

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

170,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Malware: Detection and Defense

Iyas Alodat

Abstract

In today's cyber security landscape, companies are facing increasing pressure to protect their data and systems from malicious attackers. As a result, there has been a significant rise in the number of security solutions that can identify malware. But how do you know if an image file is infected with malware? How can you prevent it from running? This blog post covers everything you need to know about malware in your images and how to prevent them from running. The malware will allow the attacker or un-legitimate user to enter the system without being recognized as a valid user. In this paper, we will look at how malware can hide within images and transfer between computers in the background of any system. In addition, we will describe how deep transfer learning can detect malware hidden beneath images in this paper. In addition, we will compare multiple kernel models for detecting malicious images. We also highly suggest which model should be used by the system for detecting malware.

Keywords: steganography, cyber security, malware, deep transfer learning, keras

1. Introduction

What is a Malware? Malware is an umbrella term that refers to anything that is malicious. It can be used to refer to malicious hackers, viruses, or even worms. Malware is a very broad term that refers to any malicious software. Annoying popups, spyware, viruses, ransomware, and worms are all examples of malware.

Identifying the Types of Malware, There are many different types of malware, depending on what the purpose of the attack is. Viruses are malicious software that can replicate themselves and infect files on your computer. Trojans are malicious software that masquerades as something legitimate, like a helpful PDF reader, but actually do something harmful. Worms are software that spreads across networks and computers, like the dreaded WannaCry attack. Spyware collects information about you and your computer's activity without your knowledge. Ransomware can lock your computer or your files until you pay a ransom. Malware can do many different things, but you can protect yourself by keeping your computer clean and being careful about what you download.

How Malware Gets into Image Files, Malware is most commonly found in image files online. The most common cases of this happening are with stock photos from websites like Shutterstock and iStock. Sometimes, malicious software is also embedded in a company's digital images. This malware can do anything from collecting user information to carrying out a denial-of-service (DoS) attack on your company's

servers. Websites that include images in their content often receive images from a stock photo website. It's possible that the image may include malicious software. If the website does not have an image scanning system in place, malicious software could make its way onto your website without you even knowing.

The crime-as-a-service field is rapidly evolving, making innovative operating emerging trends available to cybercriminals in order for them to successfully achieve their goals. These technologies have evolved into cyber threats that could be suited to the cryptographic protocols used by consumers and businesses to combat cybercrime. One of the major difficulties is undoubtedly malware classification.

Malware that appears "attractive" today will be obsolete tomorrow, filled by others with wholly distinct or improved features [1]. And all the while, newer isolates of malware collaborate with older varieties. As a result, designation in the malicious cyber environment is highly complex [2, 3].

In such a context, traditional cyber risk intelligence rollbacks and indices (IOCs) are insufficient to combat the threat. Who alters his behavior after learning that he has been identified? The biggest strength of cybersecurity deep learning is its capability to benefit from this evolution in real-time and generate classification criteria without the need for human intervention. This allows us to determine whether a person is communicating with their workstation or an automaton in real-time. Or if a cyber criminal is attempting to steal or interact with a user profile from any part of the community (remote access to a Trojan horse).

Many Facebook users revealed when they inspected a partial shot for hidden photograph tags attached to users' photos that images can undertake a lot of data that is typically inaccessible to the human eye. The type of data linked with Instagram and Facebook pictures and photos is not significant compared to the complex approaches used by targeted attacks to create images that can convey malicious code or embezzle user data. In the past several years, there has been a significant increase in malware advertisements in the wild that use the new technique of data encryption to embed subliminal meaning in photos and files.

2. Information's concealment

Steganography can hide code in plain text, such as inside an image file. This means that messages or information can be hidden inside a nonconfidential text as a carrier of these messages and information. In this way, malicious parties use this technology to compromise devices by hosting an image on a website or sending a picture in an e-mail. Hidden data or a carrier file does not have to be images; in fact that digital images are just streams of bytes like any other file making them an especially effective way to hide secret text and other data [4].

The science of steganography is a form of obfuscation that is very different from cryptography, which is the practice of writing encrypted or encrypted messages. The encrypted messages are clearly hiding something: and require specialized decryption methods.

Steganographic letters are similar to conventional letters, but they cleverly hide something unexpected. For instance, consider the following statement: (He eats solely like Lucifer, what other rogue enjoys Durian!). We can determine the core notion behind how to obscure information by reading this communication using a known technique. The hidden message, "Hello, world," is not encrypted; the reader only needs to know how to interpret the message in a specific way to identify it, and we did

not just have to add any extra data to the “carrier” in order to deliver it. Although the technique of hiding the data is far more complex, it is essentially the same concept on a reduced scale.

The mind is interpreting the secret message in plain language in the preceding example. However, computer algorithms read bytes rather than natural words. It turns out that this allows you to hide communications in plain text that are simple for algorithms to perform and evaluate while being nearly hard for humans to uncover without assistance. Indeed, due to the nature of photographic file formats, it is feasible to conceal not just text strings but also complete files in .jpg format and other image formats depending on the technology employed; this may also be accomplished without increasing the overall file size for the original image.

3. Create malware images

We will use photos from both benign and harmful archives to identify photographs using a deep learning system. We will only do a binary characterization (malware and benign class). This method can also be used to achieve multiclass grouping, assuming that each variant of malware file has images that are distinct from the others. If our dataset is complete, we convert all files to 256×256 image pixels (every pixel has a value ranging from 0 and 255) by following the procedures below for each image: First, read 8 bits from the file at a time. Second, treat the eight bits as a binary form and translate it to an integer. Third, enter the pixel value as a number.

A file with a maximum size of 64 KB will fit a 256×256 image. For any file with a size greater than 64 KB, the remaining contents would be dropped. On the other hand, if the file size is smaller than 64 KB, the remaining image would be padded with 0's. Since the identification of malware is performed in real-time, we need to identify the picture as benign or malware within seconds. Keeping the image generation process quick and fast would help us save precious time.

4. Steganography hides information

Take a look at a few of the most basic methods for hiding text in a digital image. One straightforward method is to simply insert the material into the file at the end. These works do not prohibit the photograph from being displayed regularly, nor do they alter its esthetic look. We merely put “hello world” to the file's conclusion. The hex dump output shows us putting the extra bytes.

A program can easily read or discard the plain text string. In this scenario, we'll invert the hexadecimal number and output it in plain English using a software. For example, a received image displays a picture in a photograph viewer application ordinarily, but when examined with the WinRAR archiving utility, we can find that the unpacked.jpg file contains a concealed 28-byte text file.

These types of basic approaches can be helpful in collecting user data, but they do have some disadvantages. First, they inflate the file size, and second, they change the file hash. It's still very convenient for security tools to spot because of its unexpected format.

The best way is to enter into the code at the binary stage and deal with the least important bits (LSBs) of each pixel. Pixels can be represented in a 3-byte color image, one per RGB each (red, green, blue). Suppose we have three bytes

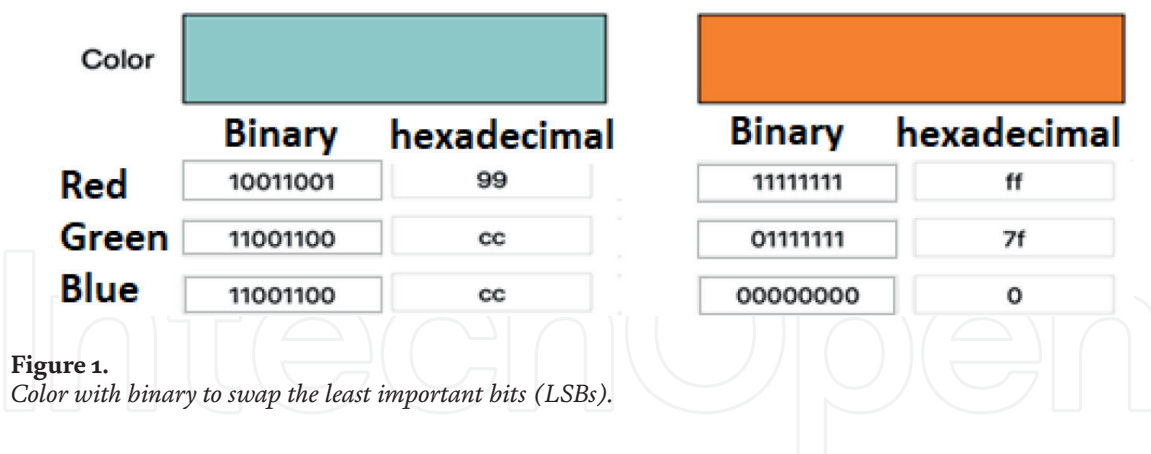


Figure 1. Color with binary to swap the least important bits (LSBs).

representing a particular color as seen in **Figure 1**. You should swap the last four bits of the orange code with the first four bits of the turquoise code, to produce the composite RGB [5].

If we write software to read and extract these last four bits separately, we have effectively hidden the purple signal within the orange color software. Two pixels for the price of one, as there is no increase in file size. We can send our cryptic information without doubling the bandwidth of the actual message or modifying the file format, thus simple detection approaches that rely on examining files to find them are rendered obsolete. In actuality, the code is extremely cluttered before the attacker reassembles it.

In a sense, this ensures that an intrusion will utilize the last four bits of encoding RGB values to write extra information without interfering with the image's graphic display or increasing the file size. Another program will then read the hidden data and use it to reassemble a malicious script or sort customer data.

LSB processing is one of several steganography processes. There are numerous other instances in which photographs and other file formats might be modified to conceal a concealed message. To relay secret communications, the attackers employed information buried in network protocols, a technique known as "network hiding." The approach is the same in both cases: hide in clear view by downloading an invisible message to the accessible carrier.

Steganography for shielding information has affected both Windows and Mac OS operating systems. An intruder has been found to use cryptography to conceal portions of the ransomware attack code, add malicious JavaScript, and even download encryption software.

5. How to detect malware in an image file

In order to detect the presence of malware in an image file, you must first understand the types of malware that can be embedded into an image file. The different types of malware you may find in an image file include: There are many ways malware can hide within an image file, but luckily there are also several ways to detect it. At the network level, malware detection can be done through an antivirus program. Some administrators prefer to use a signature-based antivirus program because it can detect known viruses that are already in the wild. Signature-based detection is particularly helpful against viruses that are polymorphic, meaning they can constantly change their code and evade detection. Other malware may be detected by a heuristic method, which means the antivirus program looks for

suspicious activity. This method can catch new viruses before they are added to the antiviruses' signature database.

6. SHA and MD5 checksum

If you are working with a file that has a large file size, SHA or MD5 checksum can be very useful in checking for malware. Many online tools can do this for you, or you can use a hex editor to calculate and compare the checksum for the file in the image. You'll need to download the file from the source and check the file size, then download the file from your image and check the file size to compare the two. If there is a difference in the file size, you may want to investigate further. If you are working with a smaller file, it may be easier to use a hex editor to view and compare the file directly.

7. Virus and worm signature detection

When dealing with image files and applications, many different types of malicious software can be hidden inside them with little or no indication. It's important to recognize what some of these indicators are so you can protect your network and its users from any potential dangers. In image files, viruses and worms are often obfuscated and hidden inside data sections. Viruses can also be hidden in executables as a DLL, EXE, or other file types. Some viruses and worms, such as Ramen and MyDoom, can be detected by signature because they have been seen before, and antivirus software vendors have created signature definitions to detect them. You can check for DLLs, EXEs, and other executable files hidden in the data section of the image file using a hex editor. You can also use antivirus software to scan the image file and look for specific signatures.

8. Hex editors and PE viewers

Depending on the complexity of the malware, it may be difficult to detect in an image file. Viruses and worms can be difficult to detect, but you can use a hex editor or a PE viewer to check an image's data section. This can also be helpful when detecting malicious code in an image file. You can view the hex data of an image file by opening it in a hex editor. This will show you any data that's been added to the image file, such as hidden code. If you are working with an EXE file, you can use a PE viewer to see the data that has been added. You can also use a debugger to debug the application and find any malicious activity.

9. Experiments setup simulation and dataset

The dataset was dealt with by Hacettepe University's Computer Engineering Multimedia Information Lab. It provides an RGB-based Core Fact Dataset for evaluating vision-based multiclass malware identification studies [6]. We used Keras Framework with TensorFlow in back-end, this about for deep learning libraries. For manipulate and process our images, we use Pandas and Scikit-learn libraries. All experiments ran using Python 3.7 with notebook IDE.

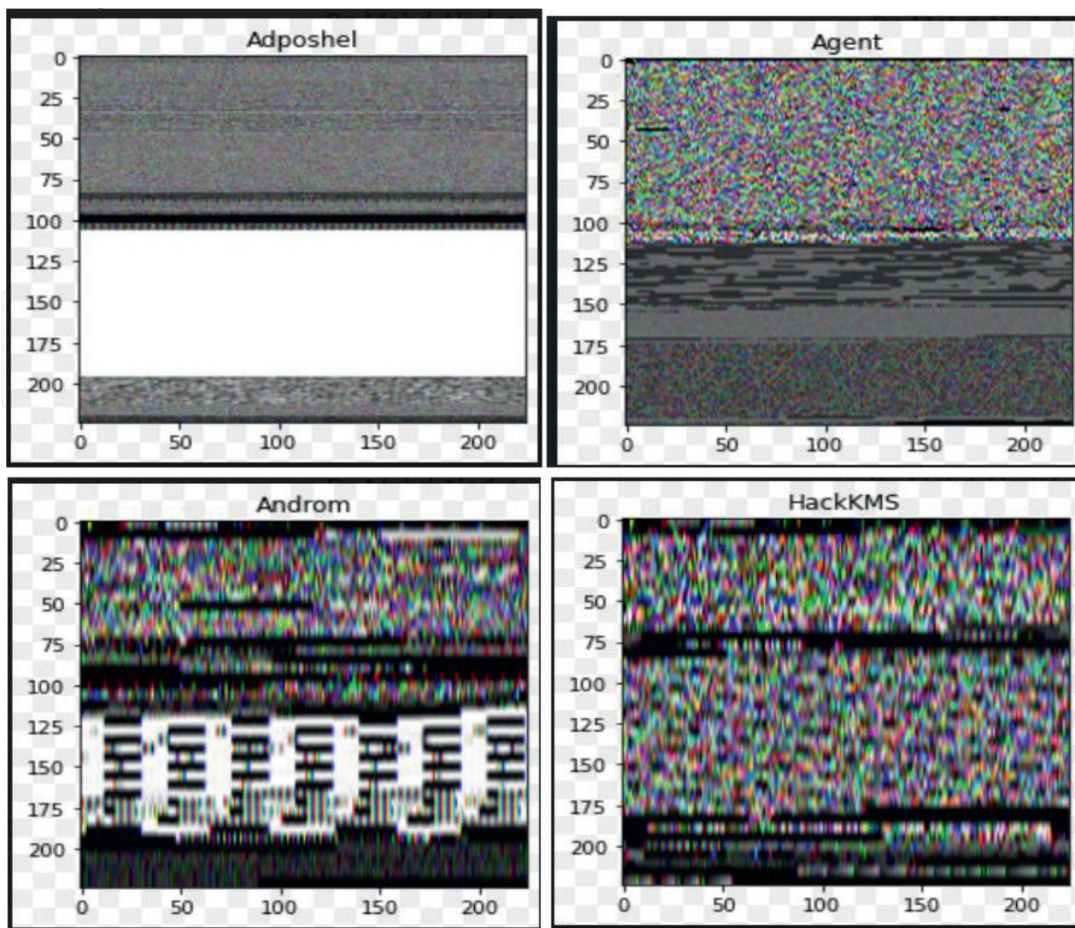


Figure 2.
Malware sample from dataset.

A Convolutional Neural Network (CNN, or ConvNet) is a sort of multilayer neural network designed to recognize visual patterns directly from pixel pictures with minimal pre-processing [7]. **Figure 2** shows an example of malware pictures. The ImageNet project is a massive graphic collection designed to be used in graphical object identification application testing. We utilized Keras, a deep-learning package. We simulate 18 types of malware using several deep transfer learning (DTL) models such as MobileNetV2, VGG16, and ResNet [8–12].

10. Performance evolution

We will utilize Classification Accuracy, Confusion Matrix, and ROC curve to assess the effectiveness of our systems.

Classification Precision It is a model classification metric in which the number of right predictions is compared to the total number of predictions produced by each model. Matrix of Confusion is one of the evaluation techniques that use the model's output and four categories: True positives are values that are supposed to be positive, whereas false positives are values that are expected to be positive but turned out negatively, making them fake. True negative values are those that are predicted to be negative and hence are true. False negative denotes Values that are predicted to be negative but are positive, hence they are false [13, 14].

The ROC curve is a curve that compares two variables, the one being the genuine positive and the other being the false positive. True positive identifies the positive values that were accurately recognized by the model as positive. It defines false positives as negative values that were likewise considered to be negative by the classifier. The ROC curve graphic will show the true positive rate of a system in proportion to its false positive rate at various locations.

11. Results

We present our ideas to each Deep Transfer Learning using a confusion matrix, with each picture explaining a distinct form of malware. As we can see from the list

	Recall	Precision	Score
MobileNetV2	0.077187	0.076349	0.040323
InceptionV3	0.843847	0.834643	0.828221
ResNet50	0.810017	0.800643	0.808221
LittleVGG	0.768251	0.811926	0.775191

Table 1.
 List of rates computed from a confusion matrix.

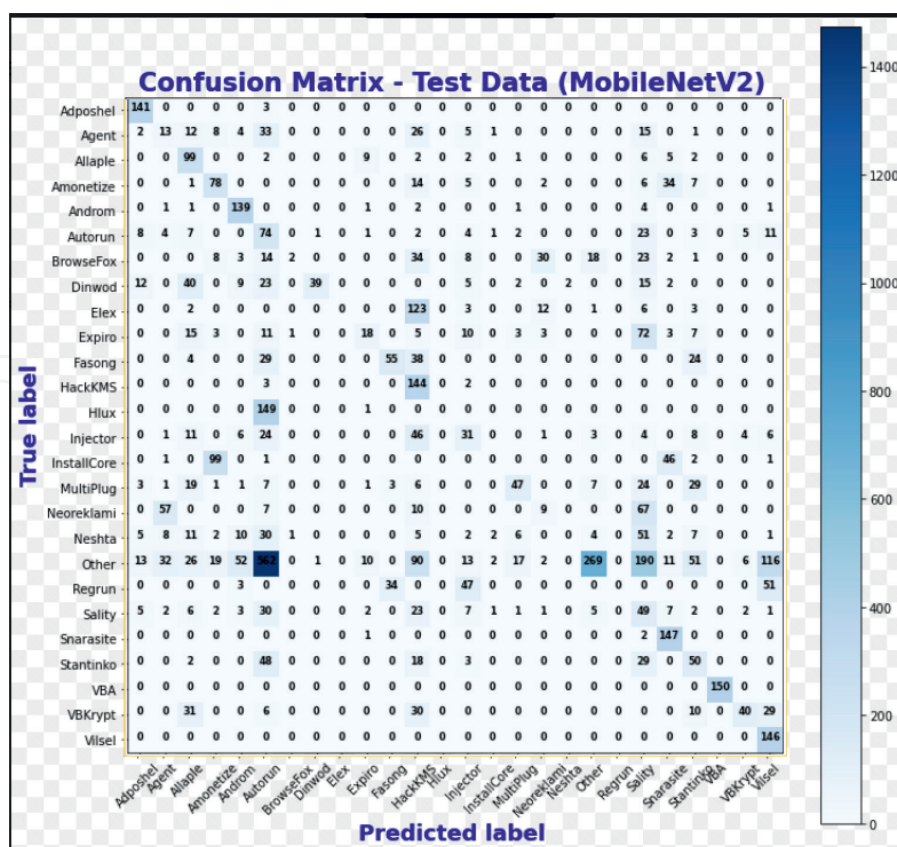


Figure 3.
 Confusion matrix for mobileNetV2.

of rates in **Table 1**, we can determine that the best system for predicting our virus is MobileNetV2, which is the greatest pick from the list of rates.

In **Figure 3**, we can see from the confusion matrix that MobileNetV2 is the best model since it has a significant number of real values on the left and predicted values on the right. The proper productions will occur on the matrix's diagonal.

We have an emphasis on precision or specialization in our work, and these matrices can explain erroneous negatives. We require false negatives matrices for nonmalware detected by malware filters. In other words, positive class malware is malware, while false negatives are not malware. In this case, false negatives are preferable to false positives.

Table 2 shows the training, validation, and testing trials we conducted for each model. At this point, we may infer that one model detects malware far better than another. As the analysis from the confusion matrix shows, the best one is obviously MobileNetV2.

The ROC curve in **Figure 4** shows that it can accurately detect the kind of malware. The capacity to appropriately analyze and recognize a picture from another side whether it was typical.

	Training	Cross-val	Testing
MobileNetV2	0.94819713	0.94819713	0.95199621
InceptionV3	0.84456111	0.84809954	0.88111015
ResNet50	0.83881235	0.82547898	0.83109547
LittleVGG	0.90258367	0.89804627	0.92458974

Table 2.
Data accuracy after transfer learning—Performance metrics.

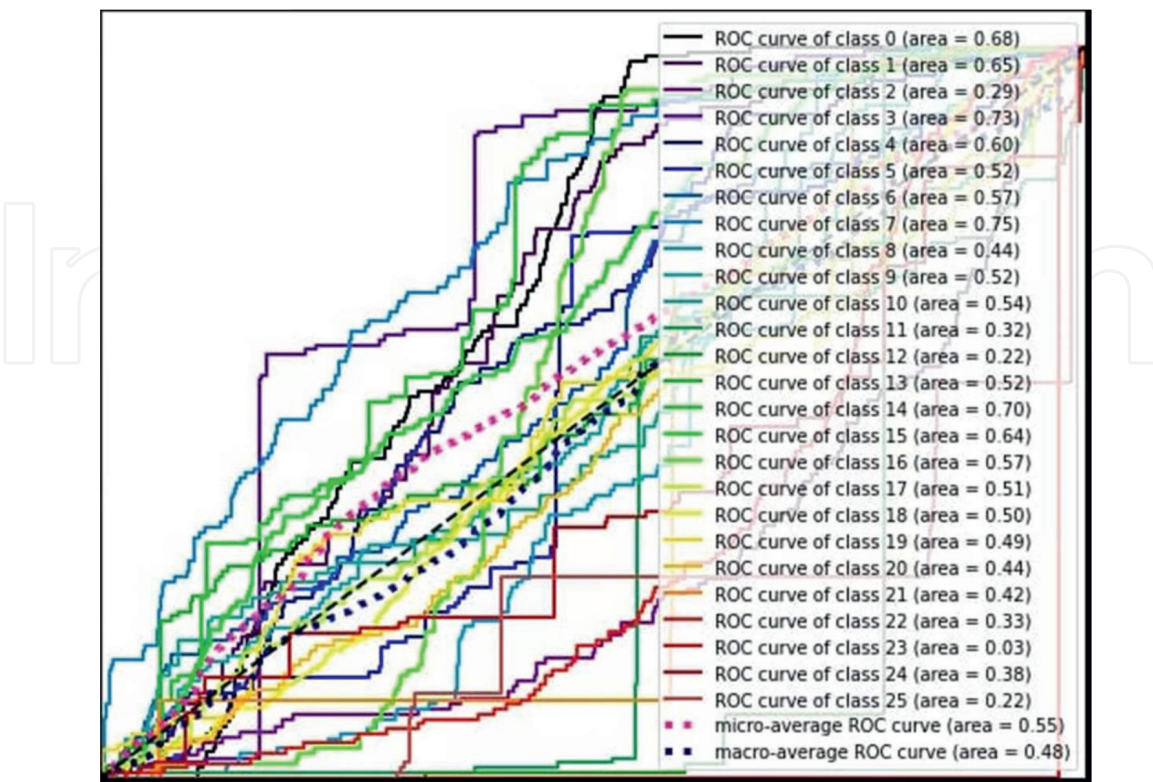


Figure 4.
ROC curve for mobileNetV2.

12. Conclusion

Malware can infect any type of file, including images. Image file types that are most likely to be infected with malware are BMP, JPEG, PNG, and GIF. Additionally, malware can get into image files by being embedded in the image itself during the creation process. If you are using images in your marketing campaign or on your website, you need to make sure they are safe. Check your images to make sure they do not contain any malicious software. If you end up with an infected image, you could put your company's systems at risk. If you are unsure if an image is malicious, you can upload it to a website like VirusTotal. This service will scan the image for malware. This is the best way to make sure your images are safe.


Many photos are transmitted by e-mail and social media, which may be hiding a specific threat. Through the analysis that we have done in this study, we can reduce the dangers of malware hidden behind those images, using artificial intelligence techniques and machine learning, we were able to reduce these risks very well, and also reduce privacy violators by discovering these types of deceptive users through temptation and carrots that the hacker uses for this purpose.

Author details

Iyas Alodat
Jerash University, Jerash, Jordan

*Address all correspondence to: eyas.odat@jpu.edu.jo

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Dini G, Martinelli F, Saracino A, Sgandurra D. MADAM: A multi-level anomaly detector for android malware. In: International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security. Berlin, Heidelberg: Springer; 2012. pp. 240-253
- [2] Chandrasekhar AM, Raghuvver K. Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers. In: 2013 International Conference on Computer Communication and Informatics. 2013. pp. 1-7. DOI: 10.1109/ICCCI.2013.6466310
- [3] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12). Red Hook, NY, USA: Curran Associates Inc.; 2012. pp. 1097-1105
- [4] Al-Juaid NA, Gutub AA, Khan EA. Enhancing PC data security via combining RSA cryptography and video based steganography. Journal of Information Security and Cybercrimes Research. 2018;1(1):5-13
- [5] Islam MR, Siddiqa A, Uddin MP, Mandal AK, Hossain MD. An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography. In: 2014 International Conference on Informatics, Electronics & Vision (ICIEV). 2014. pp. 1-6. DOI: 10.1109/ICIEV.2014.6850714
- [6] Bozkir AS, Cankaya AO, Aydos M. Utilization and comparison of convolutional neural networks in malware recognition. In: 2019 27th Signal Processing and Communications Applications Conference (SIU). 2019. pp. 1-4. DOI: 10.1109/SIU.2019.8806511
- [7] Xiao Y, Xing C, Zhang T, Zhao Z. An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access. 2019;7:42210-42219
- [8] Ahmim A, Maglaras L, Ferrag MA, Derdour M, Janicke H. A novel hierarchical intrusion detection system based on decision tree and rules-based models. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). 2019. pp. 228-233. DOI: 10.1109/DCOSS.2019.00059
- [9] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, AlNemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access. 2019;7:41525-41550
- [10] Riyaz B, Ganapathy S. A deep learning approach for effective intrusion detection in wireless networks using CNN. Soft Computing. 2020;24(22):17265-17278. DOI: 10.1007/s00500-020-05017-0
- [11] Yang L et al. A theory of transfer learning with applications to active learning. Machine Learning. 2012;90(2):161-189. DOI: 10.1007/s10994-012-5310-y
- [12] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning. In: Kůrková V, Manolopoulos Y, Hammer B, Iliadis L, Maglogiannis I, editors. Artificial Neural Networks and Machine Learning – ICANN 2018. ICANN 2018. Lecture Notes in Computer Science. Vol. 11141. Cham: Springer; 2018. DOI: 10.1007/978-3-030-01424-7_27
- [13] Alodat M, Abdullah I. Surveillance rapid detection of signs of traffic Services in Real Time. Journal of Telecommunication, Electronic and Computer Engineering (JTEC). 2018;10(2-4):193-196

- [14] Alodat M. Predicting student final score using deep learning. In: Bhatia SK, Tiwari S, Ruidan S, Trivedi MC, Mishra KK, editors. *Advances in Computer, Communication and Computational Sciences. Advances in Intelligent Systems and Computing*. Vol. 1158. Singapore: Springer; 2021