Title

SPAM: Signal Processing to Analyze Malware [Applications Corner]

Journal

IEEE Signal Processing Magazine, 33(2)

ISSN

1053-5888

Authors

Nataraj, Lakshmanan

Manjunath, BS

Publication Date

2016

DOI

10.1109/msp.2015.2507185

Peer reviewed

Lakshmanan Nataraj
and B.S. Manjunath

SPAM: Signal Processing to Analyze Malware

Cyberattacks have risen in recent times. The attack on Sony Pictures by hackers, allegedly from North Korea, received worldwide attention. U.S. President Barack Obama issued a statement and "vowed a U.S. response after North Korea's alleged cyberattack" [1]. This dangerous malware, termed *wiper,* could overwrite data and stop important execution processes. An analysis by the U.S. Federal Bureau of Investigation showed distinct similarities between this attack and the code used to attack South Korea in 2013, thus confirming that hackers reuse code from already existing malware to create new variants. This attack, along with other recently discovered attacks such as Regin and Opcleaver, give one clear message: current cybersecurity defense mechanisms are not sufficient enough to thwart these sophisticated attacks.

Today's defense mechanisms, such as commercial antivirus (AV) software, is based on scanning systems for suspicious or malicious activity. If such an activity is found, the files under suspect are either quarantined or the vulnerable system is patched with an update. In turn, the AV software is also updated with new signatures to identify such activities in the future. The scanning methods are based on a variety of techniques such as static analysis-, dynamic analysis-, and other heuristics-based techniques, which are often slow to react to new attacks and threats.

Static analysis is based on analyzing an executable without executing it. These techniques include searching for specific strings, computing cryptographic hashes, and disassembling the executable to extract features. On the other hand, dynamic analysis executes the binary executable and studies its behavioral characteristics in a virtual sandboxed environment. Some of the methods include system-call-level monitoring and memory snapshot comparison. Hackers are familiar with these standard methods and come up with ways to evade the current defense mechanisms. They produce new malware variants that easily evade the detection methods. These variants are created from existing malware using inexpensive, easily available "factory tool kits" in a virtual factory-like setting, which then spread and infect more systems. Once a system is compromised, it either quickly loses control and/or the infection spreads to other networked systems.

While security techniques constantly evolve to keep up with new attacks, hackers too change their ways and continue to evade defense mechanisms. As this never-ending billion dollar "cat and mouse game" continues, it may be useful to look at avenues that can bring in novel alternative and/or orthogonal defense approaches to counter the ongoing threats. The hope is to catch these new attacks using complementary methods that may not be well known to hackers, thus making it more difficult and/or too expensive for them to evade all detection schemes. This article focuses on such orthogonal approaches from signal and image processing that complement standard approaches.

Digital Object Identifier 10.1109/MSP.2015.2507185
Date of publication: 7 March 2016

Malware landscape

Malware—malicious software—is any software that is designed to cause damage to a computer, server, network, mobile phones, and other devices. Based on their specific function such as stealing data, spying, keylogging or others, malware are classified into different types such as trojans, backdoors, virus, worm, spyware, adware, and more. Malware are also identified by which platform they belong to, such as Windows, Linux, AndroidOS, and others. While most malware are geared toward the Windows platform, they are also quickly expanding to other platforms such as AndroidOS, Linux, and MAC OS X. Malware are further classified into families, which in turn, have many variants that perform almost the same function (Figure 1). According to the Computer Antivirus Research Organization (CARO) convention for naming malware, a malware is represented by Type:Platform/Family.Variant. For example, PWS:Win32/Zbot.gen denotes a password-stealer malware of the generic Zbot family that attacks 32-bit Windows platforms.

Malware variants are created either by making changes to the malware code or by using executable packers. In the former case, a simple mutation occurs by changing small parts of the code. These are referred to as *unpacked malware variants*. In the latter case, a more complex mutation occurs either by compressing or encrypting (usually with different keys) the main body of the code and appending a decompression/decryption routine, which during

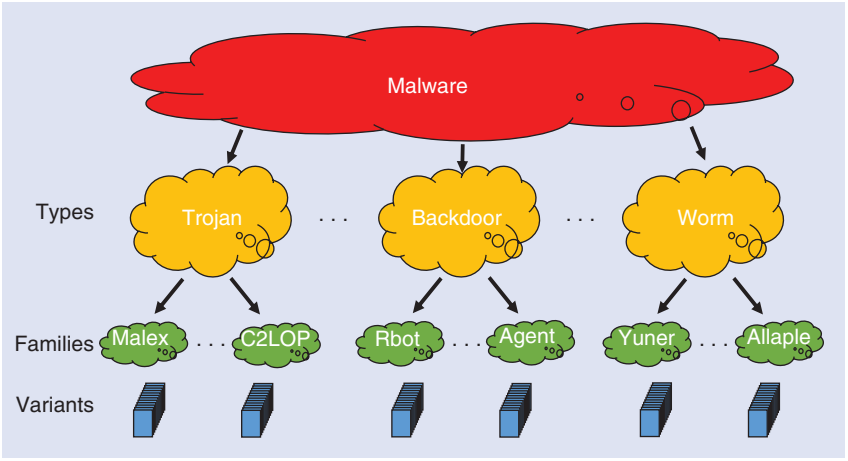

FIGURE 1. The malware landscape: malware are categorized by their type, families, and variants.

runtime decompresses/decrypts the encrypted payload. The new variants are called *packed malware variants,* and they perform the same function as the original malware but their attributes would be so different that AV software, which use traditional signature-based detection, would not be able to detect them. The tools used for obfuscation are called *executable packers*, available both as freeware and commercial tools. There are hundreds of packers that exist today that make it very easy for malware writers to create new variants.

Malware analysis

Malware classification deals with identifying the family of an unknown malware variant from a malware data set that is divided into many families. The level of risk of a particular malware is determined by its function, which is in turn reflected in its family. Hence, identifying the family of an unknown malware is crucial in understanding and stopping new malware. It is usually assumed that an unknown malware variant belongs to a known set of malware families (supervised classification). Having a high classification accuracy (the number of correctly classified families) is desirable. A closely related problem is malware retrieval, where the objective is to retrieve similar malware matches for a given query from a large database of malware. In malware detection, the objective is to determine if an unknown executable is malicious, benign, or unknown. This problem is more challenging than malware classification, where all samples are known to be malicious. In the following, we will focus on malware classification and malware retrieval.

A common way to defeat static analysis is by using packers on an executable, which compress and/or encrypt the executable code and create a new packed executable that mimics the previous executable in function but reveals the actual code only upon execution runtime. Dynamic analysis is agnostic to packing but is slow and time consuming. Furthermore, today's malware are designed to be virtual machine (VM) aware, which either do not do any malicious activity in the presence of VM or attempts a "suicide" when a VM is detected.

In this context, the challenge is to develop complementary methods that are able to quickly identify malware without the need for disassembly, unpacking, or execution. Alternative representations of malware data as one-dimensional (1-D) or two-dimensional (2-D) signals have patterns that are not captured by standard methods.

Malware images

A common method of viewing and editing malware binaries is by using Hex Editors, which display the bytes of the binaries in hexadecimal representation from "00" to "FF." Effectively, these are 8-bit numbers in the range of 0–255. Grouping these 8-bit numbers results in a 8-bit vector, from which we construct a signal or an image as shown in Figure 2. For an image, the width is fixed and the height is allowed to vary depending on the file size. To ensure that a small file does not appear horizontally stretched and a large file does not look vertically elongated, we provide some recommended image widths for different file sizes based on empirical observations [6]. Figure 3 shows an example image of a common Windows Trojan downloader, Dontovo.A, which downloads and executes arbitrary files. We can see that different sections of this malware exhibit distinctive image patterns. The .text section, which contains the executable code, has a fine-grained texture. It is followed by a black block (zeros), indicating zero padding at the end of this section. The .data section contains both uninitialized code (black patch) and initialized data (fine-grained texture). The final .rsrc section contains all the resources of the module, including the icon of the executable.

Visualizing these malware variants as images, one could make an empirical observation that there is visual similarity among malware variants of the same family (Figure 4). At the same time, the variants are also distinct from those belonging to other families. This is because the variants are created using either simple code mutations or packing. It is easy to identify the variants for unpacked malware since the structure of the variants are very similar. In the case of packed malware, the executable code is compressed and/or encrypted. During runtime, this code is then unpacked and executed. When two unpacked variants belonging to a specific malware family are using a packer to obtain packed variants of the same family, their structure no longer remains the same as that of the unpacked variants. However, the structure within the packed variants are still similar though the actual bytes may vary due to compression and/or encryption. This is because most of the current packers use weak encryption schemes [2]. The visual similarity of malware images motivated us to look at malware classification using techniques from computer vision, where image-based classification has been well studied. We use global image similarity descriptors and obtain compact signatures for these

malware, which are then used to identify their families.

Classification

Once the malware binary is converted to an image, an image similarity descriptor is computed on the image to characterize the malware. The descriptor that we use is the GIST feature [3], which is commonly used in scene classification [3], object recognition [4] and large-scale image search [5]. Every image location is represented by the output of filters tuned to different orientations and scales. A steerable pyramid with four scales and eight orientations is used. The local representation of an image is then given by $V^L(x) = V_k(x)_{k=1..N}$, where $N = 20$ is the number of subbands. To capture the global image properties while retaining some local information, the mean value of the magnitude of the local features is computed and averaged over large spatial regions: $m(x) = \sum_{x'} |V(x')| W(x' - x)$, where $W(x)$ is the averaging window. The resulting representation is downsampled to have a spatial resolution of $M \times M$ pixels (here we use $M = 4$). Thus, the feature vector obtained is of size $M \times M \times N = 320$. For faster processing, the images are usually resized to a smaller size (we use 64×64). Our experiments showed that our initial choice of image width and the width of the resized image does not significantly affect our performance.

To identify malware families, we use the nearest neighbor (NN) classifier, which assigns the family of the nearest malware to an unknown malware. We obtained four data sets: Malimg data set (Windows) [8], Malheur data set (Windows) [9], MalGenome data set (Android) [7], and VxShare ELF data set (Linux) [10]. On all four data sets we performed supervised classification with tenfold cross validation and obtained a high-classification accuracy (Table 1). Furthermore the accuracy of this method (95.14%) is comparable to that of the state-of-the-art dynamic analysis (98.12%), but 4,000 times faster [11]. In [12], we extend our approach to separate malware from benign software. To get a richer discrimination between benign and malicious samples, we adopt a section-aware approach and compute GIST descriptors on the entire binary as well as the top two sections of the binary thatcould contain the code. With more than 99% precision, this approach outperformed other static similarity features.

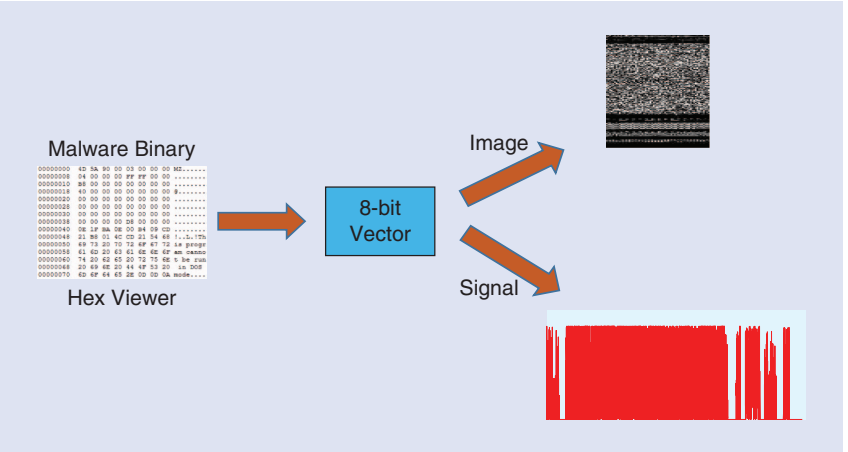

FIGURE 2. Malware can be represented as a 1-D signal or as a 2-D image.

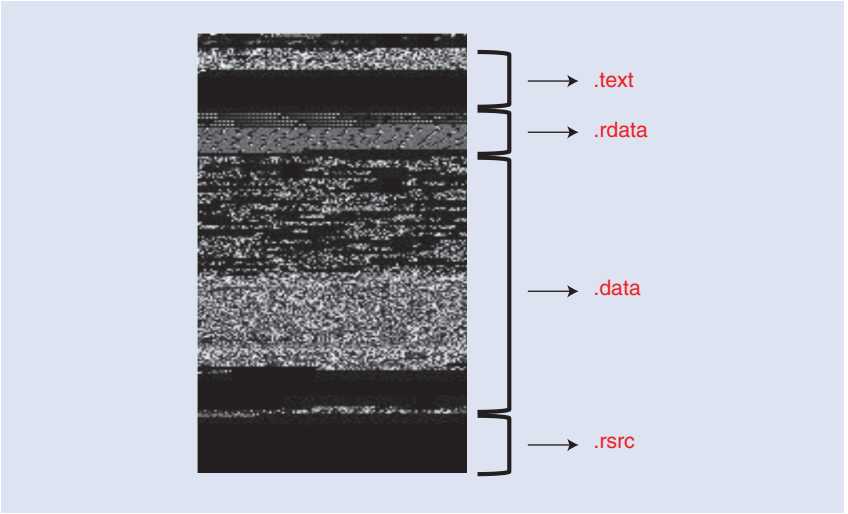

FIGURE 3. Visualizing a malware as a digital image: different sections of the executable are visible in the image.

Search and retrieval

We developed search and retrieval of malware (SARVAM) [13] (http://sarvam.ece.ucsb.edu), an online system for large-scale malware search and retrieval (Figure 5). It is one of the few systems available to the public where researchers can upload or search for a sample and retrieve similar malware matches from a large database. As in [6], we use GIST descriptors for content-based search and retrieval of malware. For fast search and retrieval, we use a scalable Balltree-based NN searching technique.

During the initial training phase of building SARVAM, we obtained a large corpus of malware samples from various sources. The image fingerprints for all the samples in the corpus are then computed and stored in a database. Simultaneously, we obtained the AV labels for all the samples from Virustotal [14], an online system that maintains a database of AV labels. These labels act as a ground truth and are later used to describe the nature of a sample, i.e., how malicious or benign a sample is. During the query phase, the fingerprint for the new sample is computed and matched with the existing fingerprints in the database to retrieve the top matches.

The initial database consisted of more than seven million samples comprising mostly malware and a few benign samples. For a new query, SARVAM finds a match in about six seconds. SARVAM

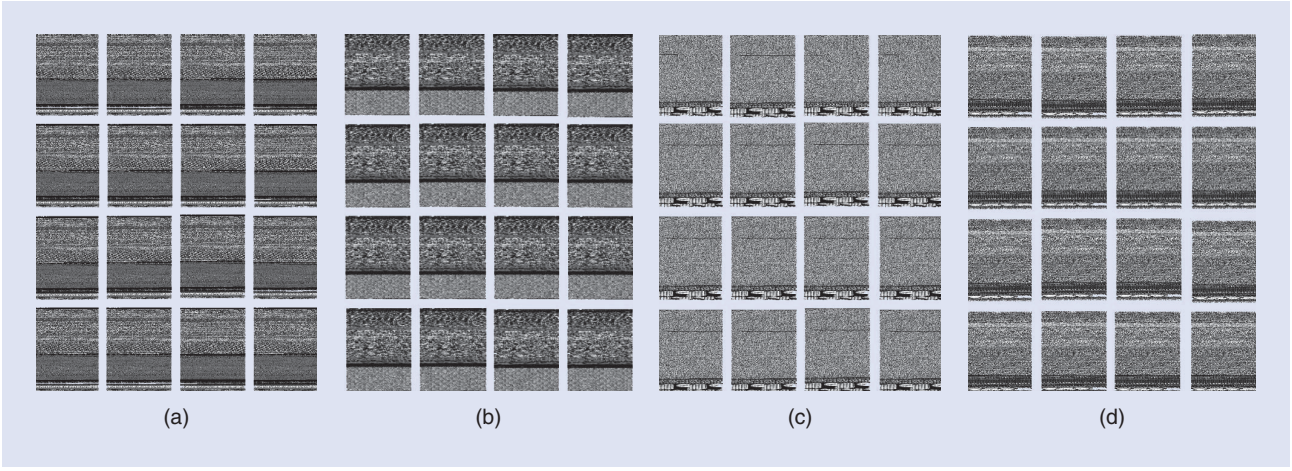

FIGURE 4. Visual similarity among malware variants of four different families. (a) Adialer.C, (b) Dialplatform.B, (c) Fakerean, and (d) Yuner.A.

Table 1. Accuracy on malware data sets from different operating systems: Windows, Linux, and Android.

Data Set	Size	Number of Families	Accuracy
Malimg (Win)	9,339	25	97.4
Malheur (Win)	3,131	24	98.37
VxShare (Linux)	568	8	83.27
Malgenome (Android)	1,094	13	84.55

has been operational since May 2012, and we have received more than 440,000 samples since then. Nearly 60% of the samples we received are variants of already existing malware from our database.

Sparsity-based malware analysis

In this section, we explore sparse representation-based classification (SRC) methods to classify malware variants into families. Such methods have been previously applied to problems where samples belonging to a class have small variations in them, such as face recognition [16] and iris recognition [18]. We developed sparsity inspired classification of malware variants (SATTVA) [15], where we model a malware variant belonging to a particular malware family as a linear combination of variants from that family. Since variants of a family have small changes in the overall structure and differ from variants of other families, projections of malware in lower dimensions preserve this "similarity."

Given a data set of N labeled malware belonging to L different malware families with P malware per family, the task is to identify the family of an unknown malware $\mathbf{u}$. We represent a malware as a digital signal $\mathbf{x}$ of range $[0, 255]$, where every entry of $\mathbf{x}$ is a byte value of the malware. Since each malware sample can have a different code-length, we normalize all vectors to a maximum length (M) by zero-padding.

The entire data set can now be represented as an $M \times N$ matrix $\mathbf{A}$, where every column represents a malware. Further, for every family k $(k = 1, 2, ..., L)$, we define an $M \times P$ matrix $\mathbf{A}_k = [\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots \mathbf{x}_{kP}]$ where $\mathbf{x}_{k\{.\}}$ represents a malware sample belonging to family k. Now, $\mathbf{A}$, can be expressed as a concatenation of block-matrices $\mathbf{A}_k$

$$\mathbf{A} = [\mathbf{A}_1 \mathbf{A}_2 .. \mathbf{A}_L] \in \mathbb{R}^{M \times N} \qquad (1)$$

Let $\mathbf{u} \in \mathbb{R}^M$ be an unknown malware whose family is to be determined, with the assumption that $\mathbf{u}$ belongs to one of the families in the data set. Then, following [16], we represent $\mathbf{u}$ as a sparse linear combination of the training samples as

$$\mathbf{u} = \sum_{i=1}^{L} \sum_{j=1}^{P} \alpha_{ij} \mathrm{x}_{ij} = \mathrm{A}\alpha, \qquad (2)$$

where $\alpha = [\alpha_{1,1}, \ldots, \alpha_{L,P}]^T$ represents the $N \times 1$ sparse coefficient vector $(N = LP)$. α will have nonzero values only for samples that are from the same family as $\mathbf{u}$. The sparsest solution to (2) can be obtained using basis pursuit [18] by solving an l_1-norm minimization problem. Estimating the family of $\mathbf{u}$ is done by computing residuals for every family in the training set and then selecting the family that has minimum residue.

Random projections

When a malware binary is represented as a numerical vector by considering every byte, the dimensions of that vector can be very high. For example, a 1-megabyte malware has around 1 million bytes and this could make the calculations computationally expensive. Hence, we project the vectors to lower dimensions using random projections (RPs). This also removes dependency on any particular feature extraction method. Previous works have demonstrated that SRC is effective in lower-dimensional RPs as well; see [16]–[18]. Let $\mathbf{R} \in \mathbb{R}^{D \times M}$ be the matrix that projects $\mathbf{u}$ from signal space M to $\mathbf{w}$ of a lower-dimensional space D $(D << M)$

$$\mathbf{w} = \mathbf{R}\mathbf{u} = \mathbf{R}\mathbf{A}\alpha. \qquad (3)$$

The entries of $\mathbf{R}$ are drawn from a zero-mean normal distribution. The above system of equations is underdetermined and sparse solutions can be obtained by reduced l_1-norm

minimization. The overall approach is shown in Figure 6.

We test our technique on two public malware data sets: the Malimg data set [8] and the Malheur data set [9]. On both data sets, we select equal number of samples to reduce any bias toward a particular family. For comparison, we use GIST descriptors, which we had previously applied for malware classification. We use the SRC framework to identify the malware family of a test sample and compare it with NN classification that we previously used in [6]. We vary the projected dimensions from 48 to 512, which are consistent for both RP and GIST. In our experiments, we choose 80% of a data set for training and 20% for testing. On both the Malimg data set [Figure 7(a)] and the Malheur data set [Figure 7(b)], the best accuracy is obtained for the combination of RPs and the SRC classification framework (92.83% for Malimg and 98.55% for Malheur). The projected dimension is 512 from higher dimensions of 840,960 (Malimg) and 3,364,864 (Malheur).

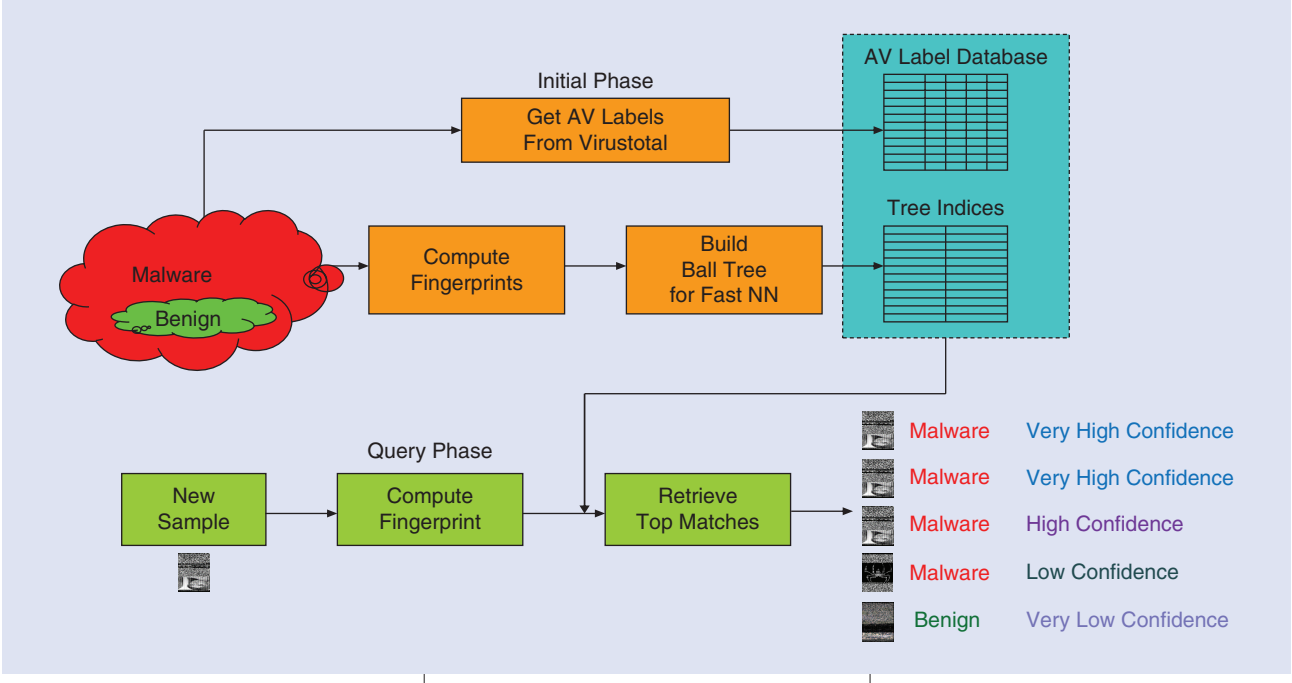

FIGURE 5. The block schematic of SARVAM: In the initial phase, the image similarity descriptors and AV labels are computed and stored in a database. In the query phase, the NNs along with their corresponding AV labels are retrieved.

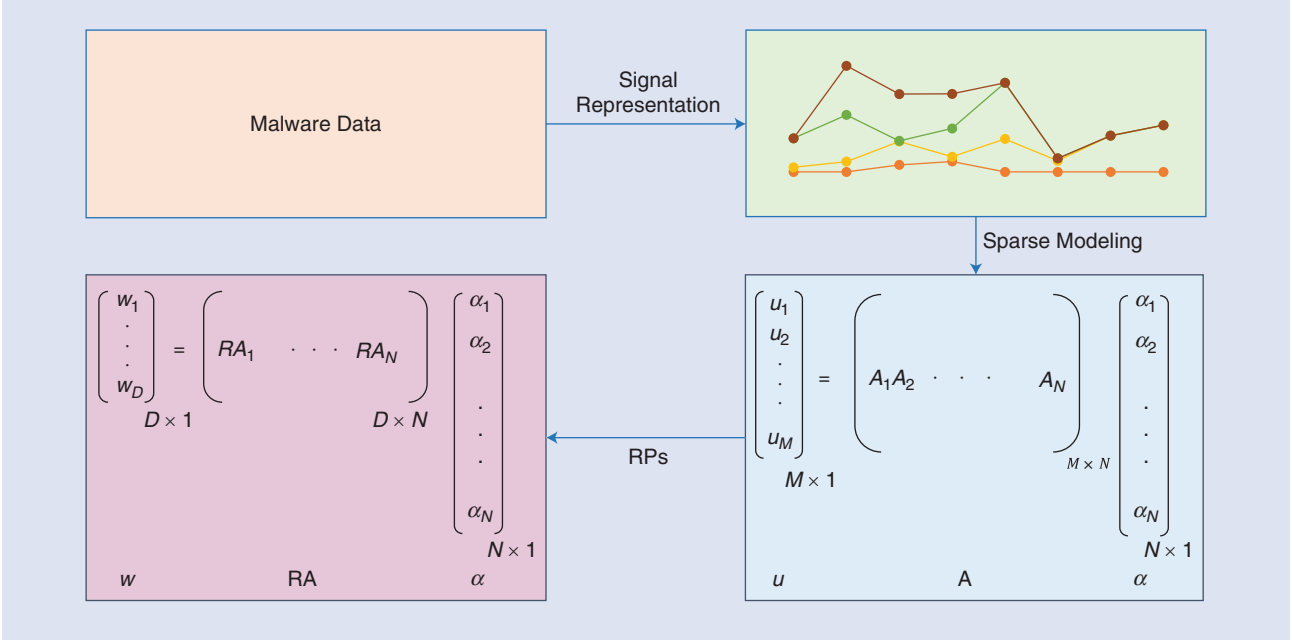

FIGURE 6. The SRC framework for malware classification.

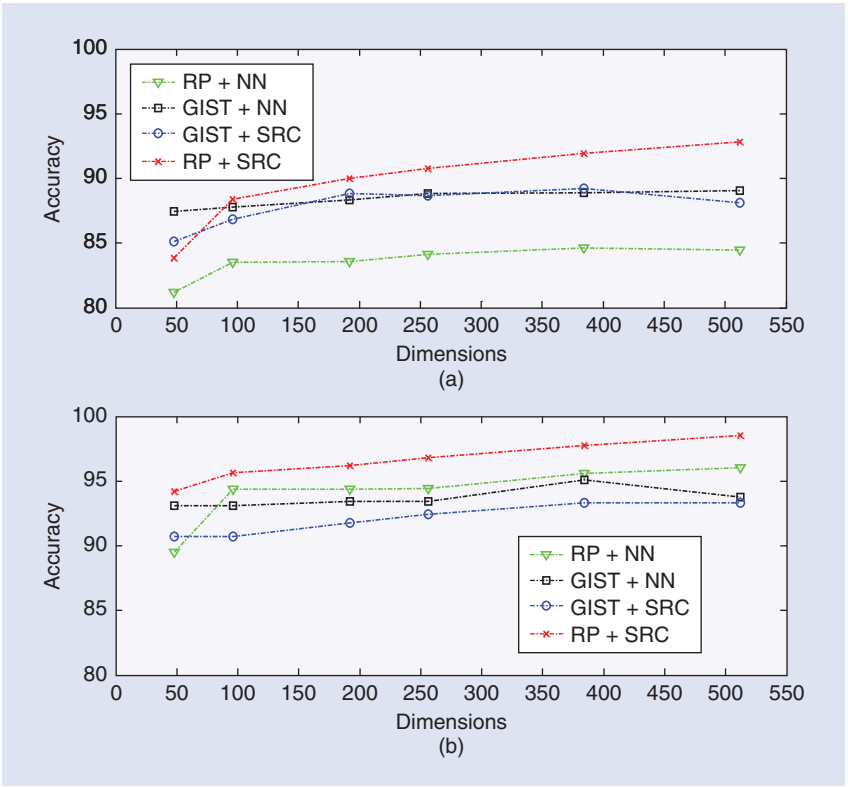

FIGURE 7. Experimental results on (a) Malimg data set and (b) Malheur data set with features using RPs and GIST, and classification algorithms using SRC and NN.

The accuracies for GIST for both classifiers are almost the same. In [15], we extend this approach using a simple thresholding scheme to reject potential outliers in a data set.

Future directions

While we explored signal- and image-based analysis of malware data, a natural complement is to treat the malware as audio-like 1-D signals and leverage automated audio descriptors. Another possible approach is computing image similarity descriptors and/or random projections on all the sections and represent a malware as a bag of descriptors, which can then be used for better characterization of malware. Using the error model in the sparse representation-based malware classification framework, we can determine the exact positions in which the malware variant differs from another variant. This approach can also be used to find the exact source from which a malware variant evolves. Patched malware that attaches to benign software can be identified using this method.

Conclusions

In this article, we explored orthogonal methods to analyze malware motivated by signal and image processing. Malware samples are represented as images or signals. Image- and signal-based features are extracted to characterize malware. Our extensive experiments demonstrate the efficacy of our methods on malware classification and retrieval. We believe that our techniques will open the scope of signal- and image-based methods to broader fields in computer security.

Acknowledgments

We would like to thank Prof. Giovanni Vigna and Prof. Christopher Kruegel of UCSB Seclab for providing the malware data and for their valuable suggestions. We are thankful to our colleagues who worked in this project: Dr. Gregoire Jacob, Dr. Dhilung Kirat and Dr. S. Karthikeyan. We would also like to thank Dr. Sukarno Mertoguno of the Office of Naval Research (ONR) for fruitful discussions. This work is supported by grants ONR N00014-11-10111 and ONR N00014-14-1-0027.

Authors

Lakshmanan Nataraj (lakshmanan_nataraj@ece.ucsb.edu) received his Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara. His research interests include malware analysis, image forensics, and data hiding. He is currently a member of research staff at Mayachitra, Inc.

B.S. Manjunath (manj@ece.ucsb.edu) is a professor in the Department of Electrical and Computer Engineering, University of California, Santa Barbara. His research interests include bioimaging, informatics, media forensics and security, steganography, large-scale image and video sensor networks, and multimedia databases. He is a Fellow of the IEEE.

References

[1] (2015, Dec.). Sony hack: Obama vows response as FBI blames North Korea. [Online]. Available: http://www.bbc.com/news/world-us-canada-30555997

[2] G. Jacob, P. M. Comparetti, M. Neugschwandtner, C. Kruegel, and G. Vigna, "A static, packer-agnostic filter to detect similar malware samples," in *Proc. 9th Int. Conf. Detection of Intrusions and Malware, and Vulnerability Assessment*, July 2012, pp. 102–122.

[3] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

[4] A. Torralba, K. P. Murphy, W. T. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," in *Proc. 9th IEEE Int. Conf. Computer Vision*, Oct. 2003, vol. 1, pp. 273–280.

[5] M. Douze, H. Jgou, H. Sandhawalia, L. Amsaleg, and M. Schmid, "Evaluation of GIST descriptors for Web-scale Image Search," in *Proc. ACM Int. Conf. Image and Video Retrieval*, July 2009, no. 19, pp. 1–8.

[6] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. 8th Int. Symp. Visualization for Cyber Security*, July 2011, no. 4, pp. 1–7.

[7] Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and evolution," in *Proc. IEEE Symp. Security and Privacy*, May 2012, no. 15, pp. 95–109.

[8] (2015, Dec.). Malimg Dataset. [Online]. Available: http://old.vision.ece.ucsb.edu/spam/malimg.shtml

[9] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *J. Comput. Security*, vol. 19, no. 4, pp. 639–668, Dec. 2011.

[10] (2015, Dec.). VirusShare. [Online]. Available: http://www.virusshare.com

[11] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang, "A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in *Proc. 4th ACM Workshop on Security and Artificial Intelligence*, Oct. 2011, pp. 21–30.

[12] D. Kirat, L. Nataraj, G. Vigna, and B. S. Manjunath, "SigMal: A static signal processing based malware triage," in *Proc. 29th Annu. Computer Security Applications Conf.*, Dec. 2013, pp. 89–98.

(continued on page 117)

imaging is the capability of imaging at a frame rate of up to tens of thousands frames/second [5]. While such ultrafast imaging technology offers great opportunities for the improvement of US imaging, especially for fast moving objects such as a beating heart, arterial flow, or a shear wave within the tissue (based on which the tissue elasticity can be quantified), it also raises significant challenges. Techniques to make full use of the GB of data acquired per second are required. There are opportunities to take advantage of the prior knowledge in both the underlying imaging physics and target tissue/organ physiology, and to generate in real-time clinically relevant information that are yet to be fully exploited.

In addition, the complex nonlinear signals generated by MBs provide another avenue for research, as the MB signals can be influenced by many variables related to the in vivo environment, such as blood pressure, proximity to vessel wall, gas saturation, and the mechanical properties of the surrounding tissue. A better understanding of the physics and advanced modeling and signal processing techniques could lead to extracting this clinically relevant information from the MB signals. Additionally, while most clinical US imaging is still in two dimensions, three-dimensional US imaging is arriving and will create further opportunities and challenges for data postprocessing. Finally, molecular imaging using targeted MBs is another exciting area of further development, where more advanced signal processing could help detect and evaluate pathologies at their earliest stage.

Acknowledgments

We would like to thank Dr. Adrian Lim, Prof. David Cosgrove, Prof. Roxy Senior, and Yuanwei Li for providing some of the clinical images used in this article; Dr. Alfred Yu for providing the flow phantom; and the U.K. Engineering and Physical Sciences Research Council for financial support (EP/M011933/1).

Authors

Antonio Stanziola (antonio.stanziola14@ic.ac.uk) is a Ph.D. student with the Ultrasound Laboratory for Imaging and Sensing Group, Imperial College London.

Matthieu Toulemonde (m.toulemonde@ic.ac.uk) is a postdoctoral researcher with the Ultrasound Laboratory for Imaging and Sensing Group, Imperial College London.

Yesna O. Yildiz (y.yildiz11@ic.ac.uk) is a Ph.D. student with the Ultrasound Laboratory for Imaging and Sensing group, Imperial College London.

Robert J. Eckersley (robert.eckersley@kcl.ac.uk) is a senior lecturer at King's College London.

Meng-Xing Tang (mengxing.tang@ic.ac.uk) is a reader (associate professor) of biomedical imaging and the head of the Ultrasound Laboratory for Imaging and Sensing group, Department of Bioengineering, Imperial College London.

References

[1] J. R. Lindner, "Microbubbles in medical imaging: Current applications and future directions," *Nat. Rev. Drug Discov.*, vol. 3, no. 6, pp. 527–533, 2004.

[2] C. Tremblay-Darveau, R. Williams, L. Milot, M. Bruce, and P. N. Burns, "Combined perfusion and Doppler imaging using plane-wave nonlinear detection and microbubble contrast agents," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 61, no. 12, pp. 1988–2000, 2014.

[3] T. Faez, M. Emmer, K. Kooiman, M. Versluis, A. F. W. van der Steen, and N. de Jong, "20 years of ultrasound contrast agent modeling," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 60, no. 1, pp. 7–20, 2013.

[4] F. Lin, C. Cachard, F. Varray, and O. Basset, "Generalization of multipulse transmission techniques for ultrasound imaging," *Ultrason. Imaging*, vol. 37, no. 4, pp. 294–311, 2015.

[5] M. Tanter and M. Fink, "Ultrafast imaging in biomedical ultrasound," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 61, no. 1, pp. 102–119, 2014.

[6] J. M. G. Borsboom, C. T. Chin, A. Bouakaz, M. Versluis, and N. de Jong, "Harmonic chirp imaging method for ultrasound contrast agent," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 52, no. 2, pp. 241–249, 2005.

[7] S. Harput, J. McLaughlan, D. M. J. Cowell, and S. Freear, "Superharmonic imaging with chirp coded excitation: Filtering spectrally overlapped harmonics," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 61, no. 11, pp. 1802–1814, 2014.

[8] M.-X. Tang, H. Mulvana, T. Gauthier, A. K. P. Lim, D. O. Cosgrove, R. J. Eckersley, and E. Stride, "Quantitative contrast-enhanced ultrasound imaging: A review of sources of variability," *Interface Focus*, vol. 1, no. 4, pp. 520–539, 2011.

[9] Y. O. Yildiz, R. J. Eckersley, R. Senior, A. K. P. Lim, D. Cosgrove, and M.-X. Tang, "Correction of non-linear propagation artifact in contrast-enhanced ultrasound imaging of carotid arteries: Methods and in vitro evaluation," *Ultrasound Med. Biol.*, vol. 41, no. 7, pp. 1938–1947, 2015.

[10] W. K. Cheung, D. M. Gujral, B. N. Shah, N. S. Chahal, S. Bhattacharyya, D. O. Cosgrove, R. J. Eckersley, K. J. Harrington et al., "Attenuation correction and normalisation for quantification of contrast enhancement in ultrasound images of carotid arteries," *Ultrasound Med. Biol.*, vol. 41, no. 7, pp. 1876–1883, 2015.

[11] K. Wei, A. R. Jayaweera, S. Firoozan, A. Linka, D. M. Skyba, and S. Kaul, "Quantification of myocardial blood flow with ultrasound-induced destruction of microbubbles administered as a constant venous infusion," *Circulation*, vol. 97, no. 5, pp. 473–483, 1998.

[12] K. Christensen-Jeffries, R. J. Browning, M.-X. Tang, C. Dunsby, and R. J. Eckersley, "In vivo acoustic super-resolution and super-resolved velocity mapping using microbubbles," *IEEE Trans. Med. Imaging*, vol. 34, no. 2, pp. 433–440, 2015.

[13] C. H. Leow, E. Bazigou, R. J. Eckersley, A. C. H. Yu, P. D. Weinberg, and M.-X. Tang, "Flow velocity mapping using contrast enhanced high-frame-rate plane wave ultrasound and image tracking: Methods and initial in vitro and in vivo evaluation," *Ultrasound Med. Biol.*, vol. 41, no. 11, pp. 2913–2925, 2015.

SP

APPLICATIONS CORNER *(continued from page 110)*

[13] L. Nataraj, D. Kirat, B. S. Manjunath, and G. Vigna, "SARVAM: Search And RetrieVAl of Malware," in *Proc. Annu. Computer Security Conf. Worshop on Next Generation Malware Attacks and Defense*, Dec. 2013.

[14] (2015, Dec.). Virustotal. [Online]. Available: https://www.virustotal.com/

[15] L. Nataraj, S. Karthikeyan, and B. S. Manjunath, "SATTVA: SpArsiTy inspired classificaTion of malware VAriants," in *Proc. 3rd ACM Workshop on Information Hiding and Multimedia Security*, June 2015, pp. 135–140.

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Apr. 2008.

[17] D. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 1–53, Jan. 2009.

[18] J. K. Pillai, V. M. Patel, R. Chellapa, and N. K. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1877–1893, Feb. 2011.

SP