

Figure 2: Ensemble classification approach using veto Voting

ADD	AND	CALL	CMP	DEC
0.02%	0.0%	0.14%	0.73%	0.0%
0.04%	0.0%	0.28%	1.49%	0.0%
0.08%	0.05%	1.12%	2.52%	0.0%
0.1%	0.07%	1.25%	2.59%	0.0%
0.12%	0.07%	1.37%	2.66%	0.0%
0.14%	0.07%	1.51%	2.73%	0.0%
0.15%	0.07%	1.62%	2.79%	0.0%
0.19%	0.08%	1.99%	3.08%	0.0%
0.21%	0.08%	2.67%	4.27%	0.0%
0.23%	0.08%	2.81%	4.99%	0.0%
0.25%	0.09%	2.95%	5.06%	0.0%
0.5%	0.34%	3.31%	5.32%	0.0%
0.52%	0.34%	3.45%	5.39%	0.0%
0.54%	0.36%	3.58%	5.46%	0.0%

4.3 Principle Component Analysis

Opcode	Value
ADD	0.1964695559555954
AND	-0.2070616743183362
CALL	0.13921350465699048
CMP	-0.09148781143933675
DEC	-0.0333326144530759
INC	-0.08242422962726506
JA	-0.012591353992482419
JB	-0.009222056675904108
JBE	0.020716939021728796
JE	-0.004553424442303005
JGE	-0.005182430952544504
JL	0.1806058473300203
JLE	-0.3108804034690761
JMP	0.1260712964265208

4.4 Feature Labelling By using PCA method

Opcode	Value
ADD	0.1964695559555954
CALL	0.13921350465699048
JBE	0.020716939021728796
JL	0.1806058473300203
JMP	0.1260712964265208
LEA	0.12018573960694805
OR	0.3170181722279416
PUSH	0.1939875085505544
SAR	0.29943489477050794
SETB	0.00298518152178714

Opcode	Value
AND	-0.2070616743183362
CMP	-0.09148781143933675
DEC	-0.0333326144530759
INC	-0.08242422962726506
JA	-0.012591353992482419
JB	-0.009222056675904108
JE	-0.004553424442303005
JGE	-0.005182430952544504
JLE	-0.3108804034690761
JNZ	-0.119004860583489

4. Results

4.1 Dataset Creation using OllyDbg Disassembler

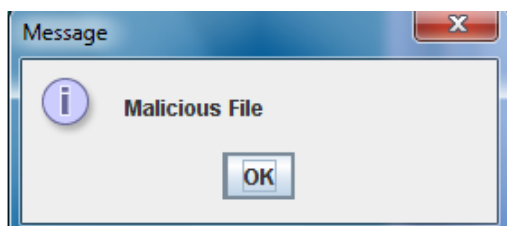
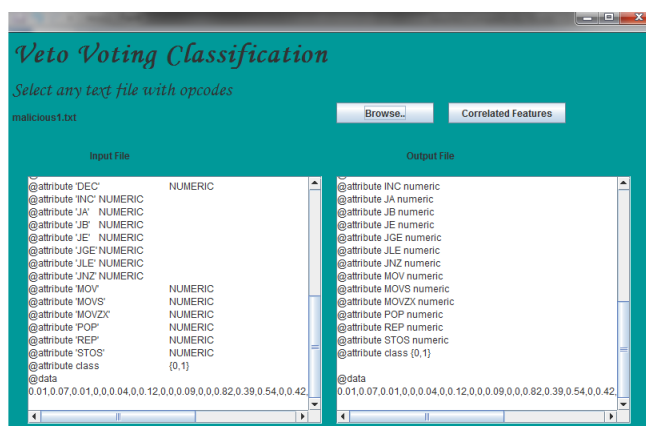
```

    rtrace1 - Notepad
    File Edit Format View Help
    Address Thread Command ; Registers and con
    Run trace closed

    New session
    Address Thread Command Registers and comments
    Flushing gathered information
    77639E8C 00000260 PUSH EBP
    77639E8D 00000260 MOV EBP,ESP EBP=00C2FAA0
    77639E8F 00000260 PUSH ECX
    77639E90 00000260 PUSH ECX
    77639E91 00000260 LEA EAX,DWORD PTR SS:[EBP-8] EAX=00C2FA98
    77639E94 00000260 PUSH EAX
    77639E95 00000260 CALL mtdll.rtlInitializeExceptionChain
    77639E9A 00000260 PUSH DWORD PTR SS:[EBP+C]
    77639E9D 00000260 PUSH DWORD PTR SS:[EBP+8]
    77639EA0 00000260 CALL mtdll.77639EAB EAX=75473388, EDX=00381
    <ModuleEntryPoint> 00000260 CALL Setup.00380DE6 Program entry p
    00381682 00000260 JMP Setup.00381540
    00381540 00000260 PUSH 58
    00381542 00000260 PUSH Setup.0045D630
    00381547 00000260 CALL Setup.003869F0 EAX=00C2FA2C, EBP=00C2F
    0038154C 00000260 LEA EAX,DWORD PTR SS:[EBP-68] EAX=00C2F9D4
    0038154F 00000260 PUSH EAX pStartupInfo = 00C2F9D4
    00381550 00000260 CALL DWORD PTR DS:[<&KERNEL32.GetStartupInfow]
    00381556 00000260 XOR ESI,ESI
    00381558 00000260 CMP DWORD PTR DS:[473878],ESI
    0038155E 00000260 JNZ SHORT Setup.00381568
    00381560 00000260 PUSH ESI
    00381561 00000260 PUSH ESI
    00381562 00000260 PUSH 1
    00381564 00000260 PUSH ESI
    00381565 00000260 CALL DWORD PTR DS:[<&KERNEL32.HeapSetInformati
    00381568 00000260 MOV EAX,5A4D EAX=00005A4D
    
```

4.2 Opcode Extraction with Occurences for each *.txt file in the dataset

4.5 Veto Voting Ensemble Classification



[4] D. Bilar, "Callgraph properties of executables and generative mechanisms, AI Commun., Special Issue on Network Anal. in Natural Sci.and Eng., vol. 20, no. 4, pp. 231-243, 2007.

[5] I. Santos, Y. K. Peña, J. Devesa, and P. G. Garcia, "N-grams-based file signatures for malware detection," S3Lab, Deusto Technological Found., 2009.

[6] R. Sekar, M. Bendre, D. Bollineni, and Bollineni, R. Needham and M. Abadi, Eds., "A fast automaton-based method for detecting anomalous program behaviors," in Proc. 2001 IEEE Symp. Security and Privacy, IEEE Comput. Soc., Los Alamitos, CA, USA, 2001, pp. 144-155."

[7] Wei-Jen Li, W. L. K. Wang, S. Stolfo, and B. Herzog, "Fileprints: Identifying file types by n-gram analysis, in Proc. 6th IEEE Inform. Assurance Workshop, June 2005, pp. : 64-71.

[8] I. Santos, F. Brezo, B. Sanz, C. Laorden, and Y. P. G. Bringas, "Using opcode sequences in single-class learning to detect unknown malware," IET Inform. Security, vol. 5, no. 4, pp. 220227, 2011.

Author Profile



Ms. Shital Kuber, Received the Bachelors degree (B. E.) computer Engineering in 2013 from VPCOE, Baramati. she is now pursuing M. E. degree, from VPCOE, Baramati.



Prof. Digambar Padulkar is working as Assistant professor, Department Of Computer Engineering, Vidya Pratishthan's College Of Engineering, Baramati, Pune, Maharashtra.. His research areas include Uncertain Data Mining, Applications of Data Mining in Business and Intelligent predictions.

5. Conclusions

1. The less frequent opcodes having an importance rating for classifying benign and malicious software. While mov has a negative impact on the classification and identification of software.
2. Opcode mov is a poor indicator of benign and malicious software. mov opcodes inhibits the ability to correctly classify software when used with other opcodes. Using the eigenvector prefilter, irrelevant features can safely remove.
3. Ensemble SVM classifier provides good accuracy to detect malware as compared to other methods.

Acknowledgement

I avail this opportunity to express my deep sense of gratitude and whole hearted thanks to my guide Prof. D. M. Padulkar for giving his valuable guidance, inspiration and encouragement to embark this paper. Without his Coordination, guidance and reviewing, this task could not be completed alone.

References

[1] P. O'Kane, S. Sezer, K. McLaughlin, and E. Im, "SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection", IEEE Transaction on information Forensic And Security, VOL. 8, NO. 3, March 2013.

[2] A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, "Detecting unknown malicious code by applying classification techniques on opcode patterns, Security Informatics, vol. 1, pp. 1-22, 2012.

[3] D. Bilar, "Opcodes as predictor for malware, Int. J. Electron. Security Digital Forensics, vol. 1, no. 2, pp. 156 - 168, 2007.