Dynamic Analysis of Ransomware using Opcodes and Opcode Categories

Domhnall Carlin, Philip O'Kane, Sakir Sezer

Centre for Secure Information Technologies (CSIT), Queen's University Belfast, N.Ireland.

ABSTRACT

The explosion of ransomware in recent years has served as a costly reminder that the malware threatscape has moved from that of socially-inept hobbyists to career criminals. This paper investigates the efficacy of dynamic opcode analysis in distinguishing cryptographic ransomware from benignware, and presents several novel contributions. Firstly, a new dataset of cryptoransomware dynamic run-traces, the largest of its kind in the literature. We release this to the wider research community to foster further research in the field. Our second novel contribution demonstrates that a short run- length of 32k opcodes can provide highly accurate detection of ransomware (99.56%) compared to benign software. Third, our model offers a distinct advantage over other models in the literature, in that it can detect a form of benign encryption (i.e. file zipping) with 100% accuracy against not only ransomware, but also the non-encrypting benignware in our dataset. The research presented here demonstrates that dynamic opcode tracing is capable of detecting ransomware in comparable times to static analysis, without being thwarted by obfuscation tactics.

Keywords: Malware, Invasive Software, Ransomware, Cryptoransomware, Machine Learning, Opcode, Dynamic Analysis.

1 INTRODUCTION

Beginning with Brain, the first PC virus, in 1986, the modern malware pandemic has escalated at an alarming pace. This deliberately disruptive software was formerly the preserve of hobbyists seeking infamy and bragging rights, but has moved to enable a worldwide criminal ecosystem, providing a rich platform for the extortion of individuals, businesses, the public sector and even sovereign states. McAfee (McAfee Labs Threats Report December 2017, 2017) state that 1.5 million new ransomware samples were found in Q3 of 2017, a rise of 36% on the previous quarter. This drove the total number of ransomware samples held by McAfee to breach 12 million.

A. The rise of ransomware

Ransomware is deliberately malicious software that denies or limits access to a machine or data, often with just enough access to facilitate payment of the ransom demand. The perfect storm of anonymous payment channels and internet access, available encryption, and increased attack surface has given criminals the opportunity to cash in on the recent data explosion. The incessant demand for connected devices and the subsequent exponential rise in data volumes has fast outpaced security considerations, leaving low-hanging targets for the exploitative (O’Kane, Sezer, & Carlin, 2018).

Early versions of ransomware were largely unsuccessful, as they faltered on one or more of the ‘perfect storm’ factors mentioned previously. The AIDS PC virus from 1989 was distributed via floppy disc, but used easily-breakable encryption and easily-traceable payment methods (a post office box). However, even as recently as 2010, ransomware was viewed by some authors as a non-event:

‘The ransomware phenomenon is a reality that has to be monitored but in some ways it is not a mature and complex enough activity that deserves such communication around it. Ransomwares[sic] as a mass extortion mean is certainly doomed to failure...may means[sic] that criminals have evolved to something else and other sources of income.’ (Gazet, 2010, P.90)

Latter-day ransomware has evolved dramatically, increasing in aggression, sophistication and potential with each strain and iteration. In (An ISTR Special Report: Ransomware and Businesses 2016, 2016) Symantec listed the average ransom demand as $294 in 2015, rising to $679 in 2016. This, coupled with the fluctuations in Bitcoin exchange rates, has helped to drive the rise in ransomware-as-a-service (RaaS). Here, customized binaries are available for as little as $50 and a 10% cut of the proceeds. With such a high potential return, and such ease of access to the offending ransomware, the onslaught may only be beginning. A shift from a shotgun approach to specific big-ticket targets has led to the ransom of the San Francisco metro system, private healthcare practices, the UK National Health Service, hotel key-cards, and specific Big Data stores. With the data explosion and rapid increase in connected devices, the prospect of escalating attacks is becoming all too real.

The remainder of this paper is presented as follows: Section Two provides highlights of the literature to date on dynamic analysis of malware and ransomware, and the objectives of the present work. Section Three briefly describes the methods used to generate the dataset for the present research and the experimental analyses. Section Four presents the results of our experiments and Section Five provides conclusions on the importance of the results.

BACKGROUND

A. Dynamic opcode analysis of malware

Research into overcoming issues inherent in signature detection and obfuscation has focused on observing behaviours of the malware at run-time. Opcodes (operation codes) are the readable machine language instructions that perform CPU operations. Monitoring the run-time opcodes provides a snapshot of the malware’s behaviour, while circumventing obfuscation strategies.

Santos et al. (2011) inspected n-grams (i.e. combinations of observed opcode sequences) in a static analysis. Accuracy and F-measure were reported as >85% in detecting malware from benign. Runwal et al. (2012) found graph approaches to detecting metamorphic malware using opcodes to be successful not only between metamorphic malware and benignware, but metamorphic malware and other malware types. O’Kane et al. (2013) found that 99.5% of variance in their data was attributed to the top 8 opcodes, providing feature reduction from the original 150 opcodes. Carlin et al. (2017) used run-time opcodes to classify 48,000 malware executables. The effects of run-length (i.e. trace length) and n-grams were also analysed. The authors found that traces of 32k opcode length using n=1 showed the highest levels of accuracy (99.05%) using a Random Forest classifier and 10-fold cross-validation. This indicates that dynamic opcode analysis, when applied to a broad range of malware, can speedily and accurately detect the malicious behaviours of unseen malware.

B. Dynamic analysis of ransomware

Dynamic analyses of ransomware have mainly concentrated on taint analysis and network monitoring. Ahmadian et al. (2015) attempted to detect ransomware prior to the malicious data encryption by detecting domain generation algorithm (DGA) and key exchange activities. Ransomware often employs a DGA to connect to a randomized domain, to avoid hard-coding Command and Control (C&C) domains into the malware. If the DGA generates the valid domain successfully, a connection is made to the C&C, and a key exchange takes place. Using a text analysis approach, the authors sought to check if a domain was probably being generated at random using typical rules for English. This appears to be conceptually flawed, as non-English domains, or even acronyms would potentially be flagged. A second parallel approach monitored traffic on out- ward network connections, allowing the user the opportunity to deny attempted connections. The authors claim 100% accuracy with 0 false negatives (FN) using their framework. However, the dataset only contained 20 samples of ransomware and the overall accuracy was actually 50% across these 20. While a FN rate of 0 may have been observed, the research did not examine benign files and so false positives could not be reported. Monika et al. (2016) examined ransomware on both Android and Windows. Using dynamic analysis, the system changes and network communications of each sample were observed. It was found that there was consistency in approach across strains of ransomware, including file writes, registry manipulation and network access.

Cabaj et al. (2015) investigated six CryptoWall samples dynamically, focusing on the network activity, which allowed the tracking of infected proxies and the infrastructure behind the infection. This work was furthered in Cabaj et al. (2016) with an expanded dataset (350 instances). In this work, multiple domain names resolved to the same IP addresses were found to be responsible for multiple C&C services.

Scaife et al. (2016) employed dynamic analysis on 492 ransomware samples from 14 families. Change-monitoring techniques were used to analyse file type changes, similarity measurement, entropy, file dele- tion, and file type funnelling. Ransomware was detected with a 100% TP rate, with the median file-loss due to encryption as low as 10 out of 5100 files. This model was then used with simulated user behaviour on 30 Windows applications. The only FP the model triggered was during benign packing and encryption by the 7Zip application.

Mbol et al. (2016) employed entropy analysis to detect encryption behaviour of TorrentLocker on JPEG files. With an abnormal behaviour defined as belonging to a process that opens a large number of files, and the structure of the output differing from the input structure. The authors found that the algorithm could differentiate between typical JPEGs and encrypted JPEGs using only the first 128kb with 99.95% accuracy. They postulated that, because sections of JPEGs resemble encrypted files closely, more highly-structured files (e.g. DOC or TXT) could be distinguished even more easily. However, using the abnormal behaviour definition to diagnose file-encryption by ransomware would mostly likely classify typical benign compression or encryption behaviours as malicious. Similarly, malware could bypass the first rule of that definition by starting child processes for each file encryption.

Previous dynamic approaches to ransomware analysis have largely focussed on the effects of the file on the host system, rather than directly observing the behaviour of the file. For example, monitoring the network traffic or file I/O behaviours on disc, rather than instructions the executable is issuing. As the latter technique has proven successful at distinguishing malware from benignware in large datasets and with high accuracy, it would seem logical that ransomware should also yield to such analyses. The research presented here investigates the application of these approaches to ransomware, using a novel dataset with respect to the following research questions:

1) Can opcode counts extracted from run-traces offer accurate ransomware detection over a large sample size?
2) Does a 32k opcode run-length offer superior accuracy over full-length traces?
3) Can logical feature reduction, based on domain knowledge, be used to reduce the feature set, and therefore model complexity, while maintaining accuracy?
4) Can the method(s) which successfully detect(s) ransomware behaviour, differentiate these from benign encryption (e.g. file zipping) behaviours?

2 Methodology

The methodology employed for the work presented here uses the framework created in Carlin et al. (2017), with slight modifications for use in the context of ransomware. For brevity, this process is briefly restated here. To acquire samples, the repository at VirusShare.com (Roberts, 2014) was selected for its size, modernity and facilities, with 40 million malware samples at the time of writing. For the present paper, the cryptoransom collection was used, containing approximately 36,000 files.

When PE files were extracted, 21,378 remained for the experiment. Benign files were harvested from Windows machines, and the SMOTE algorithm (Chawla et al., 2004) was implemented to oversample the minority class, resulting in 3,591 benign files. The benign zipping class was created using 7Zip to pack and encrypt a collection of files. This was oversampled to create a dataset of 1000 benign zipping samples.

A clean baseline snapshot of a Windows 10 VirtualBox VM was used to execute each file for nine minutes. This time was used to provide a pragmatic balance between number of samples executed and execution time. The open source debugger OllyDbg v2 was used to trace the dynamic opcodes of each file during runtime, with StrongOD v0.4.8.892 used to mask the presence of the debugger, as per (O'Kane et al., 2013). The host Windows OS was crafted to include the items found in a typical system, including a full recycling bin, browsing history, recent documents, Flash, .Net, Java etc. Further anti-virtualization mitigations were not utilized at the execution stage, as the presence of VBoxGuestAdditions is an easy indicator of the system being under a virtualised environment, but is necessary for automation. One main aim of dynamic investigations of malware is to experience the binary as a user would normally experience it, including virtualized platforms. Lastly, the anti-virtualization attacks by malware may also provide further useful features for detection when operating as low level as opcode instructions. This dataset was then input to a Random Forest classifier, as implemented in WEKA 3.9. Finally, a matching algorithm was employed to parse through the runtraces, examine the opcodes and match the opcode's parent opcode category, counting the occurrences of the latter. Fig.1 illustrates the translation of opcodes to their behavioural category.

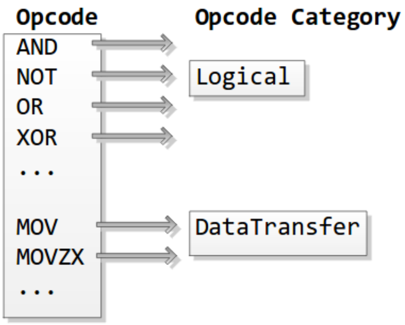


Fig. 1. Opcodes translating to their parent categories

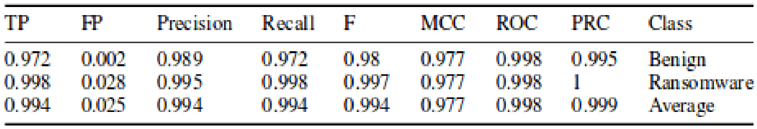
In this example, with standard opcode counting the counts for all opcodes depicted would be incremented by one. For the opcode category translation, AND NOT OR XOR are all seen as examples of Logical opcodes, so the Logical count would be incremented by 4. Similarly, DataTransfer would be incremented by two, rather than one apiece for MOV and MOVZX. This gives a form of logical feature reduction which is based upon domain knowledge, and discussed further later in this paper. As this dataset is the largest dynamic opcode analysis of ransomware, to the best of our knowledge, we wish to release this dataset to the wider research community in order to further the body of research into mitigating dangerous malware threats.

3 Results

A. Benignware vs ransomware

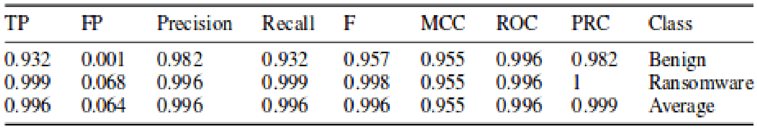
To confirm past work that showed a 32k length run-trace provided optimal accuracy with lower computational overhead, both 32k run-traces (32k) and full-length run-traces (FL) were compared. Table I shows the metrics for the full-length traces, and Table II lists the metrics for the 32k set. Overall the results show that the classifier can distinguish between each class with high levels of accuracy. The F1 measure was 0.994 (FL) and 0.996 (32k) averaged across the classes, indicating balance between both precision and recall. The shorter run-length did, however, introduce higher levels of FP ratings for ransomware, when compared to the full- length traces. Area-under Receiver Operating Characteristic curve (ROC) indicated excellent classification performance at 99.8% and 99.6% respectively.

TABLE I
MACHINE LEARNING METRICS FOR FULL-LENGTH BENIGN AND RANSOMWARE TRACES



TP	FP	Precision	Recall	F	MCC	ROC	PRC	Class
0.972	0.002	0.989	0.972	0.98	0.977	0.998	0.995	Benign
0.998	0.028	0.995	0.998	0.997	0.977	0.998	1	Ransomware
0.994	0.025	0.994	0.994	0.994	0.977	0.998	0.999	Average

TABLE II
MACHINE LEARNING METRICS FOR 32K RUN-LENGTH BENIGN AND RANSOMWARE TRACES



TP	FP	Precision	Recall	F	MCC	ROC	PRC	Class
0.932	0.001	0.982	0.932	0.957	0.955	0.996	0.982	Benign
0.999	0.068	0.996	0.999	0.998	0.955	0.996	1	Ransomware
0.996	0.064	0.996	0.996	0.996	0.955	0.996	0.999	Average

B. Benignware vs ransomware vs zipping

To investigate whether the RF model can distinguish ransomware from benign encryption (zipping), a three-class problem was put to the classifier. Again, the 32k run-length was compared to the full-length traces.

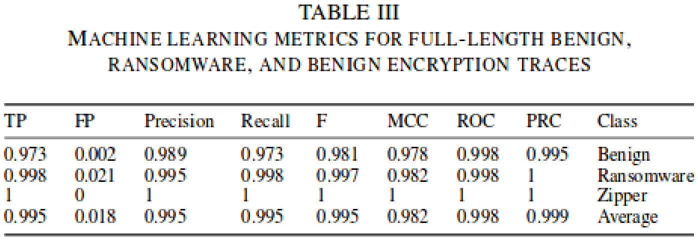


TABLE III
MACHINE LEARNING METRICS FOR FULL-LENGTH BENIGN, RANSOMWARE, AND BENIGN ENCRYPTION TRACES

TP	FP	Precision	Recall	F	MCC	ROC	PRC	Class
0.973	0.002	0.989	0.973	0.981	0.978	0.998	0.995	Benign
0.998	0.021	0.995	0.998	0.997	0.982	0.998	1	Ransomware
1	0	1	1	1	1	1	1	Zipper
0.995	0.018	0.995	0.995	0.995	0.982	0.998	0.999	Average

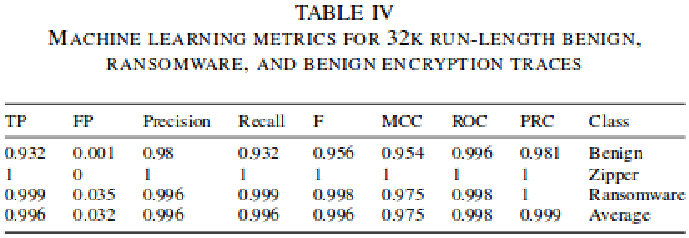


TABLE IV
MACHINE LEARNING METRICS FOR 32K RUN-LENGTH BENIGN, RANSOMWARE, AND BENIGN ENCRYPTION TRACES

TP	FP	Precision	Recall	F	MCC	ROC	PRC	Class
0.932	0.001	0.98	0.932	0.956	0.954	0.996	0.981	Benign
1	0	1	1	1	1	1	1	Zipper
0.999	0.035	0.996	0.999	0.998	0.975	0.998	1	Ransomware
0.996	0.032	0.996	0.996	0.996	0.975	0.998	0.999	Average

As with the two-class problem above, the classification metrics showed excellent performance in distinguishing the classes, with a TP rate of 99.6%. The highlight of these results is the 100% rating for each score within the Zipper class, i.e. 100% of the time, the classifier could distinguish not only benignware from ransomware, but zipping from both other classes.

C. Accuracies

The accuracies (stated as correctly classified instances out of all classifications) for each run-length (full vs 32k) and all classes are depicted in fig 2.

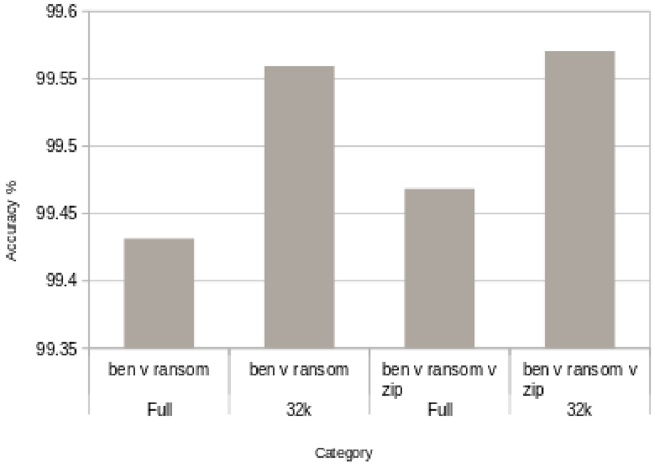


Fig. 2. Accuracies (correctly classified instances) for all classes and run-lengths.

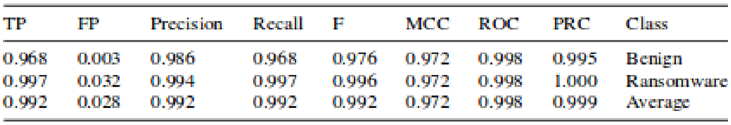
Investigating at the 32k run-length shows higher accuracy levels than using the full-length traces, for both the two-class and three-class analyses. As the zipping class is detected with 100% accuracy, this has improved the overall accuracy in the three-class analysis, with the other classes showing almost identical results (Benign +0.1%) between the two-class and three-class problems. Overall, the 32k run-length benign vs ransomware vs zipping analysis showed the highest level of accuracy.

D. Opcode categories

Table V shows the machine learning metrics for ransomware vs benignware, when the feature vector comprised parent opcode categories, rather than individual opcodes. This logical feature reduction provided 46 numerical features, prior to any attempts at statistical feature reduction.

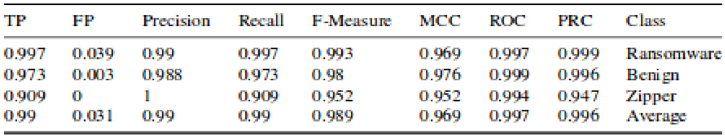
Overall accuracy (i.e. correctly classified instances) was 99.27%. TP detections for the benign category improved from 93.2% to 96.8% from the full opcode feature set (table II), though the ransomware category declined slightly from 99.9% to 99.7%, yielding an average of 99.2% compared to 99.6%. However, considering the ratio metrics (MCC, ROC and PRC), overall performance increased with the reduced feature set.

TABLE V
MACHINE LEARNING METRICS FOR 32K RUN-LENGTH TRACES OF RANSOMWARE AND BENIGN, USING OPCODE CATEGORY FEATURES



TP	FP	Precision	Recall	F	MCC	ROC	PRC	Class
0.968	0.003	0.986	0.968	0.976	0.972	0.998	0.995	Benign
0.997	0.032	0.994	0.997	0.996	0.972	0.998	1.000	Ransomware
0.992	0.028	0.992	0.992	0.992	0.972	0.998	0.999	Average

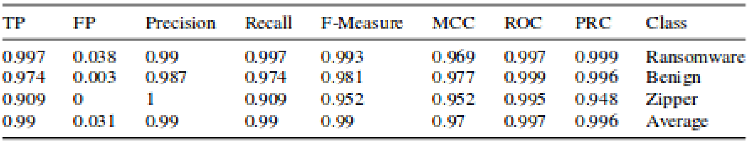
TABLE VI
MACHINE LEARNING METRICS FOR 32K RUN-LENGTH TRACES OF RANSOMWARE, BENIGNWARE AND ZIPPING, USING OPCODE CATEGORY FEATURES



TP	FP	Precision	Recall	F-Measure	MCC	ROC	PRC	Class
0.997	0.039	0.99	0.997	0.993	0.969	0.997	0.999	Ransomware
0.973	0.003	0.988	0.973	0.98	0.976	0.999	0.996	Benign
0.909	0	1	0.909	0.952	0.952	0.994	0.947	Zipper
0.99	0.031	0.99	0.99	0.989	0.969	0.997	0.996	Average

Table VI shows the results when a three-class problem is considered, similar to Table III, but with the logically-reduced feature set. Overall accuracy was 98.96%, compared with 99.57% for the full opcode feature set. While this is obviously lower, a lower accuracy can be expected with increased class numbers, due to decision boundaries. This may be particularly notable in this experiment, due to the inter-class imbalance being increased with the additional class. Overall F1, ROC and PRC scores were excellent, indicating a strong predictive model, which was within 0.1% of the scores of the two-class comparative experiment.

TABLE VII
MACHINE LEARNING METRICS FOR 32K RUN-LENGTH TRACES OF RANSOMWARE, BENIGNWARE AND ZIPPING, USING OPCODE CATEGORY FEATURES REDUCED BY GAIN RATIO (N=20)



TP	FP	Precision	Recall	F-Measure	MCC	ROC	PRC	Class
0.997	0.038	0.99	0.997	0.993	0.969	0.997	0.999	Ransomware
0.974	0.003	0.987	0.974	0.981	0.977	0.999	0.996	Benign
0.909	0	1	0.909	0.952	0.952	0.995	0.948	Zipper
0.99	0.031	0.99	0.99	0.99	0.97	0.997	0.996	Average

To investigate whether these metrics could be improved by using traditional feature selection on top of logical feature reduction, Gain Ratio was employed using a Ranker search algorithm to find the best features using 10-fold cross-validation. This yielded 20 features with a merit score >0, and so the feature vector was reduced to only include these 20 opcode categories. A separate Random Forest model was trained and tested using 10-fold cross-validation, as in all other experiments. The results are presented in table VII. Overall accuracy increased slightly from 98.96% to 98.97% using the top 20 features. This slightly improved the performance of the two minority classes, while maintaining the accuracies of the ransomware class.

4 FUTURE RESEARCH DIRECTIONS

Further investigation is required into the effects of class imbalance within our dataset, and potential avenues for mitigation. Future work could refactor this as a two-class problem, incorporating the benign and zipping classes as one. Similarly, the ransomware class could be investigated with respect to other non-cryptographic malware.

5 CONCLUSION

This paper makes several novel contributions to the current body of knowledge. First, we present a dynamic opcode analysis of ransomware on the largest dataset of its kind in the literature, to the best of our knowledge. We wish to release this dataset to the wider research community to help propagate future work in the field.

Second, we confirm findings of our previous work that 32k run-length is most accurate. This is important, particularly in the context of ransomware, as speedy detection and mitigation of the threat is key in preventing data loss. Further, this serves as confirmation that our methodological decision to limit runtime to nine minutes is more than adequate to capture the vast majority of information required to enable a classifier to correctly predict a class.

Third, the three-class problem showed the most accurate results, due to the 100% accuracy of the zipping class. It is widely accepted in machine learning that increasing numbers of classes typically increases misclassification. Despite this, the comparison of ransomware with benign and benign encryption showed the highest accuracy. A key advantage of this model over past work is the ability of our model to detect ransomware, benignware and benign encryption behaviours, with 100% accuracy in the latter case.

Finally, we explored the potential of logical feature reduction using domain knowledge boosted by statistical feature reduction, which has not previously been applied in this manner. This indicates that opcode categories, while slightly less accurate than opcodes, can provide a highly accurate level of ransomware detection, with reduced computational effort from a reduced feature set.

This research confirms our past work into the advantages of dynamic opcode analysis in the context of malware detection. Within the realm of ransomware, fast detection is essential, and our model demonstrates that short run-times are capable of offering high detection rates, while mitigating obfuscation techniques.

6 REFERENCES

Ahmadian, M. M., Shahriari, H. R., & Ghaffarian, S. M. (2015, 9). Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. *Information Security and Cryptology (ISCISC), 2015 12th International Iranian Society of Cryptology Conference on* (pp. 79-84). IEEE. doi:10.1109/ISCISC.2015.7387902

Cabaj, K., Gawkowski, P., Grochowski, K., & Kosik, A. (2016, 10). Developing malware evaluation infrastructure. *8*, pp. 981-989. IEEE. doi:10.15439/2016F490

Cabaj, K., Gawkowski, P., Grochowski, K., & Osojca, D. (2015). Network activity analysis of CryptoWall ransomware. *Przegląd Elektrotechniczny, 91*, 201-204.

Carlin, D., O'Kane, P., & Sezer, S. (2017). Dynamic Analysis of Malware Using Run-Time Opcodes. In *Data Analytics and Decision Support for Cybersecurity* (pp. 99-125). Springer International Publishing. doi:10.1007/978-3-319-59439-2_4

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004, 6). Editorial, special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter, 6*, 1. doi:10.1145/1007730.1007733

Gazet, A. (2010, 2). Comparative analysis of various ransomware virii. *Journal in Computer Virology, 6*, 77-90. doi:10.1007/s11416-008-0092-2

ISTR. (2016). *An ISTR Special Report: Ransomware and Businesses 2016.* Tech.rep., Symantec. Retrieved from www.symantec.com/content/en/us/.../ISTR2016_Ransomware_and_Businesses.pdf

Mbol, F., Robert, J.-M., & Sadighian, A. (2016). An Efficient Approach to Detect TorrentLocker Ransomware in Computer Systems. In *Cryptology and Network Security* (Vol. 10052, pp. 532-541). Springer International Publishing. doi:10.1007/978-3-319-48965-0_32

McAfee Labs (2017). *McAfee Labs Threats Report December 2017.* Tech. rep.,McAfee. Retrieved from https://www.mcafee.com/us/resources/ reports/rp-quarterly-threats-dec-2017.pdf

Monika, Zavarsky, P., & Lindskog, D. (2016). Experimental Analysis of Ransomware on Windows and Android Platforms: Evolution and Characterization. *Procedia Computer Science, 94*, 465-472. doi:10.1016/j.procs.2016.08.072

O'Kane, P., Sezer, S., & Carlin, D. (2018). The Evolution of Ransomware. *IET Networks*.

O'Kane, P., Sezer, S., McLaughlin, K., & Im, E. G. (2013). SVM training phase reduction using dataset feature filtering for malware detection. *IEEE transactions on information forensics and security, 8*, 500-509.

Roberts, J.-M. (2014). *VirusShare.com*. Retrieved from http://www.virusshare.com

Runwal, N., Low, R. M., & Stamp, M. (2012, 5). Opcode graph similarity and metamorphic detection. *Journal in Computer Virology, 8*, 37-52. doi:10.1007/s11416-012-0160-5

Santos, I., Brezo, F., Sanz, B., Laorden, C., & Bringas, P. G. (2011). Using opcode sequences in single-class learning to detect unknown malware. *IET Information Security, 5*, 220. doi:10.1049/iet-ifs.2010.0180

Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016, 6). CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. (pp. 303-312). IEEE. doi:10.1109/ICDCS.2016.46

KEY TERMS

Malware: Deliberately malicious software, often colloquially called a ‘computer virus’.
Invasive Software: Software that intentionally infects a machine with malicious purposes.
Ransomware: Malware that intends to deny the user access to the device or data that it is attacking.
Cryptoransomware: Ransomware that encrypts users’ data, rendering it potentially irretrievable until a ransom demand is met. This is in contrast to Lockers, where access to the machine is denied, but the data remains intact.
Machine Learning: A branch of Artificial Intelligence, where statistical models are built upon existing knowledge in order to infer from unseen information.
Opcode: The human-readable mnemonic of machine code.
Dynamic Analysis: Any analysis technique whereby the code under investigation is actually executed.

RandomForest: an ensemble learner, which combines the decisions of multiple smaller learners (decision trees). RF differs from traditional tree-based learning algorithms, in that a random number of features are used at each node in the tree to decide the parameter, which improves noise immunity and reduces the tendency to overfit.

K-fold cross-validation: a model validation technique, whereby the data is separated into k-folds, typically 10. For the first iteration, the model is trained on folds 1-9, and tested on the tenth. The results are added to the result set, and the model is then discarded. A fresh model is then trained in iteration two, using folds 1-8 and 10, which is then tested on fold 9. The process is continued until k-iterations have been completed. The average of the result sets is then calculated, giving a performance estimate for the averaged models, as shown in 3.7. This can be viewed as 10 stratified iterations of a 90/10% holdout validation, but with the benefit of ensuring every data point is used to train and test the model.

F/F1 measure/score: the harmonic mean of precision and sensitivity (1 is the ideal). It is used as a more complex measure of a classifier's accuracy, beyond simply correctly classified instances. It is worth noting that the F-measure does not take into account the true negative classifications, which may be viewed as a disadvantage when compared to MCC.

Receiver Operating Characteristic: when the True Positive% rate is plotted against the False Positive % rate. 1 is ideal.

Matthews correlation coefficient (MCC): A correlation coefficient between the predicted and observed classifications. MCC is used as a measurement of the quality of a binary classifier, and is particularly useful with imbalanced classes, as it accounts for all false/positive measures.

Area under PRC: Area Under Precision Recall curve (1 is the ideal) plotted with precision against recall.

Precision/Positive Predictive Value: Number of true positive classifications out of predicted positive class.

True positive rate/Recall/Sensitivity: Number of instances correctly classified as positive out of all that were actually positive.

Number false positives/False positive rate: Number of instances falsely classified as positive when they were truly negative, out of all that were truly negative (i.e. Type I error).

BIOGRAPHICAL NOTES

Dr. Domhnall Carlin is a Research Fellow at the Centre for Secure Information Technologies at Queen's University Belfast. His research interests include machine learning for the detection of malware and fileless malware. Domhnall holds an MSc in Software Development and PhD in Computer Science from Queen's University, Belfast.

Dr. Philip O'Kane received his B.Eng in Electronic Systems and MSc in Informatics from the University of Ulster in 1994 and 2009 respectively. He was awarded a PhD from Queens University Belfast for his work on the detection of obfuscated malware. His career started as an embedded software engineer in the Telecoms industry and later moved to application development for the Finance industry. He has 20 years of software development in industry and is presently working as a Lecturer at the Centre for Secure Information Technologies, Queen's University Belfast. His interests include the low-level analysis and detection of malware.

Prof. Sakir Sezer leads Network Security Systems research at CSIT. His research interests include cloud, SDN/VNF and mobile security, traffic forensics, malware analysis and high-performance networks and content processing. He has successfully managed numerous EPSRC and EU research projects comprising consortia of academic and industrial partners. His patented research has led to major advances in the field of high-performance content and security processing and is being commercialised by Titan IC Systems. He has published 146 research papers, has given many plenary and keynote talks at major international (e.g. IEEE, IET) conferences.

REFERENCE

Reference to this paper should be made as follows: Carlin, D., O’Kane, P., and Sezer, S. (2018). Dynamic Analysis of Ransomware Using Opcodes and Opcode Categories. *International Journal on Cyber Situational Awareness*, Vol. 3, No. 1, pp. 84-97.