Malware Classification

Andrea Mambretti

mbr@ccs.neu.edu

CS 7775

Problem?

- Hundreds of malware samples discovered every day
- Reverse engineering not always possible
- Manual categorization is
  - Time consuming
  - Expensive
  - Subject to human mistakes

Malware

- Software that is intended to damage or disable computers puter systems
- Divided in families (e.g Zeus, S
- Each family cont centralized, updated versi
- **Famil stics**

CLASSIFICATION

IDEAS?

Yeah...what about the data though?

- Good malware collections are hard to find
- Encryption and packing do not help in the task
- Virus Total and similar?
    - Hard to distinguish new and old samples
    - Miss-classification can affect the ground truth

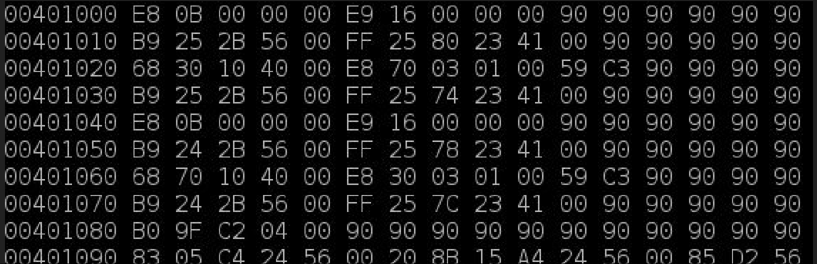
kaggle™

Microsoft

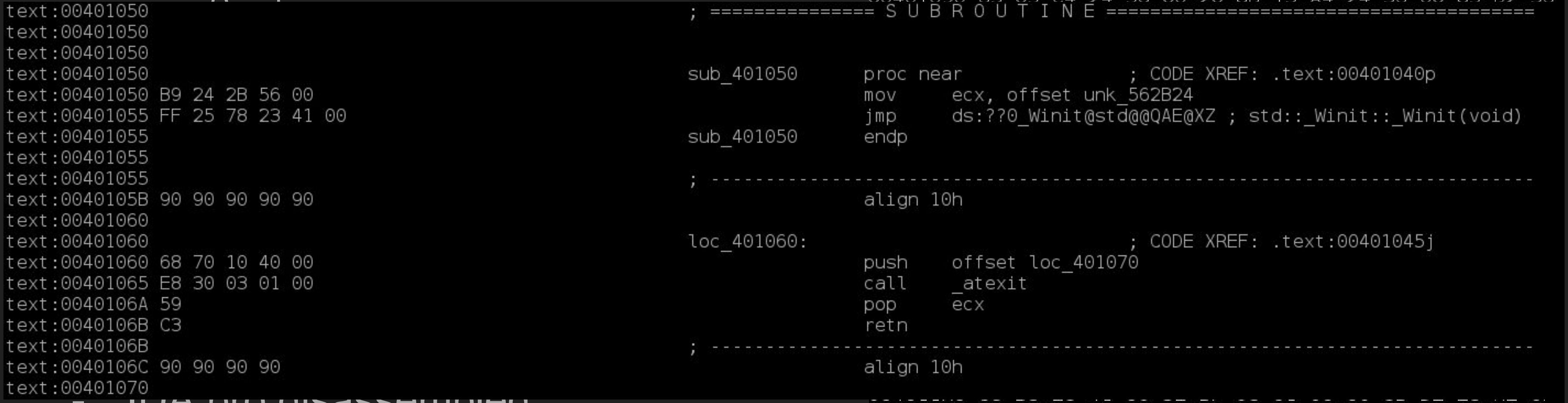
Microsoft Malware Classification Challenge

- 9 Families
    - Ramnit
    - Lollipop
    - Kelihos_ver3
    - Si

- IDA pro disassembled



```
00401000 E8 0B 00 00 00 E9 16 00 00 00 90 90 90 90 90 90
00401010 B9 25 2B 56 00 FF 25 80 23 41 00 90 90 90 90 90
00401020 68 30 10 40 00 E8 70 03 01 00 59 C3 90 90 90 90
00401030 B9 25 2B 56 00 FF 25 74 23 41 00 90 90 90 90 90
00401040 E8 0B 00 00 00 E9 16 00 00 00 90 90 90 90 90 90
00401050 B9 24 2B 56 00 FF 25 78 23 41 00 90 90 90 90 90
00401060 68 70 10 40 00 E8 30 03 01 00 59 C3 90 90 90 90
00401070 B9 24 2B 56 00 FF 25 7C 23 41 00 90 90 90 90 90
00401080 B0 9F C2 04 00 90 90 90 90 90 90 90 90 90 90 90
00401090 83 05 C4 24 56 00 20 8B 15 A4 24 56 00 85 D2 56
004011B0 00 00 A3 E4 23 56 00 EB 0B 29 35 84 24 56 00 BF
```



```
text:00401050                                         ; =============== S U B R O U T I N E =======================================
text:00401050
text:00401050
text:00401050                                         sub_401050      proc near               ; CODE XREF: .text:00401040p
text:00401050 B9 24 2B 56 00                                          mov     ecx, offset unk_562B24
text:00401055 FF 25 78 23 41 00                                       jmp     ds:??0_Winit@std@@QAE@XZ ; std::_Winit::_Winit(void)
text:00401055                                         sub_401050      endp
text:00401055
text:00401055                                         ; ---------------------------------------------------------------------------
text:0040105B 90 90 90 90 90                                          align 10h
text:00401060
text:00401060                                         loc_401060:                             ; CODE XREF: .text:00401045j
text:00401060 68 70 10 40 00                                          push    offset loc_401070
text:00401065 E8 30 03 01 00                                          call    _atexit
text:0040106A 59                                                      pop     ecx
text:0040106B C3                                                      retn
text:0040106B                                         ; ---------------------------------------------------------------------------
text:0040106C 90 90 90 90                                             align 10h
text:00401070
```

Have we got features?

- Size of the bytes file
- Strings within the file
- Number of Basic Block
- Sections (e.g. .text, .idata, .rdata)
- # of calls
- # of mov
- # of jmps (e.g. jge, jmp, je etc.)
- # of pop
- # of push
- # of xor
- # of sub
- # of add

Dynamic features non-available due to missing PE header

Multi-class classifiers

- Support Vector Machines
- Random Forest
- Neural Networks

Total features vector of

410318 elements

Number of samples of

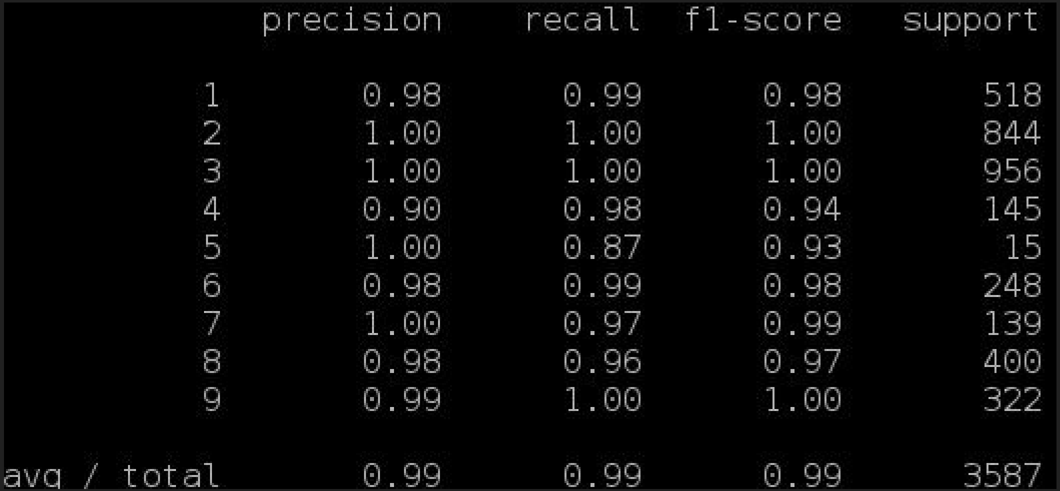
10868 elements

SVM Results

Best configuration found {'kernel': 'rbf', 'C': 100, 'gamma': 0.0001}

Test size 33%

Train size 66%

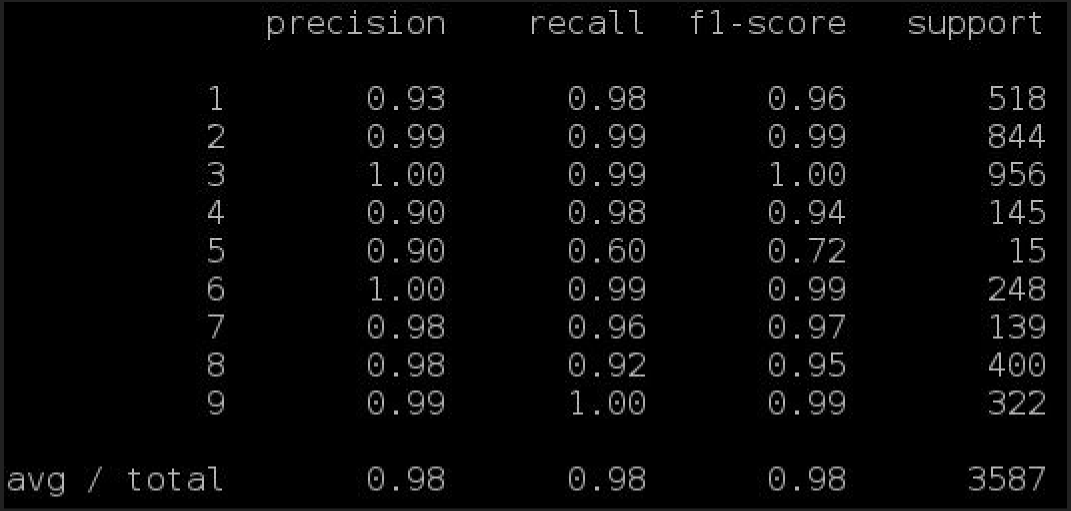


```
             precision    recall  f1-score   support

          1       0.98      0.99      0.98       518
          2       1.00      1.00      1.00       844
          3       1.00      1.00      1.00       956
          4       0.90      0.98      0.94       145
          5       1.00      0.87      0.93        15
          6       0.98      0.99      0.98       248
          7       1.00      0.97      0.99       139
          8       0.98      0.96      0.97       400
          9       0.99      1.00      1.00       322

avg / total       0.99      0.99      0.99      3587
```

Random Forest Results

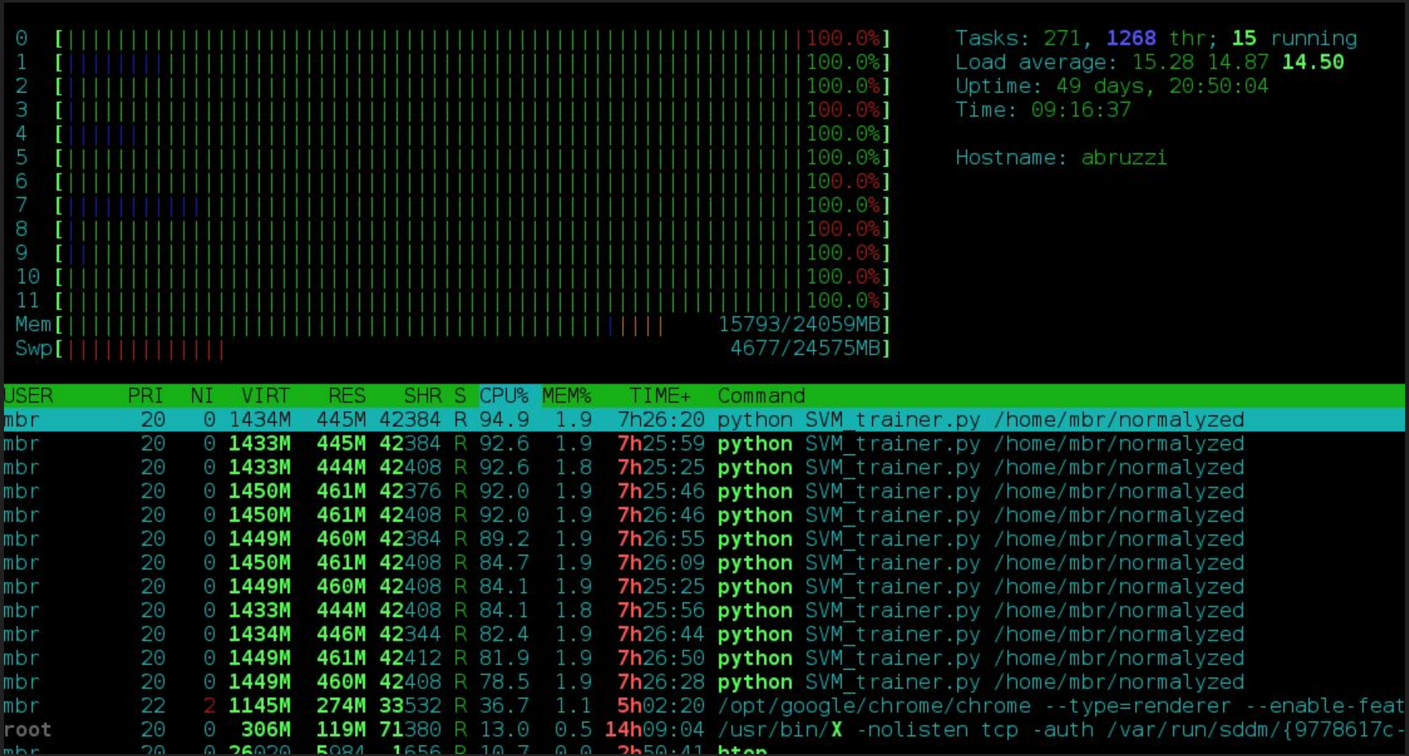
Best configuration found {'bootstrap': False, 'min_samples_leaf': 1, 'min_samples_split': 2, 'criterion': 'gini', 'max_features': 3, 'max_depth': None}

Test size 33% Train size 66%



```
             precision    recall  f1-score   support

          1       0.93      0.98      0.96       518
          2       0.99      0.99      0.99       844
          3       1.00      0.99      1.00       956
          4       0.90      0.98      0.94       145
          5       0.90      0.60      0.72        15
          6       1.00      0.99      0.99       248
          7       0.98      0.96      0.97       139
          8       0.98      0.92      0.95       400
          9       0.99      1.00      0.99       322

avg / total       0.98      0.98      0.98      3587
```

Neural Network Results



```
0  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Tasks: 271, 1268 thr; 15 running
1  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Load average: 15.28 14.87 14.50
2  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Uptime: 49 days, 20:50:04
3  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Time: 09:16:37
4  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
5  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]   Hostname: abruzzi
6  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
7  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
8  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
9  [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
10 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
11 [||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
Mem[|||||||||||||||||||||||||||||||||||||||||||||||        15793/24059MB]
Swp[||||||||||||                                            4677/24575MB]

USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
mbr        20   0 1434M  445M 42384 R 94.9  1.9  7h26:20 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1433M  445M 42384 R 92.6  1.9  7h25:59 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1433M  444M 42408 R 92.6  1.8  7h25:25 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1450M  461M 42376 R 92.0  1.9  7h25:46 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1450M  461M 42408 R 92.0  1.9  7h26:46 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1449M  460M 42384 R 89.2  1.9  7h26:55 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1450M  461M 42408 R 84.7  1.9  7h26:09 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1449M  460M 42408 R 84.1  1.9  7h25:25 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1433M  444M 42408 R 84.1  1.8  7h25:56 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1434M  446M 42344 R 82.4  1.9  7h26:44 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1449M  461M 42412 R 81.9  1.9  7h26:50 python SVM_trainer.py /home/mbr/normalyzed
mbr        20   0 1449M  460M 42408 R 78.5  1.9  7h26:28 python SVM_trainer.py /home/mbr/normalyzed
mbr        22   2 1145M  274M 33532 R 36.7  1.1  5h02:20 /opt/google/chrome/chrome --type=renderer --enable-feat
root       20   0  306M  119M 71380 R 13.0  0.5 14h09:04 /usr/bin/X -nolisten tcp -auth /var/run/sddm/{9778617c-
mbr        20   0 26020  5984  1656 R 10.7  0.0  2h50:41 htop
```

Related Works

- Learning and Classification of Malware Behaviour (DIMVA 2008)
  - ~10K samples, SVM, 14 Families, Dynamic features
- Lines of malicious code: Insights into the Malicious Software Industry (ACSAC 12)
  - 11 Families, Dynamic features, Dormant Functionalities
- Say no to overfitting (winner of Microsoft competition)
  - Semi-supervised learning, n-grams, bytes visualization

Conclusion

- So far the best algorithm is SVM for this kind of classification
    - If it will ever terminate we will see if NN are better in this setting

- The feature selection seems reflect real characteristics of the malware families

- Acknowledge to sklearn, my SSD disk and multiprocessing :)

The end...questions?