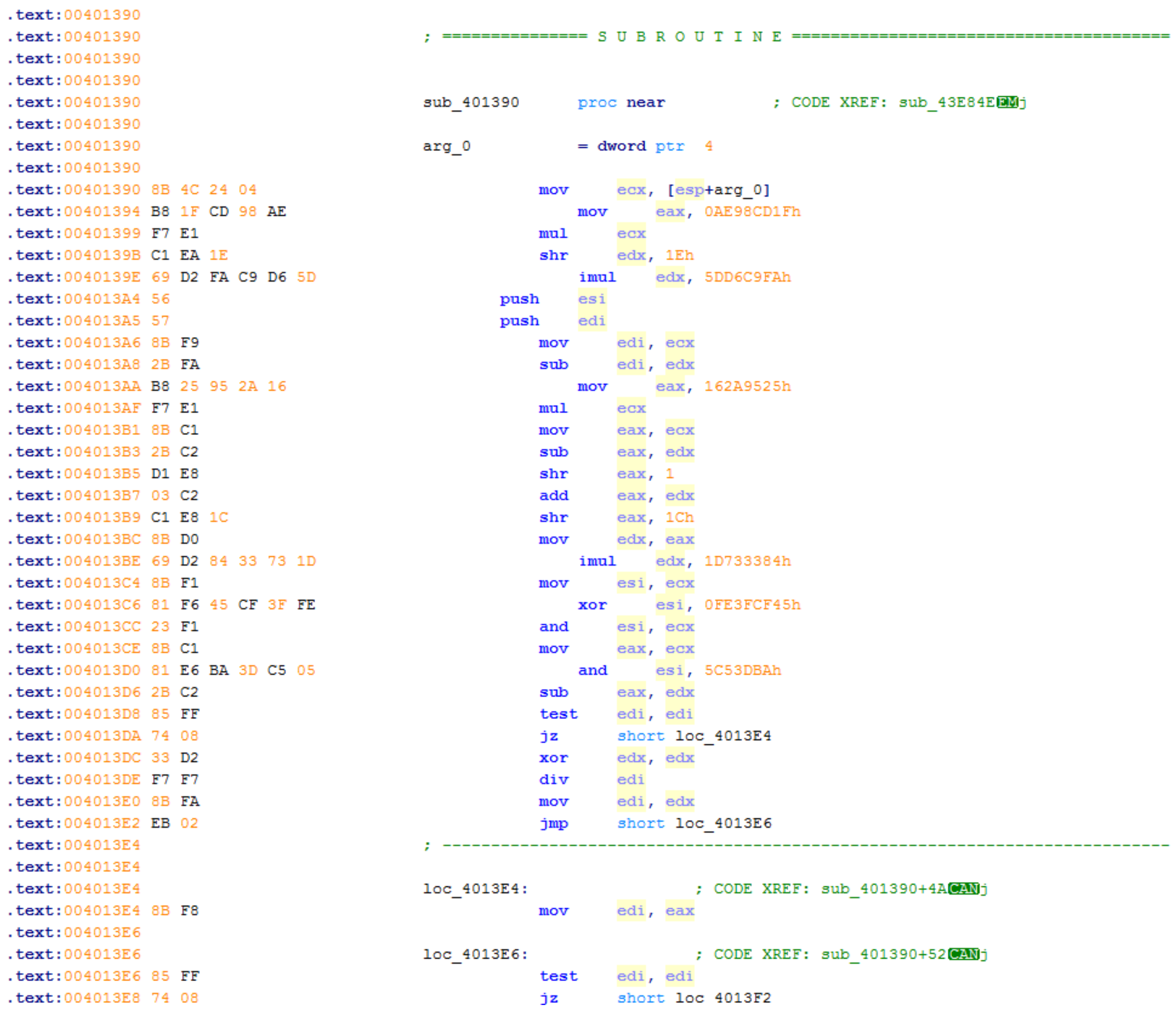
```
.text:00401000                         ;
.text:00401000                         ; +-------------------------------------------------------------------------+
.text:00401000                         ; |   This file has been generated by The Interactive Disassembler (IDA)    |
.text:00401000                         ; |       Copyright (c) 2013 Hex-Rays, <support@hex-rays.com>       |
.text:00401000                         ; |          License info:                          |
.text:00401000                         ; |                Microsoft                 |
.text:00401000                         ; +-------------------------------------------------------------------------+
.text:00401000                         ;
.text:00401000
.text:00401000                         ; ---------------------------------------------------------------------------
.text:00401000                         ; Format      : Portable executable for 80386 (PE)
.text:00401000                         ; Imagebase   : 400000
.text:00401000                         ; Section 1. (virtual address 00001000)
.text:00401000                         ; Virtual size              : 0002964D ( 169549.)
.text:00401000                         ; Section size in file          : 00029800 ( 169984.)
.text:00401000                         ; Offset to raw data for section: 00000400
.text:00401000                         ; Flags 60000020: Text Executable Readable
.text:00401000                         ; Alignment     : default
.text:00401000                         ; OS type     :  MS Windows
.text:00401000                         ; Application type:  Executable 32bit
.text:00401000
.text:00401000                                 include uni.inc ; see unicode subdir of ida for info on unicode
.text:00401000
.text:00401000                                 .686p
.text:00401000                                 .mmx
.text:00401000                                 .model flat
.text:00401000
.text:00401000                         ; ===========================================================================
.text:00401000
.text:00401000                         ; Segment type: Pure code
.text:00401000                         ; Segment permissions: Read/Execute
.text:00401000                         _text           segment para public 'CODE' use32
.text:00401000                                 assume cs:_text
.text:00401000                                 ;org 401000h
.text:00401000                                 assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing
.text:00401000 56                              push    esi
.text:00401001 8D 44 24 08                         lea     eax, [esp+8]
.text:00401005 50                              push    eax
.text:00401006 8B F1                               mov     esi, ecx
.text:00401008 E8 1C 1B 00 00                          call    ??0exception@std@@QAE@ABQBD@Z ; std::exception::exception(char const * const &)
.text:0040100D C7 06 08 BB 42 00                       mov     dword ptr [esi], offset off_42BB08
.text:00401013 8B C6                               mov     eax, esi
.text:00401015 5E                              pop     esi
.text:00401016 C2 04 00                            retn    4
.text:00401016                         ; ---------------------------------------------------------------------------
.text:00401019 CC CC CC CC CC CC CC                    align 10h
.text:00401020 C7 01 08 BB 42 00                       mov     dword ptr [ecx], offset off_42BB08
.text:00401026 E9 26 1C 00 00                          jmp     sub_402C51
.text:00401026                         ; ---------------------------------------------------------------------------
.text:0040102B CC CC CC CC CC                          align 10h
```

IDA Pro Dump



```
; +-------------------------------------------------------------------------+
; |   This file has been generated by The Interactive Disassembler (IDA)    |
; |       Copyright (c) 2013 Hex-Rays, <support@hex-rays.com>       |
; |          License info:                                  |
; |                Microsoft                          |
; +-------------------------------------------------------------------------+
;


; -------------------------------------------------------------------------
; Format      : Portable executable for 80386 (PE)
; Imagebase   : 400000
; Section 1. (virtual address 00001000)
; Virtual size                  : 0002964D ( 169549.)
; Section size in file              : 00029800 ( 169984.)
; Offset to raw data for section: 00000400
; Flags 60000020: Text Executable Readable
; Alignment     : default
; OS type       :  MS Windows
; Application type:  Executable 32bit

        include uni.inc ; see unicode subdir of ida for info on unicode

        .686p
        .mmx
        .model flat

; ===========================================================================

; Segment type: Pure code
; Segment permissions: Read/Execute
_text           segment para public 'CODE' use32
        assume cs:_text
        ;org 401000h
        assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing
        push    esi
```

IDA Pro Dump



```
.text:00401390
.text:00401390                       ; =============== S U B R O U T I N E =======================================
.text:00401390
.text:00401390
.text:00401390                       sub_401390      proc near             ; CODE XREF: sub_43E84EEMj
.text:00401390
.text:00401390                       arg_0           = dword ptr  4
.text:00401390
.text:00401390 8B 4C 24 04                   mov     ecx, [esp+arg_0]
.text:00401394 B8 1F CD 98 AE                mov     eax, 0AE98CD1Fh
.text:00401399 F7 E1                         mul     ecx
.text:0040139B C1 EA 1E                      shr     edx, 1Eh
.text:0040139E 69 D2 FA C9 D6 5D             imul    edx, 5DD6C9FAh
.text:004013A4 56                            push    esi
.text:004013A5 57                            push    edi
.text:004013A6 8B F9                         mov     edi, ecx
.text:004013A8 2B FA                         sub     edi, edx
.text:004013AA B8 25 95 2A 16                mov     eax, 162A9525h
.text:004013AF F7 E1                         mul     ecx
.text:004013B1 8B C1                         mov     eax, ecx
.text:004013B3 2B C2                         sub     eax, edx
.text:004013B5 D1 E8                         shr     eax, 1
.text:004013B7 03 C2                         add     eax, edx
.text:004013B9 C1 E8 1C                      shr     eax, 1Ch
.text:004013BC 8B D0                         mov     edx, eax
.text:004013BE 69 D2 84 33 73 1D             imul    edx, 1D733384h
.text:004013C4 8B F1                         mov     esi, ecx
.text:004013C6 81 F6 45 CF 3F FE             xor     esi, 0FE3FCF45h
.text:004013CC 23 F1                         and     esi, ecx
.text:004013CE 8B C1                         mov     eax, ecx
.text:004013D0 81 E6 BA 3D C5 05             and     esi, 5C53DBAh
.text:004013D6 2B C2                         sub     eax, edx
.text:004013D8 85 FF                         test    edi, edi
.text:004013DA 74 08                         jz      short loc_4013E4
.text:004013DC 33 D2                         xor     edx, edx
.text:004013DE F7 F7                         div     edi
.text:004013E0 8B FA                         mov     edi, edx
.text:004013E2 EB 02                         jmp     short loc_4013E6
.text:004013E4                       ; ---------------------------------------------------------------------------
.text:004013E4
.text:004013E4                       loc_4013E4:                       ; CODE XREF: sub_401390+4ACANj
.text:004013E4 8B F8                         mov     edi, eax
.text:004013E6
.text:004013E6                       loc_4013E6:                       ; CODE XREF: sub_401390+52CANj
.text:004013E6 85 FF                         test    edi, edi
.text:004013E8 74 08                         jz      short loc_4013F2
```

Evaluation Function

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log(pij)$$

p_{ij} is the submitted probability of sample i is class j
y_{ij}=1 if sample i is class j,
y_{ij}=0 for others

Evaluation Example

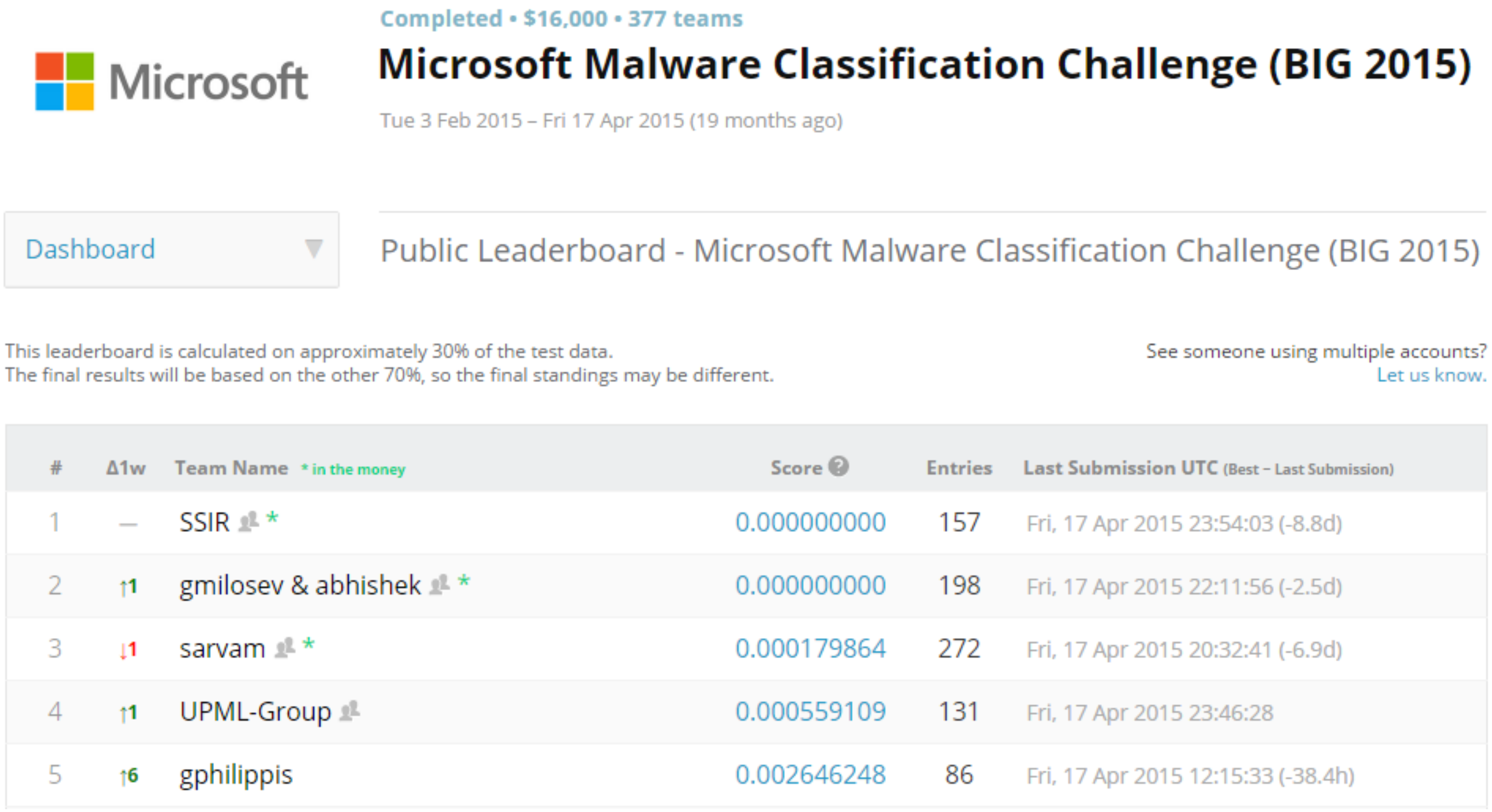
- Submission

```
00000000,0.5,0.5,0,0,0,0,0,0,0,0
00000001,0,0,0.5,0.5,0,0,0,0,0,0
00000002,0,0,1,0,0,0,0,0,0,0
```

- logloss = - (log(0.5)+log(0)+log(1))/3
  - log(0) => log(1e-15)

Leader Board

- Public vs. Private



Microsoft

Completed • $16,000 • 377 teams

Microsoft Malware Classification Challenge (BIG 2015)

Tue 3 Feb 2015 – Fri 17 Apr 2015 (19 months ago)

Dashboard ▼

Public Leaderboard - Microsoft Malware Classification Challenge (BIG 2015)

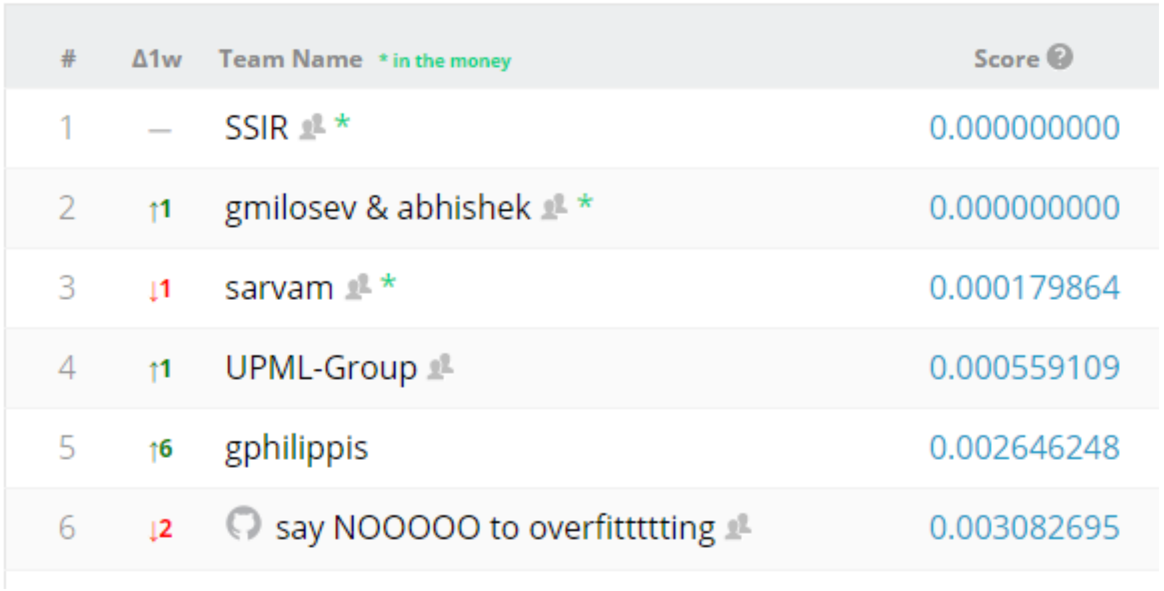
This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

See someone using multiple accounts?
Let us know.

#	Δ1w	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	—	SSIR *	0.000000000	157	Fri, 17 Apr 2015 23:54:03 (-8.8d)
2	↑1	gmilosev & abhishek *	0.000000000	198	Fri, 17 Apr 2015 22:11:56 (-2.5d)
3	↓1	sarvam *	0.000179864	272	Fri, 17 Apr 2015 20:32:41 (-6.9d)
4	↑1	UPML-Group	0.000559109	131	Fri, 17 Apr 2015 23:46:28
5	↑6	gphilippis	0.002646248	86	Fri, 17 Apr 2015 12:15:33 (-38.4h)

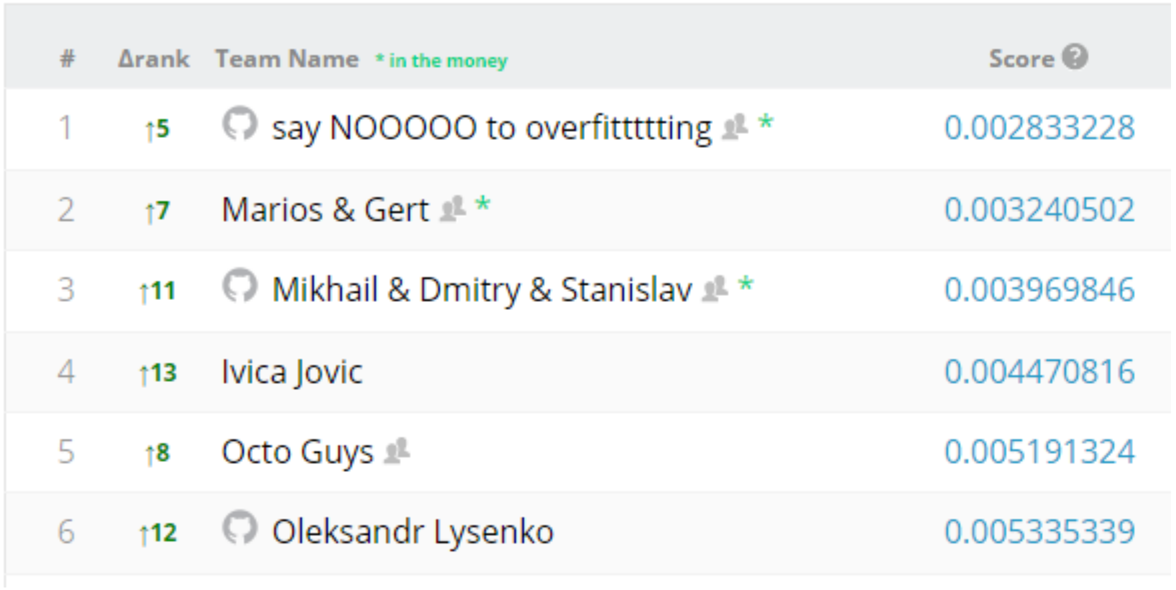
Leader Board

Public



#	Δ1w	Team Name * in the money	Score
1	—	SSIR *	0.000000000
2	↑1	gmilosev & abhishek *	0.000000000
3	↓1	sarvam *	0.000179864
4	↑1	UPML-Group	0.000559109
5	↑6	gphilippis	0.002646248
6	↓2	say NOOOOO to overfittttting	0.003082695

Private



#	Δrank	Team Name * in the money	Score
1	↑5	say NOOOOO to overfittttting *	0.002833228
2	↑7	Marios & Gert *	0.003240502
3	↑11	Mikhail & Dmitry & Stanislav *	0.003969846
4	↑13	Ivica Jovic	0.004470816
5	↑8	Octo Guys	0.005191324
6	↑12	Oleksandr Lysenko	0.005335339

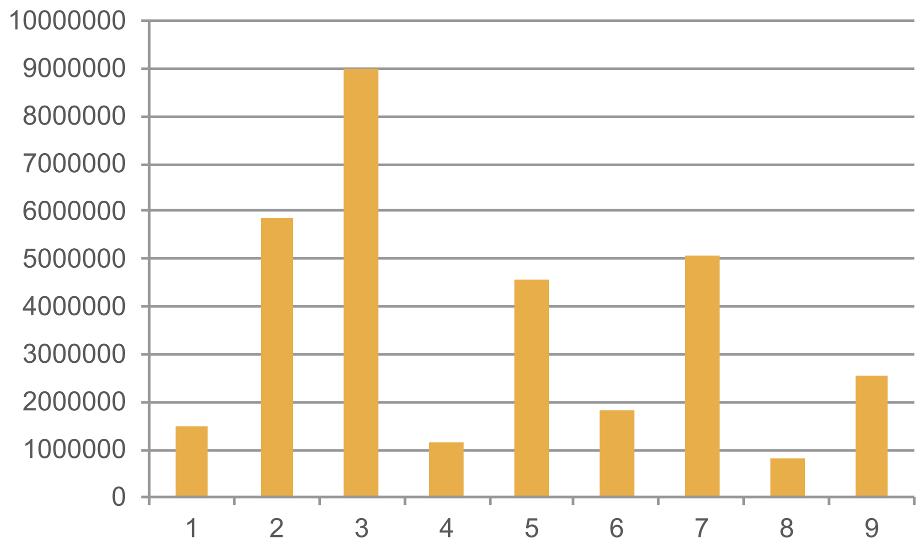
HOW TO SOLVE IT?

First Feature Set

- Binary size
- Hex count
- String length stats
- TLSH

Binary Size

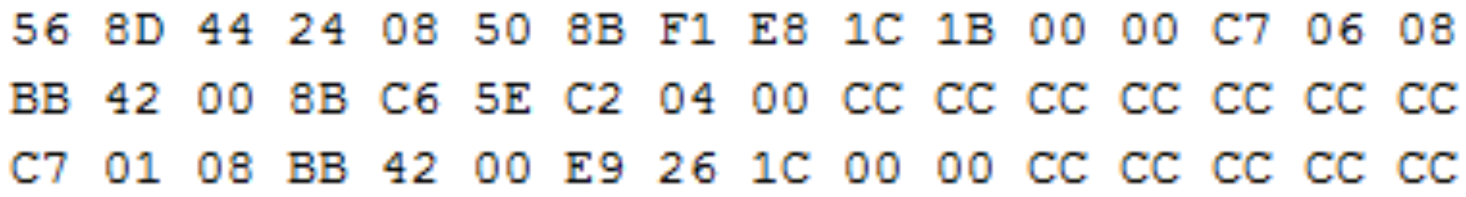
	Category	Avg. Size
1	Ramnit	1482170
2	Lollipop	5829530
3	Kelihos_ver3	8982630
4	Vundo	1120950
5	Simda	4552330
6	Tracur	1801150
7	Kelihos_ver1	5051900
8	Obfuscator.ACY	827118
9	Gatak	2555070



Hex Count

- Count of HEX
- 00, 01, 02,…, FE, FF, ??
- 257 dimensions

- 1-gram

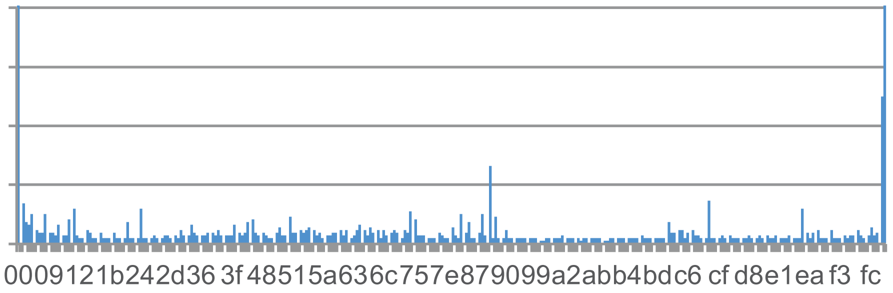


```
56 8D 44 24 08 50 8B F1 E8 1C 1B 00 00 C7 06 08
BB 42 00 8B C6 5E C2 04 00 CC CC CC CC CC CC CC
C7 01 08 BB 42 00 E9 26 1C 00 00 CC CC CC CC CC
```

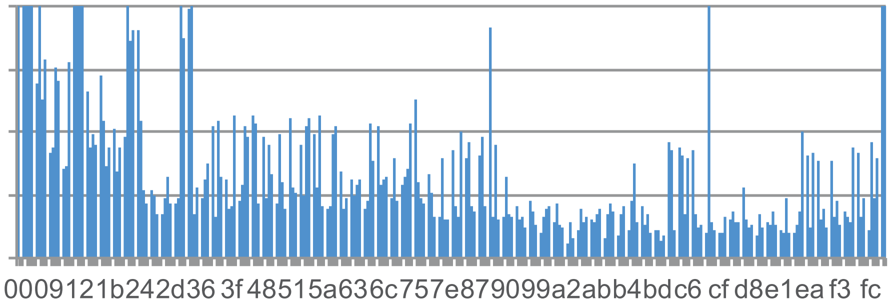
Hex Count Distribution

Protect against tomorrow's threats | Machine Learning

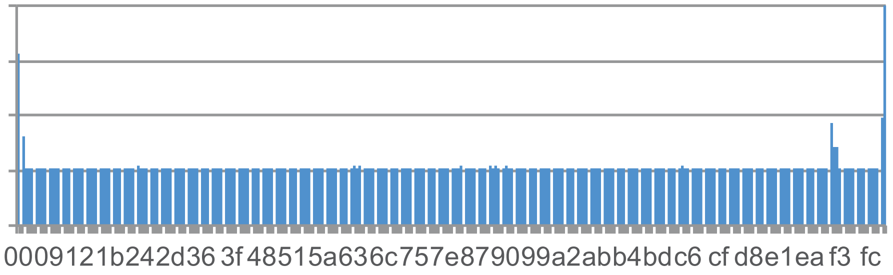
1. Ramnit



2. Lollipop

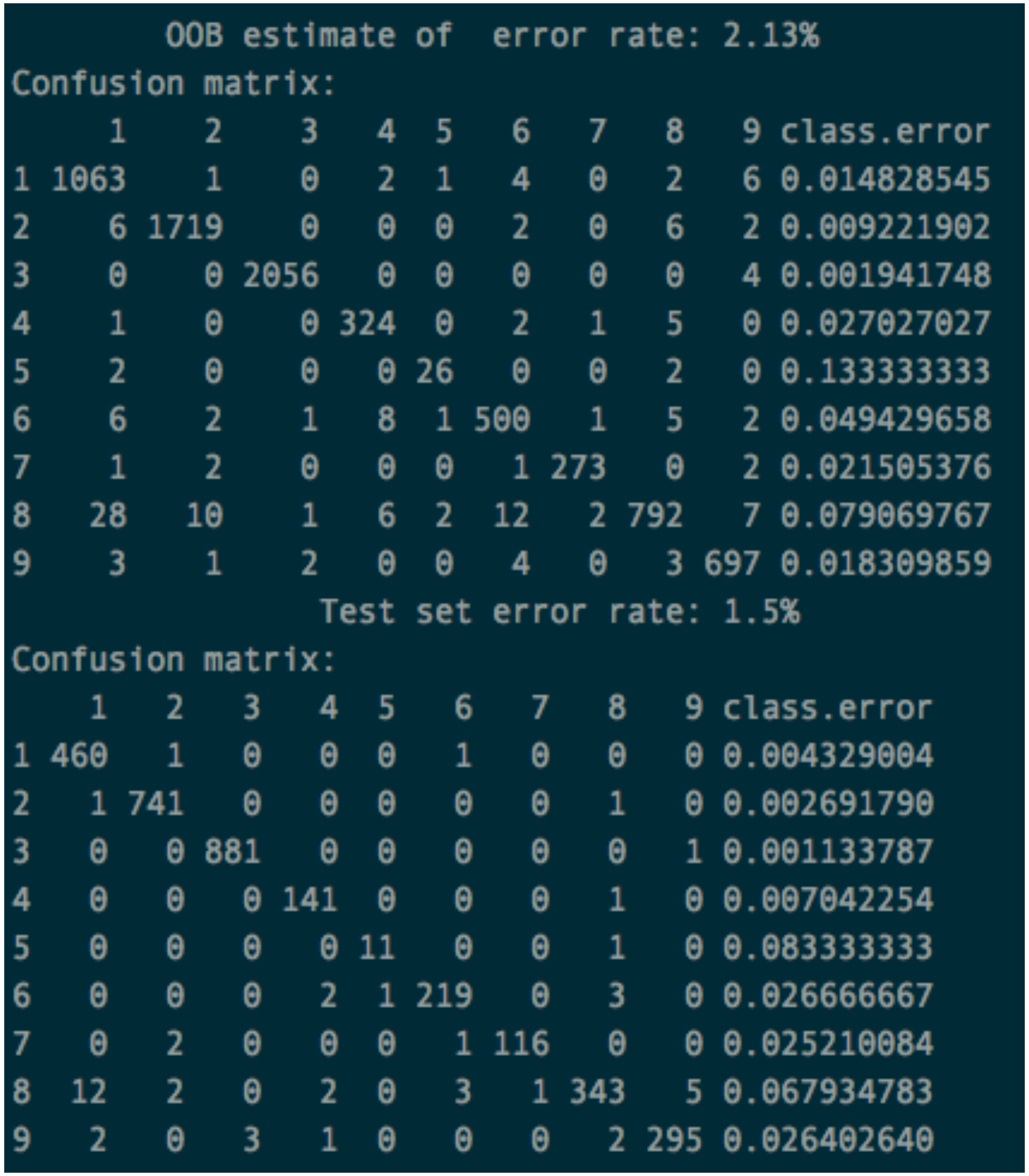


3. Kelihos_v3



Hex Count Confusion Matrix?

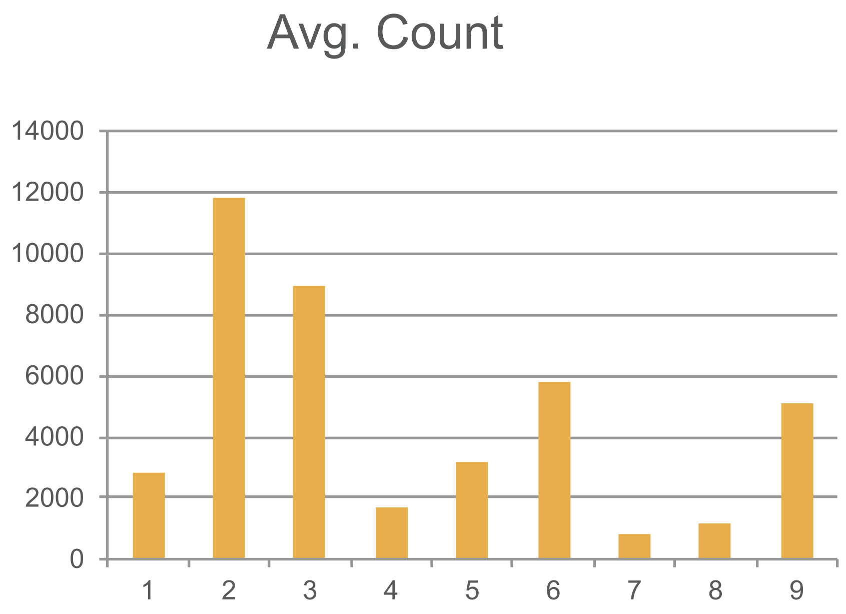
Protect against tomorrow's threats | Machine Learning

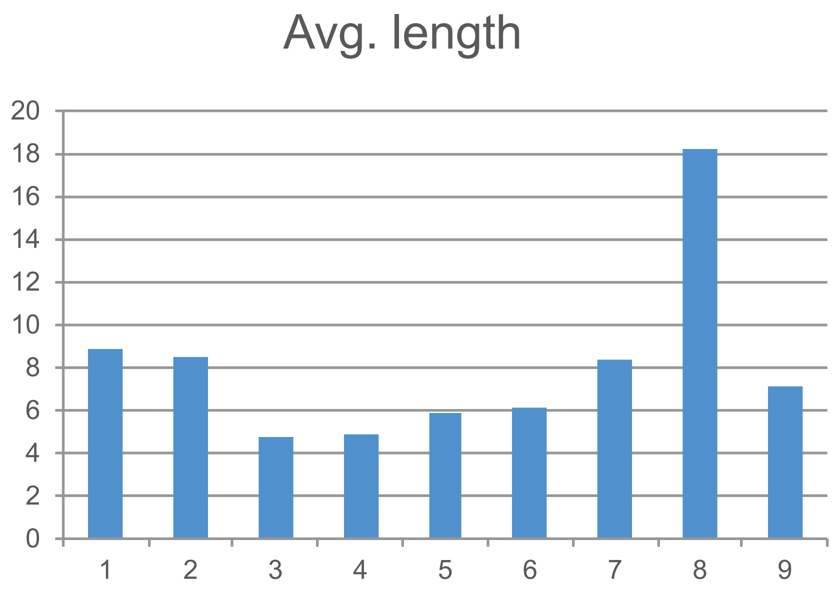


```
        OOB estimate of  error rate: 2.13%
Confusion matrix:
     1    2    3   4  5   6   7   8   9 class.error
1 1063    1    0   2  1   4   0   2   6 0.014828545
2    6 1719    0   0  0   2   0   6   2 0.009221902
3    0    0 2056   0  0   0   0   0   4 0.001941748
4    1    0    0 324  0   2   1   5   0 0.027027027
5    2    0    0   0 26   0   0   2   0 0.133333333
6    6    2    1   8  1 500   1   5   2 0.049429658
7    1    2    0   0  0   1 273   0   2 0.021505376
8   28   10    1   6  2  12   2 792   7 0.079069767
9    3    1    2   0  0   4   0   3 697 0.018309859
                Test set error rate: 1.5%
Confusion matrix:
    1   2   3   4  5   6   7   8   9 class.error
1 460   1   0   0  0   1   0   0   0 0.004329004
2   1 741   0   0  0   0   0   1   0 0.002691790
3   0   0 881   0  0   0   0   0   1 0.001133787
4   0   0   0 141  0   0   0   1   0 0.007042254
5   0   0   0   0 11   0   0   1   0 0.083333333
6   0   0   0   2  1 219   0   3   0 0.026666667
7   0   2   0   0  0   1 116   0   0 0.025210084
8  12   2   0   2  0   3   1 343   5 0.067934783
9   2   0   3   1  0   0   0   2 295 0.026402640
```

String Stats

- String: printable chars where length > 4
- String count, avg. length, max length





TLSH

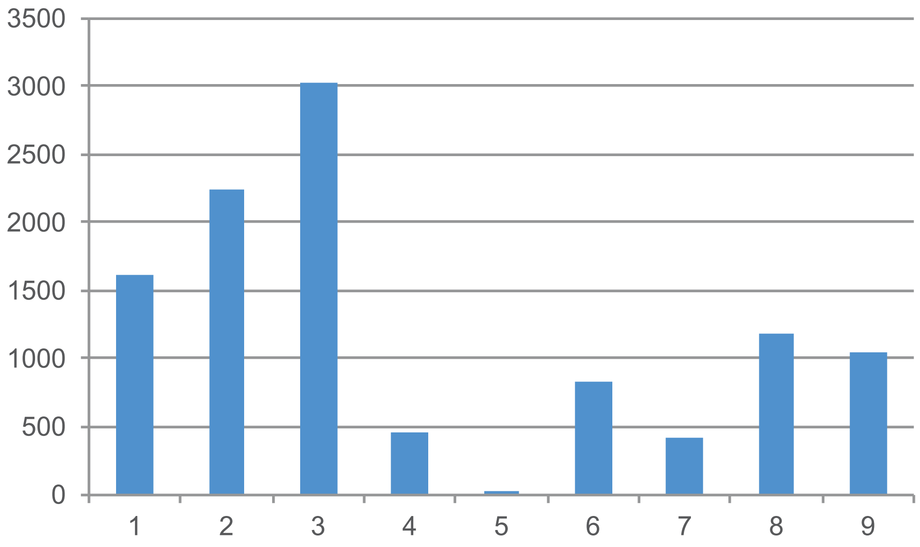
- Trend Micro Locality Sensitive Hash
- Fuzzy matching for similarity comparison
- Get the most similar class by voting of Top5 similar files from training data

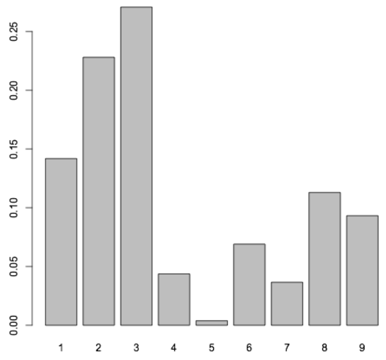
	TLSH
Text 1	E491A51FA380022245B021E9770F3A6FF706C1780365C631581EF6263731EAA87F96EE
Text 2	5B91940FA380026245B021A9771F7A6FF706C1780765C671981EF6263731EAA87F96DE

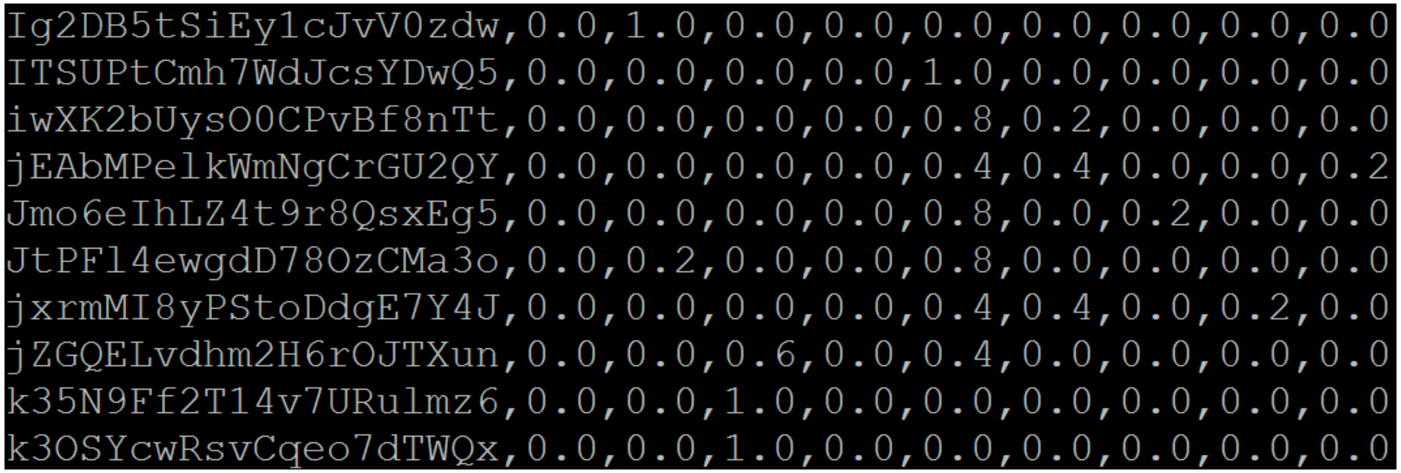
The distance between Text 1 and Text 2

distance(Text1,Text2) = 11

TLSH







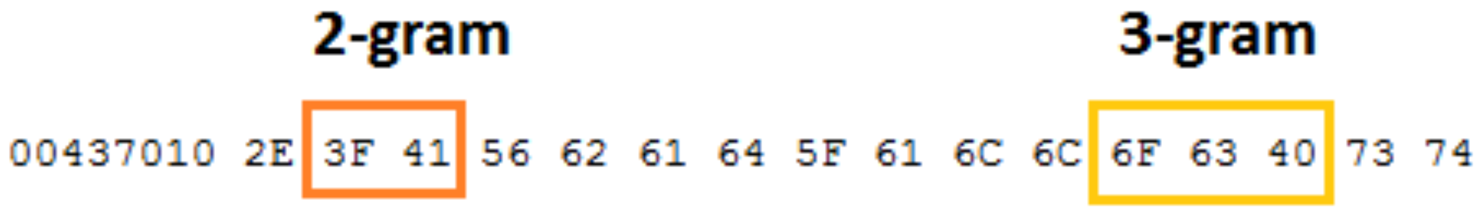
```
Ig2DB5tSiEy1cJvV0zdw,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
ITSUPtCmh7WdJcsYDwQ5,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0
iwXK2bUysO0CPvBf8nTt,0.0,0.0,0.0,0.0,0.8,0.2,0.0,0.0,0.0
jEAbMPelkWmNgCrGU2QY,0.0,0.0,0.0,0.0,0.4,0.4,0.0,0.0,0.2
Jmo6eIhLZ4t9r8QsxEg5,0.0,0.0,0.0,0.0,0.8,0.0,0.2,0.0,0.0
JtPFl4ewgdD78OzCMa3o,0.0,0.2,0.0,0.0,0.8,0.0,0.0,0.0,0.0
jxrmMI8yPStoDdgE7Y4J,0.0,0.0,0.0,0.0,0.4,0.4,0.0,0.2,0.0
jZGQELvdhm2H6rOJTXun,0.0,0.0,0.6,0.0,0.4,0.0,0.0,0.0,0.0
k35N9Ff2T14v7URulmz6,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0
k3OSYcwRsvCqeo7dTWQx,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0
```

More Features

- HEX n-gram
- API call
- Import table
- Instruction
- Domain knowledge

2-gram/3-gram

- 2-gram: (256+1)^2= 66,049
- 3-gram: (256+1)^3= 16,974,593



```
00437010 2E 3F 41 56 62 61 64 5F 61 6C 6C 6F 63 40 73 74
```

HEX 2-gram/3-gram

- Important 2-gram Example
- Feature selection: reduce feature size

BiHEX	1. Ramnit	2. Lollipop	3. Kelihos_ver3
97 86	1.412	2.047	26.651
4b e5	1.718	0.722	13.201
f7 99	1.746	12.539	13.606
75 08	228.09	288.78	13.168
4e 47	146.318	12.159	13.512

API Call

- API used in PE

API	1. Ramnit	2. Lollipop	3. Kelihos_ver3
IsWindow()	0.164	0.257	0.987
DispatchMessageA()	0.159	0.845	0.987
GetCommandLineA()	0.355	0.981	0.025
DllEntryPoint()	0.656	0	0
GetIconInfo()	0.023	0	0.936

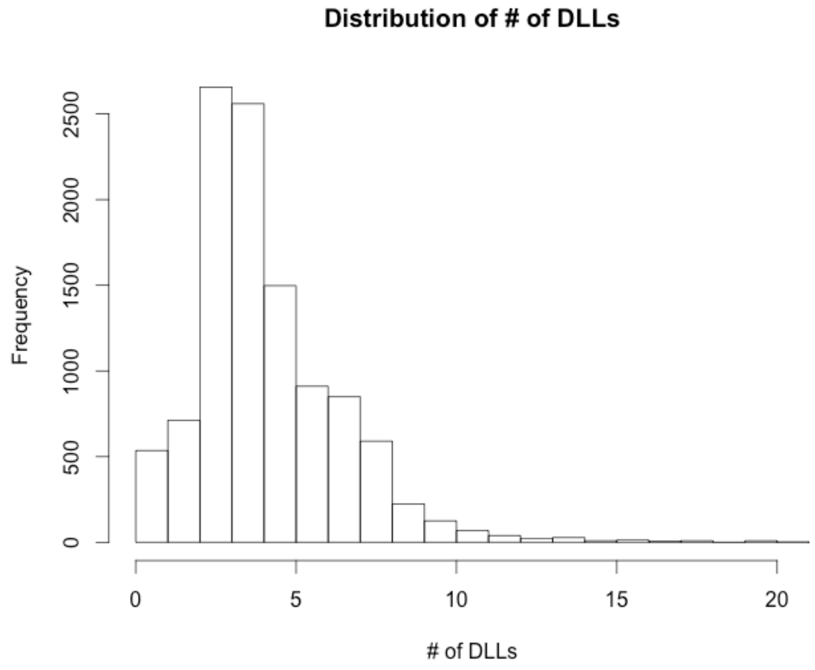
Import Table

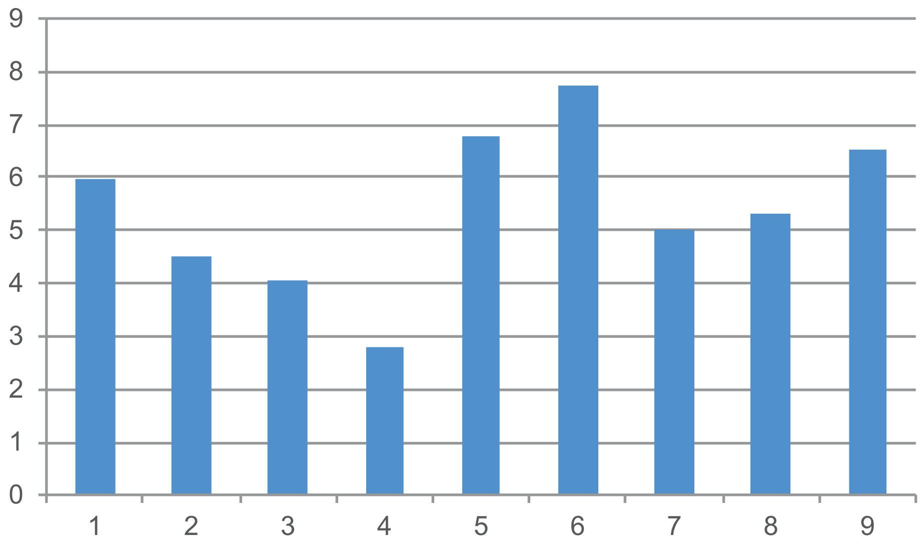
- A lookup table for calling functions in other module

1. Ramnit	2. Lollipop	3. Kelihos_ver3
KERNEL32.dll	KERNEL32.dll	USER32.dll
USER32.dll	USER32.dll	KERNEL32.dll
ADVAPI32.dll	ADVAPI32.dll	MSASN1.dll
ole32.dll	OPENGL32.dll	UXTHEME.dll
OLEAUT32.dll	OLEAUT32.dll	CLBCATQ.dll
msvcrt.dll	GDI32.dll	DPNET.dll
APPHELP.dll	WS2_32.dll	NTSHRUI.dll

Other Info in Import Table

- Number of distinct DLL





Instruction Frequency

Protect against tomorrow's threats | Machine Learning

- Very powerful

instruction	1. Ramnit	2. Lollipop	3. Kelihos_ver3
imul	86.768	2257.3	0.002
movzx	289.17	118.79	0
sbb	68.815	17.375	4.746
jnz	1154.8	154.57	7.842
mov	12336.6	7059.8	158.94

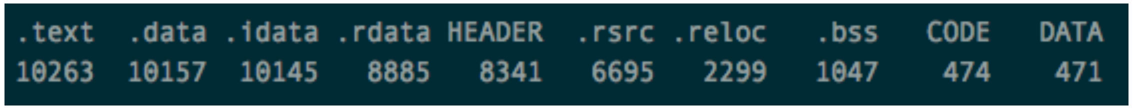
More Domain Knowledge

- Segment
- Packer
- Other type of binary

Segments

Protect against tomorrow's threats | Machine Learning

- Common segment name



```
.text  .data .idata .rdata HEADER  .rsrc .reloc   .bss   CODE   DATA
10263  10157  10145   8885   8341   6695   2299   1047    474    471
```

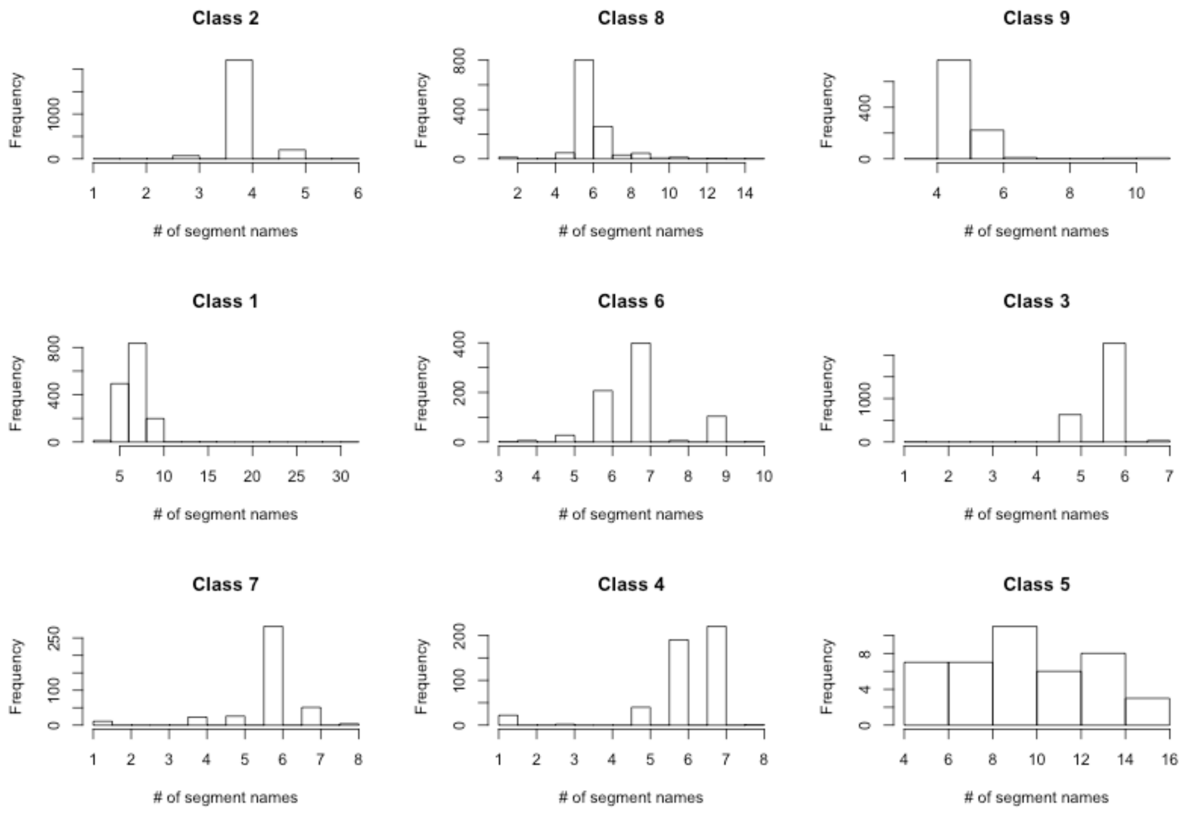
- Unique segment name

1. Ramnit	2. Lollipop	3. Kelihos_ver3
_data	_text	_rdata
_text	_data	_text
_rdata	_rdata	_data
_bss	_zenc	
_gnu_deb		
_tls		

Other Info from Segments

Protect against tomorrow's threats | Machine Learning

- Number of Segments

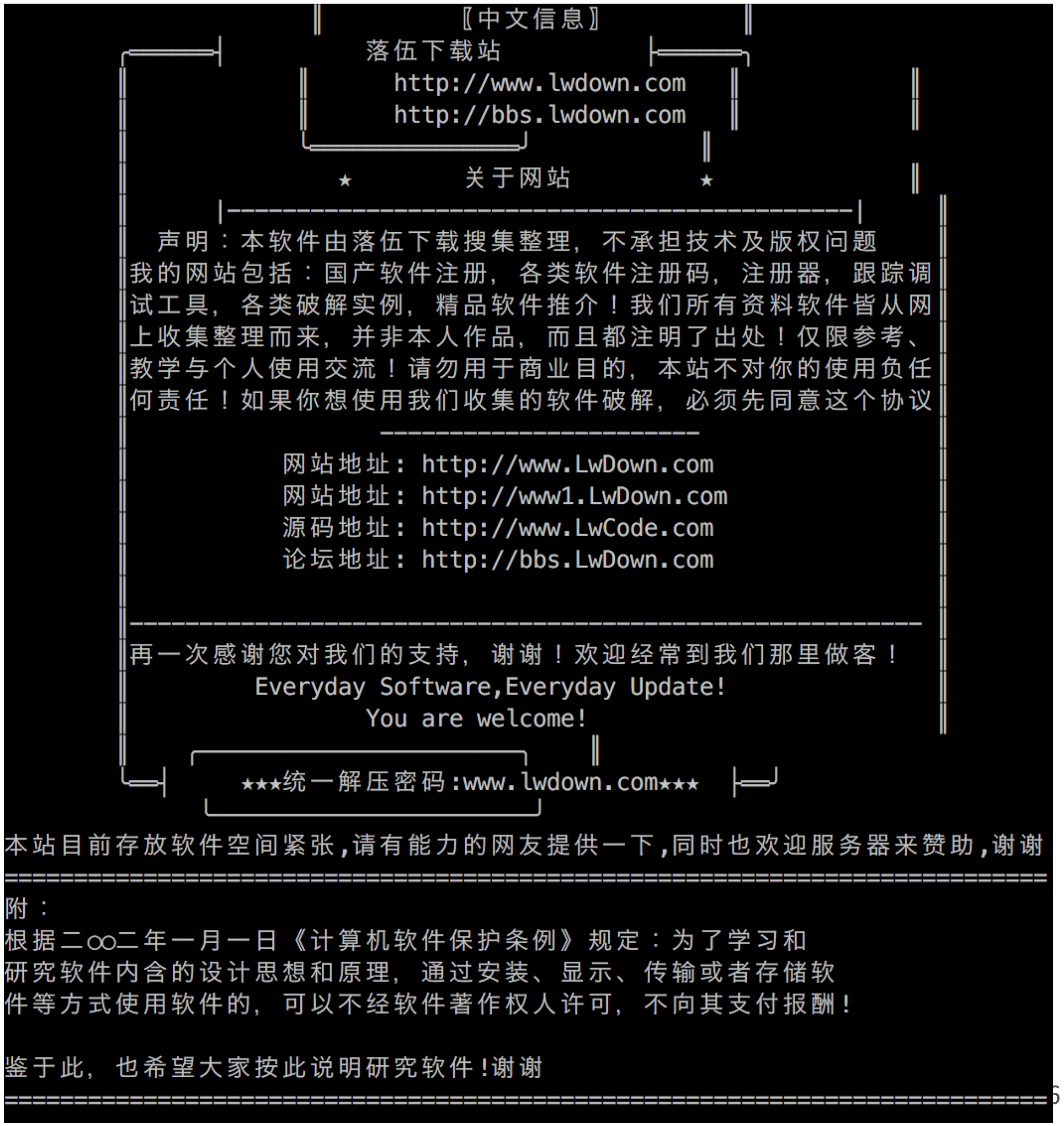


Packer

- Common segment name of Packer
- UPX0/UPX1 only in class 8. Obfuscator.ACY

Other Type of Binary

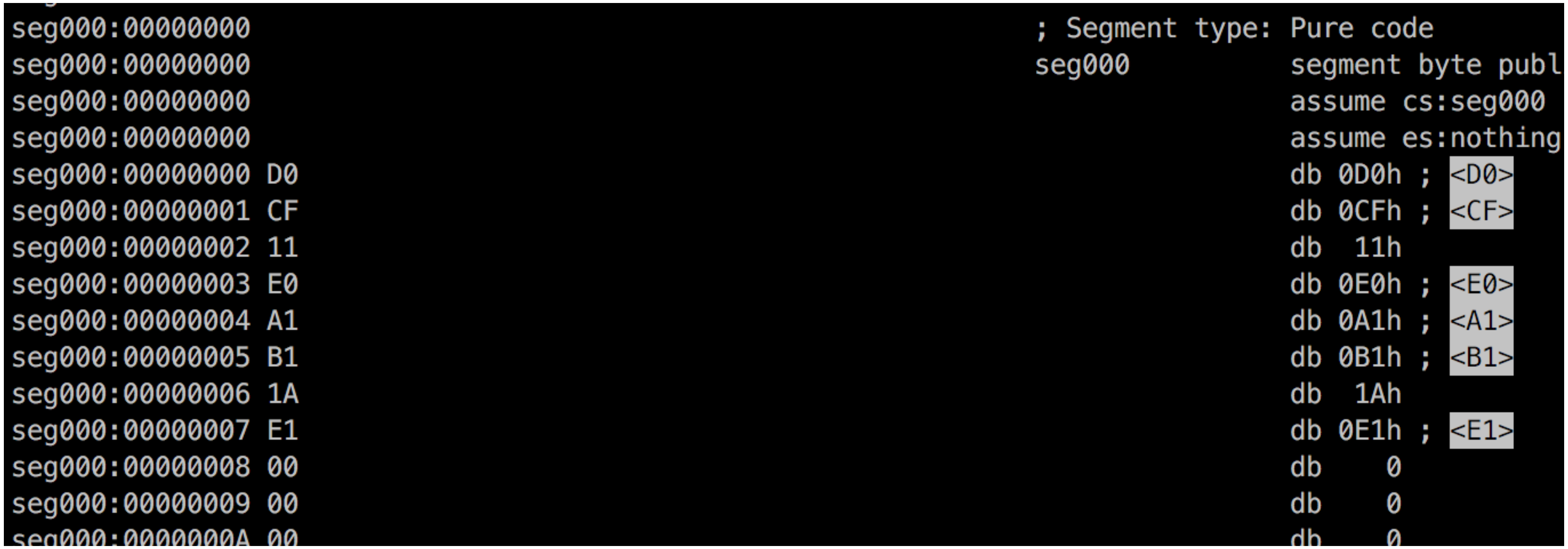
- RAR files



```
                     ║          〖中文信息〗             ║
      ╒══════════╡       落伍下载站            ╞══════════╕
      │           ║      http://www.lwdown.com       ║           ║
      │           ║      http://bbs.lwdown.com       ║           ║
      │           ╘═══════════════════════╛        ║
      │        ★            关于网站            ★            ║
      │   |--------------------------------------------------------|    ║
      │ 声明：本软件由落伍下载搜集整理，不承担技术及版权问题     ║
      │我的网站包括：国产软件注册，各类软件注册码，注册器，跟踪调║
      │试工具，各类破解实例，精品软件推介！我们所有资料软件皆从网║
      │上收集整理而来，并非本人作品，而且都注明了出处！仅限参考、║
      │教学与个人使用交流！请勿用于商业目的，本站不对你的使用负任║
      │何责任！如果你想使用我们收集的软件破解，必须先同意这个协议║
      │                  ------------------------                 ║
      │            网站地址：http://www.LwDown.com               ║
      │            网站地址：http://www1.LwDown.com              ║
      │            源码地址：http://www.LwCode.com               ║
      │            论坛地址：http://bbs.LwDown.com               ║
      │                                                           ║
      │-------------------------------------------------------------║
      │再一次感谢您对我们的支持，谢谢！欢迎经常到我们那里做客！  ║
      │          Everyday Software,Everyday Update!               ║
      │                  You are welcome!                         ║
      │      ┌────────────────────────┐  ║
      ╘══╡    ★★★统一解压密码:www.lwdown.com★★★   ╞══╛
              └────────────────────────┘
本站目前存放软件空间紧张,请有能力的网友提供一下,同时也欢迎服务器来赞助,谢谢
==============================================================================
附：
根据二○○二年一月一日《计算机软件保护条例》规定：为了学习和
研究软件内含的设计思想和原理，通过安装、显示、传输或者存储软
件等方式使用软件的，可以不经软件著作权人许可，不向其支付报酬！

鉴于此，也希望大家按此说明研究软件!谢谢
==============================================================================
```

Other Type of Binary

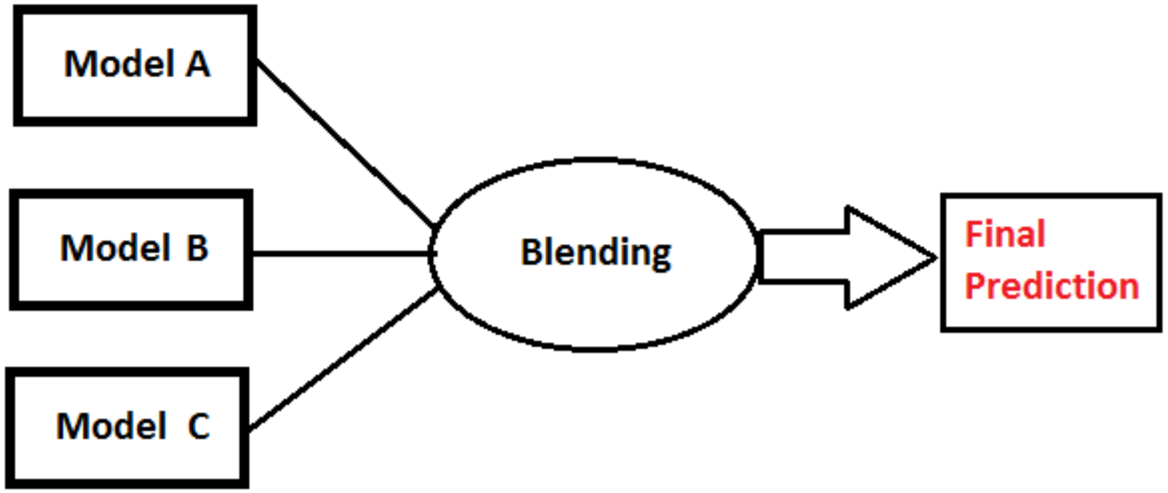
- Microsoft Office files



```
seg000:00000000                                           ; Segment type: Pure code
seg000:00000000                                           seg000          segment byte publ
seg000:00000000                                                           assume cs:seg000
seg000:00000000                                                           assume es:nothing
seg000:00000000 D0                                                        db 0D0h ; <D0>
seg000:00000001 CF                                                        db 0CFh ; <CF>
seg000:00000002 11                                                        db  11h
seg000:00000003 E0                                                        db 0E0h ; <E0>
seg000:00000004 A1                                                        db 0A1h ; <A1>
seg000:00000005 B1                                                        db 0B1h ; <B1>
seg000:00000006 1A                                                        db  1Ah
seg000:00000007 E1                                                        db 0E1h ; <E1>
seg000:00000008 00                                                        db    0
seg000:00000009 00                                                        db    0
seg000:0000000A 00                                                        db    0
```

Ensemble: Linear Blending

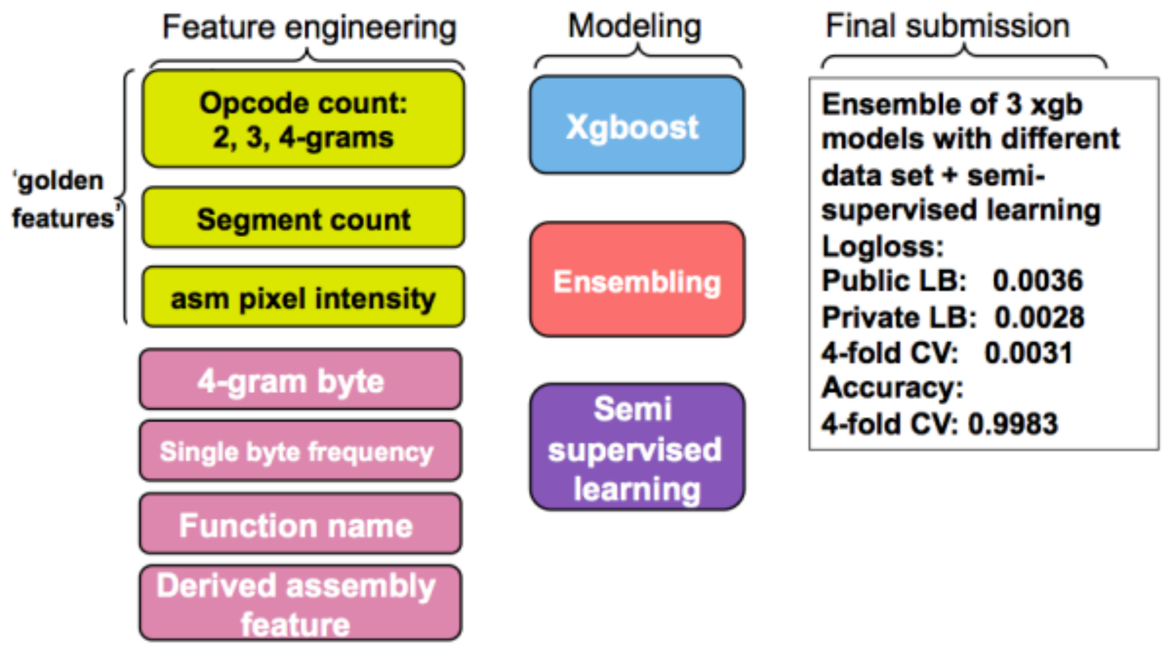
- Combine the result from several models
- Vote of models



WORK OF WINNING TEAM

Features

- Instruction n-gram
- ASM pixel map

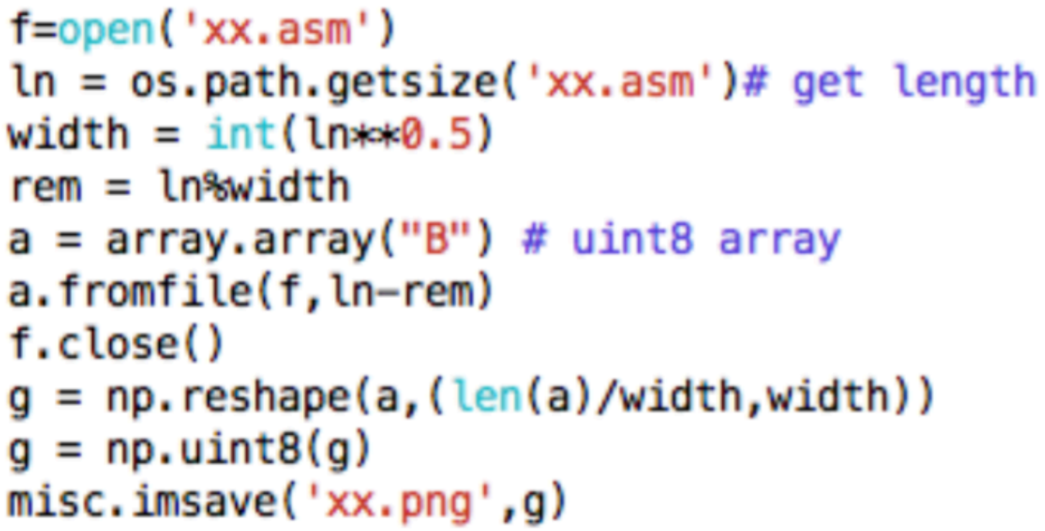


Ensemble of 3 xgb models with different data set + semi-supervised learning
Logloss:
Public LB: 0.0036
Private LB: 0.0028
4-fold CV: 0.0031
Accuracy:
4-fold CV: 0.9983

http://blog.kaggle.com/2015/05/26/microsoft-malware-winners-interview-1st-place-no-to-overfitting/

Features

- ASM pixel map (intensity of first 1000 bytes)



```
f=open('xx.asm')
ln = os.path.getsize('xx.asm')# get length
width = int(ln**0.5)
rem = ln%width
a = array.array("B") # uint8 array
a.fromfile(f,ln-rem)
f.close()
g = np.reshape(a,(len(a)/width,width))
g = np.uint8(g)
misc.imsave('xx.png',g)
```

xgboost

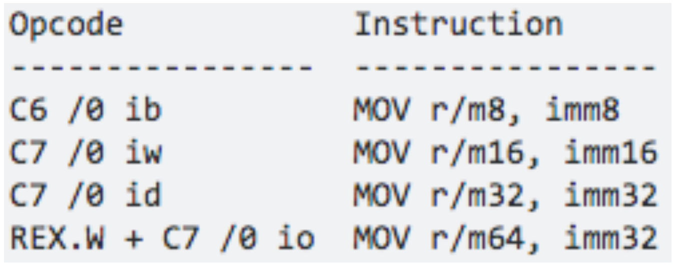
- Gradient boosting package
- Widely used in Kaggle competition

dmlc XGBoost

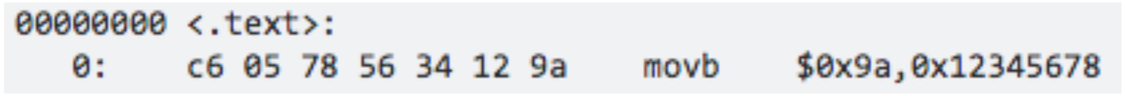
CONCLUSION

Physical Meaning of Features

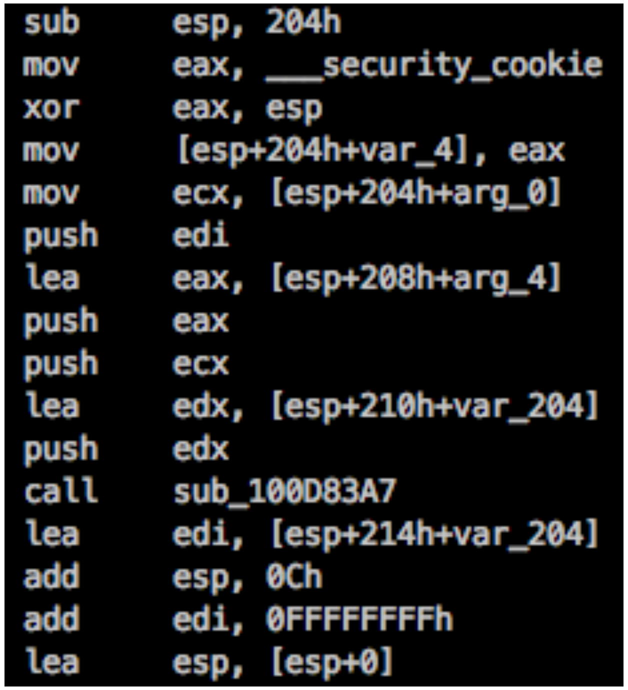
- Hex n-gram
  - Opcode + imm/addr
- Instruction n-gram
  - Opcode



```
Opcode              Instruction
----------------    ----------------
C6 /0 ib            MOV r/m8, imm8
C7 /0 iw            MOV r/m16, imm16
C7 /0 id            MOV r/m32, imm32
REX.W + C7 /0 io    MOV r/m64, imm32
```



```
00000000 <.text>:
   0:    c6 05 78 56 34 12 9a     movb    $0x9a,0x12345678
```



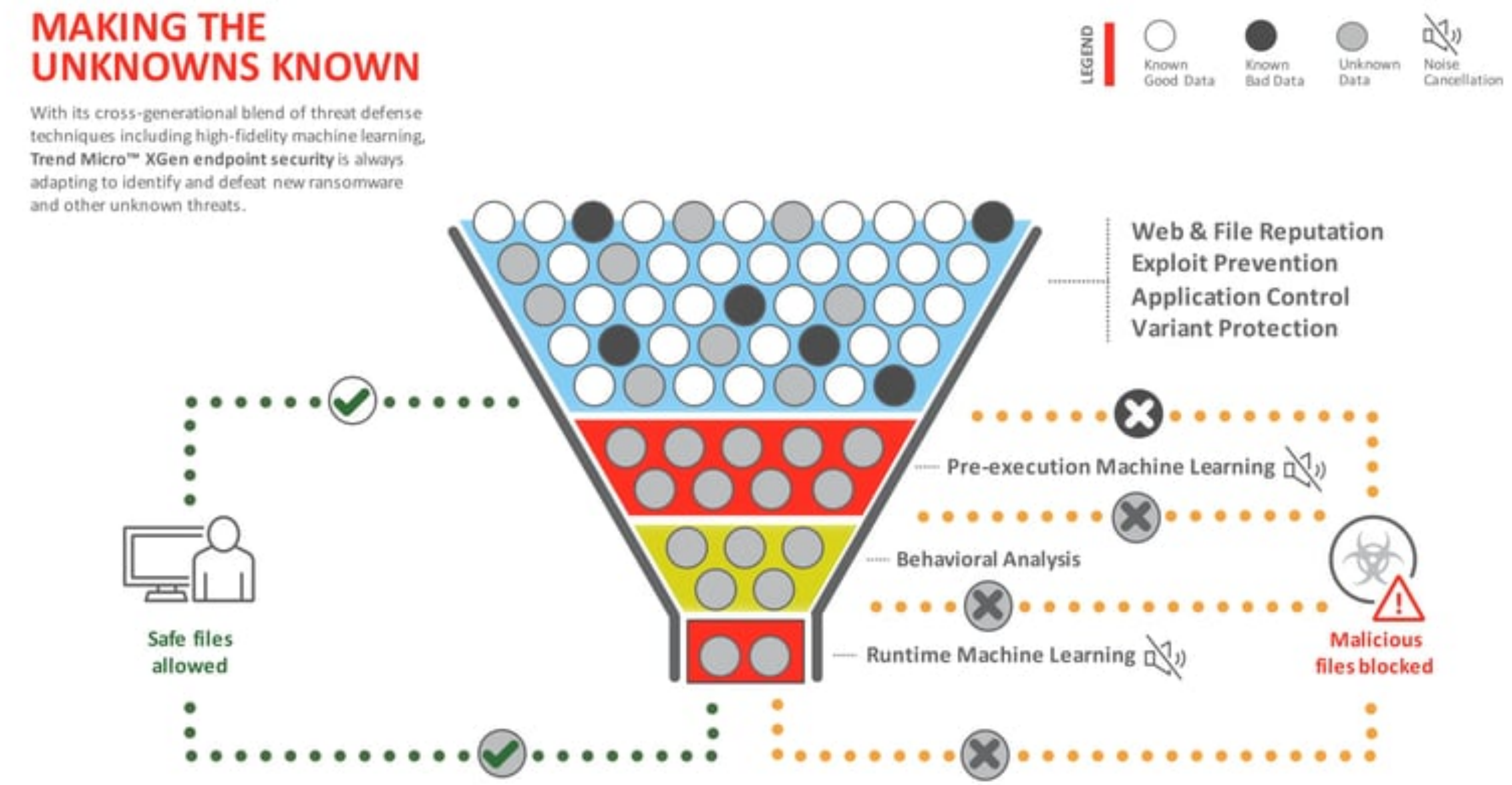
```
sub     esp, 204h
mov     eax, ___security_cookie
xor     eax, esp
mov     [esp+204h+var_4], eax
mov     ecx, [esp+204h+arg_0]
push    edi
lea     eax, [esp+208h+arg_4]
push    eax
push    ecx
lea     edx, [esp+210h+var_204]
push    edx
call    sub_100D83A7
lea     edi, [esp+214h+var_204]
add     esp, 0Ch
add     edi, 0FFFFFFFFh
lea     esp, [esp+0]
```

Happy Ending?

- Welcome to the real world!
- New malware family
- Mis-labelling

- Mechanism to mitigate the issues.

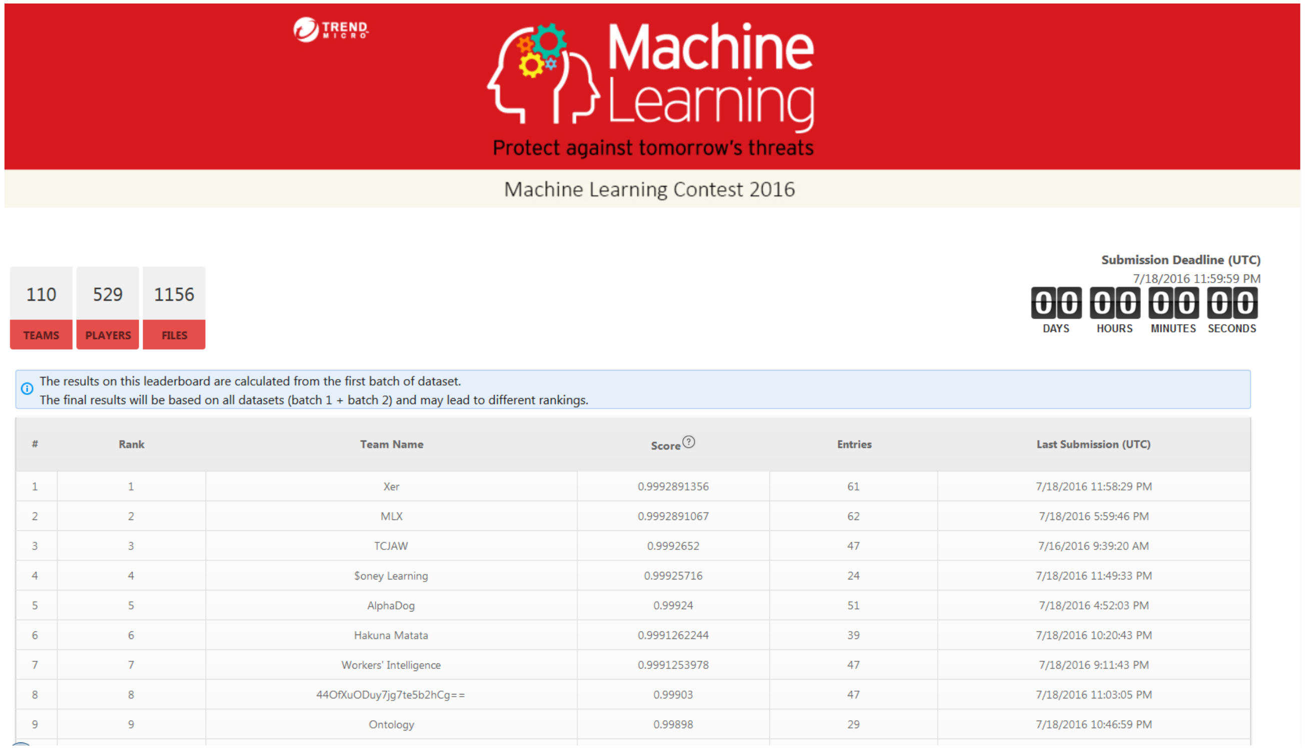
Trend Micro XGen


MAKING THE UNKNOWNS KNOWN
With its cross-generational blend of threat defense techniques including high-fidelity machine learning, Trend Micro™ XGen endpoint security is always adapting to identify and defeat new ransomware and other unknown threats.
LEGEND
Known Good Data
Known Bad Data
Unknown Data
Noise Cancellation
Web & File Reputation
Exploit Prevention
Application Control
Variant Protection
Pre-execution Machine Learning
Behavioral Analysis
Runtime Machine Learning
Safe files allowed
Malicious files blocked

Trend Micro ML Contest

Protect against tomorrow's threats | Machine Learning

- Malware Identification Challenge
- 134 teams, 626 players, from 6+ countries
- Real-time scoring



TREND MICRO

Machine Learning

Protect against tomorrow's threats

Machine Learning Contest 2016

110	529	1156
TEAMS	PLAYERS	FILES

Submission Deadline (UTC)
7/18/2016 11:59:59 PM

00	00	00	00
DAYS	HOURS	MINUTES	SECONDS

The results on this leaderboard are calculated from the first batch of dataset.
The final results will be based on all datasets (batch 1 + batch 2) and may lead to different rankings.

#	Rank	Team Name	Score	Entries	Last Submission (UTC)
1	1	Xer	0.9992891356	61	7/18/2016 11:58:29 PM
2	2	MLX	0.9992891067	62	7/18/2016 5:59:46 PM
3	3	TCJAW	0.9992652	47	7/16/2016 9:39:20 AM
4	4	$oney Learning	0.99925716	24	7/18/2016 11:49:33 PM
5	5	AlphaDog	0.99924	51	7/18/2016 4:52:03 PM
6	6	Hakuna Matata	0.9991262244	39	7/18/2016 10:20:43 PM
7	7	Workers' Intelligence	0.9991253978	47	7/18/2016 9:11:43 PM
8	8	44OfXuODuy7jg7te5b2hCg==	0.99903	47	7/18/2016 11:03:05 PM
9	9	Ontology	0.99898	29	7/18/2016 10:46:59 PM

How to Improve Model

- Use domain knowledge
  - Unpack, unzip ...
- Improve feature representation
  - Distinctive features for classes which you don't do well
- Regulate overfitting

How to Improve Model

- Find which items cannot be covered by model
- Adjust current features
- Find new features
- Tuning algorithm parameters
- Use different algorithm
- Ensemble/Blending

Local Library vs. Cloud Platform

Cloud platform is not necessarily easier

- Glue & Integration
  - Data (pre-)processing
  - Model training / prediction
  - Evaluation
- Diversity of ML algorithms
- Parameter tuning

Protect against tomorrow's threats | Machine Learning

THANK YOU