Fight phishing with Machine Learning - the Fresh-Phish project

A WhoisXML API customer success story

In spite of its simplicity, phishing is one of the most intensive harmful activities on the Internet, having a tremendous negative impact. Machine learning is the maybe most exciting, paradigm changing software technology of our age. And it has become a part of the armamentary of the fight against phishing. WHOIS and DNS data are necessary for the actuation this powerful weapon. The data available from WhoisXML API, Inc's WHOIS API and DNS lookup API services were used in one of the pioneering projects.

A key ingredient of a phishing attack, either e-mail based or done via a social network is a webpage serving as a trait, a lure and a catch: the victim is convinced to visit the link received via the initial message. The page appearing there frequently resembles the design of some trusted entity (a bank, a company, etc.) and asks for sensitive data of the person to be swindled, which then end up in the attacker's database. A key element of the fight against phishing is thus the identification of these malicious sites.

But the task is far from being easy. An obvious idea is to keep the malicious URLs on a blacklist that is available publicly, and indeed, popular browsers use such blacklists to warn the user if a potentially dangerous site is visited. But how to maintain such a database? Cybercriminals are clever enough to make these websites appear, change and disappear frequently; they can collect a number of victims before they get reported by humans. Clearly they should be detected automatically somehow.

The task of selecting potentially malicious sites from amongst a huge number of pages automatically scraped from the Web cries for a machine learning approach. After decades of scientific development, machine learning become now a well-established and amazingly successful technology becoming an inherent part of our everyday life.

As a basic idea, the algorithm needs a data set for training: a number of sites about which we know from some other source whether they are legitimate or phishy. These can be obtained from already existing databases such as Phishtank. In the learning phase, the algorithm will process these data to get into a phase when it can already tell, with some accuracy, whether a site not seen by the algorithm before is phishy or not. But how does the algorithm become so clever? We have to tell it what are the **features** to observe: some characteristics of the page and its contents that can potentially be related to the phishy nature. (What is weird about these algorithms is that we shall never know intuitively **how** they exactly they come to the decision. They just work, with a surprising accuracy.) What are these features? Surely the age of the domain, the validity of its registration, or the plausibility of the claimed identity of the registrant are important points to check by the human or machine making the decision, and these come from WHOIS and DNS data.

H. Shirazi and his coworkers at the Department of Computer Science, Colorado State University, were the first to introduce an open framework, "Fresh-Phish", for creating current machine learning

data for phishing websites, as they had observed that there were no well-elaborated training data (URLs+features+labels) available for these algorithms. Using a number different features, including the aforementioned DNS and WHOIS-based ones, they had built a a large labeled dataset and analyzed several machine learning classifiers against this dataset to determine which is the most accurate. They published the details of their approach first on the prestigious IEEE International Conference on Information Reuse and Integration (IRI) in 2017, which was soon followed by a journal article in the International Journal of Multimedia Data Engineering and Management. The framework is implemented in Python, using the most popular and advanced Machine Learning libraries, including TensorFlow and scikit-learn. The project is available on GitHub: https://github.com/FreshPhish/dataset. In addition to the source code of the program, the created datasets are there, too, and they keep on uploading fresh ones.