Machine Learning Detection in Malvertising

Marc Riccione
Texas A&M University
marcr@tamu.edu

Aashish Ananthanarayan
Texas A&M University
aashish.2704@tamu.edu

***Abstract*—Malvertising uses ads on web browsers, emails, and legitimate online advertising networks such as The New York Times, Spotify, and the Atlantic to trick unsuspecting users into clicking on the ad and thus triggering malware [1]. Malvertising is used in two ways to trigger attacks on unsuspecting users. The first way is by tricking users without the user even clicking on the ad. The second way is by tricking users when they click on the ad. Detecting malware in advertisements is a crucial need for ad networks in today's age for free websites that utilize advertisements to make money. A way of detecting malware in web browsers and email is presented through the detection of machine learning. The machine learning detection mechanism can be implemented in a chrome extension. This will allow a user to detect malware in real-time. Also, machine learning detection in malvertising can be expanded to be used by ad networks and publishers as well. This will allow them to screen ads for malware in real-time before posting on their websites.**

I. Introduction

Malvertising is a serious issue that affects users and the publishing companies involved. Malvertising is used to trick users into installing malware, through the use of advertisements. The way that hackers can get their ads onto websites is by buying the ad space or exploiting a vulnerability in the system [3]. This not only compromises the computers that they use but also hurts the reputation of the publisher of that website. Free of use sites are very common in today's age, and the best way in which these free for use sites can stay free is through the use of advertisements [2]. When a user gets malware from a legitimate website, not only is the website's reputation tarnished, but the user's computer is also infected with malware. Once malware can be installed on one's computer, a bad actor can modify or delete files, greatly reduce a computer's performance, and allow hackers to download data on the user.

Malvertising can infect a user's computer without the user clicking on the ad. One way this can be done is through a ”drive-by download” which installs malware on the computer of a user viewing the ad [1]. The ”drive-by download” exploits browser vulnerabilities to infect the user's computer. Another way malvertising can infect someone's computer without them clicking on the ad is by automatically force redirecting the user to a malicious site [1]. Almost half of all malvertising is auto-redirects [2]. A third way that malvertising can be used to infect someone's computer with malware without them clicking on it is by using advertising that isn't wanted, malicious content, or pop-ups [1].

Malvertising can also infiltrate a user's computer when the user clicks on the ads. One way malvertising can infect a user's computer by clicking is by running code that then injects malware into the user's computer [1]. Another way is by lying to the user on the ad's content. The user then clicks on the ad believing that it will bring the user to the website that the ad shows, but instead the ad redirects the user to a malicious website [1]. The third way malvertising can be used to infect a user's computer when the user clicks on the ad is by redirecting a user to a website that appears to not have malicious intent and therefore looks like the website that the user was promised. This website is operated by a hacker who is hoping that the user allows himself to be vulnerable to a phishing attack [1].

As one can see, there are many ways in which malvertising hurts people and their bottom line by steering users away from their websites. The owners of legitimate websites need a good way of detecting malware in advertisements, and so do the users to protect their computers. This is not an easy task because some ad networks receive millions of advertisers, and also users need the detection of malware to happen in real-time right when they open an email or go to a website. Also, the way advertisers get their advertisements on a website is through real-time bidding [1]. Because there is real-time bidding being utilized, it makes it significantly harder for an ad network to test the ads before they are put on a website.

Malware detection can best be detected by using machine learning. We think that this would work because there have been many instances of machine learning detection for malware. Also, the detection has to generalize to many forms of malware. This work was able to detect malware using a convolutional neural network. The final proof of concept would not only have a machine learning algorithm that can detect malware, but also a chrome extension that a user can simply press to scan a web page or email for malvertising.

Through this project, we hope to use machine learning and build an efficient tool to detect malvertising on websites. We have described the related work, system design, and the results of our experiment. Further on, we would also like to touch on some of the improvements that we have made for the project since our presentation. Our machine learning model was completely ready, but our only final stretch goal was to integrate the chrome extension to the model to allow an automated flow, thus ensuring that both work simultaneously to give the user proper results.

II. Related Work

Machine learning has been utilized in the cybersecurity field already to detect malware. One way this is done is by an AI distinguishing between benign and malicious code by using data that has both benign and malicious code [4]. Within these programs, the machine learning program finds traits that are specified by the person who programmed the AI. These traits then have different weights that can be used to put more of an emphasis on one piece of data than the other.

There are many examples of machine learning being used in the real world to detect malware. One example would be using machine learning to separate malware binaries into known families by using features that were extracted from malware samples [5]. Another example would be using machine learning to detect malware on Internet of things. The results from this research had a 97.4 percent accuracy [6]. Another piece of research used machine learning to detect malware in Android apps. This research utilized a weighted distance function, KNN, and Naive Bayes to obtain a 93.27 percent accuracy [7]. These examples show how machine learning is being used successfully in the cybersecurity industry especially when it comes to detecting malware.

III. System Design

For our design and project, we would like to build an extension tool that could detect malvertising taking place in a browser. Just like Google Chrome has an extension for Netflix Party, Grammarly and so on, we plan to add another such button to the side of the browser that the user can download. Once this is done, the extension will help the user figure out whether a particular ad or link has malware in it.

The use will be pretty simple. First, the add pops up on the browser. The tool then detects whether an advertisement has popped up and then starts scanning for malware in the advertisement. Sometimes, advertisements in even webpages/emails might contain links that consist of malware and this could infect the user's machine. The tool can also accept a link and scan for potential malware, thus alerting the user in case their computer might be compromised.

Before building the tool though, we wanted to better understand how malware detection works using machine learning. We first began by looking through articles and research papers to check the work around us and see the different techniques used for malware detection. We would like to utilize one of these techniques in order to build our extension.

Regardless of the input/way the tool functions, the steps to build the tool would primarily be the same. The first step would be to collect a dataset. We first began by working on files that contain malware to see if we could successfully detect the presence of malware in a file. Initially, we referenced [7] for their open-source malware detection dataset which contains multiple asm files. Some of the asm files are infected with malware and some are not. These files were annotated accordingly.

Later on, we came across a dataset that really helped our project improve. Our data contained 137597 samples of Portable Executable Headers. This is a file format used mostly as executables in Windows, but is majorly used to inject malware. We also found that most of the users affected by malvertising were Windows users, so it made more sense to tackle this problem first. These files were then classified into malignant and benign. 41323 of the files were malignant and 96274 of them were benign. A total of 56 features were available, however, we only ended up using the 8 most important features in order to avoid overfitting.

Once the dataset was ready, the second process was to train the data on the model. The dataset was split into 80 percent training and 20 percent testing. After the split was accomplished, the model was then trained. The model used in this case was a random forest consisting of about 50 trees that classify using majority voting. Once training was complete, we then tested the weights on the 20 percent testing data.

Another addition to our design was that we decided to implement a test website for the user to evaluate on. The testing environment is a simple react app that has a few links the user can click around. The first link sends us to a benign test file and the second link sends us to a malignant file. Now, the user might not know which file is what and this testing interface can be used to test our tool. The reason we implemented the testing interface was to provide users with a secure way to test our product and not visiting any sites that might be very harmful. Also, we too were a little scared of harming our own computers since we are very close to finals week and we did not want anything to go wrong especially during classes at home, so we decided to introduce the safer testing environment.

IV. Initial Evaluation and Results on the Initial Dataset

After training, we then tested the model on the split dataset and we received a decent accuracy of roughly 0.6. We had a pretty low loss to report as well, something around 0.12. However, lower losses have been reported in other papers and hence, we would like to re-train the model again and fine tune it with many more parameters. This would greatly help improve the performance when we put the model into the tool.

These are just preliminary results we received. Over the next week, we will be working on making these results better, hopefully we can touch a 0.9 accuracy and a very low loss.

V. Final Evaluation and Results

After we worked on the new dataset with the portable executable headers, we noticed that our accuracy scores improved significantly. Because our data was so massive, consisting of more than 130000 data samples, the results we received were also well grounded and concrete. Our final score for the model was around 98.9 percent which was really good, especially since it was on the testing data.

VI. Conclusion

Thus, we were successfully able to complete the first task of our project, where we set out to see if we could detect malware through machine learning models. Now that this has been accomplished, we will now try to train the model on advertising datasets. Once this is done, the weights from the trained model will be added to the browser extension and that could be used to test malvertising in real time.

We are extremely excited to see where this project is headed and hope to have a fun demo ready for everyone!

REFERENCES

[1] “What is Malvertising: Examples &amp; How It Differs From Ad Malware: Imperva,” Learning Center, 29-Dec-2019. [Online]. Available: https://www.imperva.com/learn/application-security/malvertising/.

[2] J. Pleger, “Malvertising is a Problem for Everyone because it can Impact every part of What we do on the Internet,” MarTech Advisor, 31-Mar-2017. [Online]. Available: https://www.martechadvisor.com/articles/ads/malvertising-is-a-problem-for-everyone-because-it-can-impact-every-part-of-what-we-do-on-the-internet/.

[3] A. Fiscutean, “What is malvertising? And how to protect against it,” CSO Online, 20-Mar-2019. [Online]. Available: https://www.csoonline.com/article/3373647/what-is-malvertising-and-how-you-can-protect-against-it.html.

[4] K. Crawley, “Machine Learning Malware Analysis - What You Must Know,” CCSI, 29-Apr-2019. [Online]. Available: https://www.ccsinet.com/blog/machine-learning-malware-analysis/. [Accessed: 02-Nov-2020].

[5] M. Hassen, M. M. Carvalho and P. K. Chan, ”Malware classification using static analysis based features,” 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1-7, doi: 10.1109/SSCI.2017.8285426.

[6] H. Naeem, B. Guo and M. R. Naeem, ”A light-weight malware static visual analysis for IoT infrastructure,” 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, 2018, pp. 240-244, doi: 10.1109/ICAIBD.2018.8396202.

[7] D. Ö. Şahın, O. E. Kural, S. Akleylek and E. Kiliç, ”New results on permission based static analysis for Android malware,” 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-4, doi: 10.1109/ISDFS.2018.8355377.

[8] M. F. Rafique, M. Ali, A. S. Qureshi, A. Khan, and A. M. Mirza, “Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique,” arXiv.org, 26-Nov-2019. [Online]. Available: https://arxiv.org/abs/1910.10958. [Accessed: 02-Nov-2020].