IBM researchers develop a technique to virtually patch vulnerabilities ahead of threat

February 15 2019, by Ingrid Fadelli

Fady Copty, IBM Research

Researchers at IBM have recently devised a new technique to virtually

patch security vulnerabilities before they are found. Their approach, presented at the International Workshop on Information and Operational Technology, co-located with RAID18, leverages testing techniques for supervised learning-based data generation.

"While researching a solution to find security vulnerabilities in popular software, we paused to think about the following problem: We know practically and theoretically that it is impossible to find all vulnerabilities in an application, and the security community is in a constant race to discover those vulnerabilities in the hope of finding them before the bad guys do," Fady Copty, lead researcher of the study, told *TechXplore*. "This means enforcing regulations and constantly deploying security patches to systems."

Deploying a security patch on an application is a tedious and time-consuming task, which entails a series of steps: identifying the vulnerable version of the application, managing this vulnerability, delivering the patch, deploying it and then restarting the application. Often, patches are deployed over long periods of time, hence applications can remain vulnerable for a period after a vulnerability has been discovered. To speed up this process, researchers have recently introduced virtual patches, which are enforced using intrusion detection and prevention systems.

"Virtual patching is based on a semi-manual technique of analyzing threats (application-input that demonstrates a vulnerability), and extracting the signature that identifies the vulnerability," Copty explained. "It is a useful technique, but still requires that the vulnerability itself be identified, which is an np-complete problem. There is an entire industry around this cycle of vulnerability discovery and patch. But what if we could create a virtual patch that predicts those vulnerabilities ahead of the threat discovery? At first, this sounded like a futuristic task, but with some insights from security testing techniques,

one can find a very nice direction."

Generally, security vulnerabilities are revealed by looking at inputs that should have previously presented an application error. This is because error handling is usually perceived as less important compared to developing the application's basic functions, thus it is addressed at a later stage.

"If we can do a good job at automatically creating a virtual patch that augments the SW developers' work on error handling, we can achieve the ahead-of-threat task," Copty said.

Copty and his colleagues decided to address this problem using machine learning techniques. They ran various testing tools on a given application to generate data, then used this data to train their DNN model.

"We used testing techniques that create millions of sample inputs for the application, and then ran the application with those inputs to determine the classification labels for the inputs: benign, error, or malicious," Copty explained. "Since we were looking at error handling, we merged the error and malicious classes into one class. This provided us with a classic supervised learning setup, where we trained a model to predict whether a new sample is benign or malicious."

Rather than achieving ahead-of-threat virtual patching for a single application, the researchers wanted to create an automatic system that could be used to patch a variety of applications. To enhance the generalizability of their model, they refrained from using manual feature extraction methods.

"We also wanted to eventually deploy this in an intrusion detection system," Copty explained. "This meant that the prediction had to be near-real-time. A great solution for these requirements can be found in

DNNs. DNN prediction is very fast and it is thought that DNNs do not require any feature extraction at all."

Copty and his colleagues trained a DNN model on the data that they had previously generated. The model they used, which combines a convolutional neural network (CNN) and a recurrent neural network (RNN), achieved remarkable results in predicting vulnerabilities ahead-of-threat.

"How do you test the ability to patch ahead of threat discovery? The answer is simple: we go back in time," Copty said. "We used old versions of the applications for the data generation phase, trained the model using this data, and tested the models on threats found years later and documented in the CVE database. This gave us amazing results in ahead-of-threat patching, where the model was able to predict threats found only years later. We know that this is still in the research phase and we have succeeded only on a small number of applications. However, this technology has the potential to be a game changer in the security landscape, helping defenders stay one step ahead of the attackers."

In the evaluations carried out by the researchers, their model successfully detected LibXML2 and LibTIFF vulnerabilities ahead of threat, with accuracies of 91.3% and 93.7%, respectively. To enhance their results, they expanded their model by adding a path that includes basic feature extraction, based on automatic knowledge that was extracted in the testing phase, followed by a CNN.

In the future, their technique could help developers to patch software vulnerabilities faster and more effectively, before they are actually exposed. The researchers plan to continue working on their approach, exploring its effectiveness in patching a broader range of vulnerabilities.

"Thanks to Reda Igbaria, we have now been expanding this research into more applications and recently demonstrated ahead-of-threat virtual patching for the HeartBleed vulnerability," Copty added. "We will continue targeting more applications and enhancing our data generation techniques as well as our DNN structure, and automating search of the best DNN structure."

This work was supported by the EU H2020 SMESEC project.

More information: Deep ahead-of-threat virtual patching. DOI: 10.1007/978-3-030-12085-6_9. link.springer.com/chapter/10.1 …/978-3-030-12085-6_9