Web Phishing Detection

Abstract

Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet. Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity. It will lead to information disclosure and property damage. This paper mainly focuses on applying a deep learning framework to detect phishing websites. This paper first designs two types of features for web phishing: original features and interaction features. A detection model based on Deep Belief Networks (DBN) is then presented. The test using real IP flows from ISP (Internet Service Provider) shows that the detecting model based on DBN can achieve an approximately 90% true positive rate and 0.6% false positive rate.

Phishing Feature Extraction and Definition

First, we get real traffic flow from ISP. The data set includes traffic flow for 40 minutes and 24 hours. We construct the graph structure of traffic flow and analyze the characteristics of web phishing from the view of the graph.

Each piece of data contains the following fields.(i): user node number.(ii): user IP address.(iii): access time.(iv): Uniform Resource Locator, access web address.(v): request page source.(vi): user browser type.(vii): server address to access.(viii): User Cookie.

A graph is mathematical structures used to model pairwise relations between objects. It is also a very direct way to describe the relationship between nodes in a network. The relationship between the nodes on the Internet can also be expressed through the graph structure. Therefore, we construct a graph to store the real traffic flow data and describe the relationship between the nodes in traffic flow.

Give an undirected graph , where includes two kinds of node:(i)user node ;(ii)access and . denotes an access relationship between , and .

The vertices of the graph are as follows:(i)User node has one attribute: total access times (vertex out-degree).(ii)User node has two attributes: total accessed times (vertex in-degree) and website registration time.

The edges of the graph are as follows:(i)The number of visits: which corresponds to the number of occurrences of the edge, the number of times an AD may have access to a URL, or the number of direct links between two URLs, depending on the corresponding vertex type.(ii)Cookie: the cookie field in the access record.(iii)UA: User Agent in the access record.

3.2. Feature Definition

We define two kinds of features to detect web phishing, and they are an original feature and interactive feature.

3.2.1. Original Feature

There are some features in the phishing URL, such as special characters. We definite these features in URL as an original feature as follows:(i): there are special characters in URL, such as @, Unicode, and so on. Those special characters are not allowed in a normal URL.(ii): there are too many dots or less than four dots in normal URL.(iii): the age of the domain is too short. For example, the age of the normal domain is more than 3 months.

In order to quantify the above characteristics, all the characteristic values are binary, that is, one of 0 or 1. Intuitively, the more of the 1 appear in the feature, the higher the likelihood that the site will be a phishing site.

3.2.2. Interaction Feature

There are some features in graph , such as access frequency. We define these features through a node relationship as interaction feature as follows:(i): in-degree of node from is very small. In general, the normal websites do not link to phishing sites. The phishing sites are directly accessed.(ii): out-degree of node is very small. In order to get personal private information, the phishing sites are usually terminal websites and do not link to the other sites.(iii): the frequency of from is one. In general, one user accesses the phishing site only one time and the user cannot access the phishing site more than one time.(iv): when accesses , user browser type is not the main browser. Well-known browser vendors often have a built-in filtering phishing site plug-in. A user who uses unknown browsers is more likely to access the phishing sites.(v): there is no cookie in user. The phishing site does not leave its cookie in user.

REFERENCES

1. Mishra, D. Irwin, P. Shenoy, J. Kurose, and T. Zhu, "GreenCharge: Managing renewableenergy in smart buildings," IEEE Journal on Selected Areas in Communications, vol. 31, no. 7, pp. 1281–1293, 2013.

   View at: Publisher Site | Google Scholar

2. P. Yi, T. Zhu, G. Lin et al., "Energy scheduling and allocation in electric vehicle energy distribution networks," in Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2013, USA, February 2013.

   View at: Google Scholar

3. T. Zhu, Z. Huang, A. Sharma et al., "Sharing renewable energy in smart microgrids," in Proceedings of the 2013 ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2013, USA, April 2013.

   View at: Google Scholar

4. https://www.mozilla.org/en-US/.
5. https://www.google.com/chrome/browser/index.html.

ADVANTAGES:

- Requiring low resources on host machine
- Effective when minimal FP rates are required.
- Mitigate zerohour attacks.
- Mitigate zerohour attacks.
- Constuct own classification models.

DISADVANTAGES:

- Mitigation of zero-hour phishing attacks.

- Can result in excessive queries with heavily loaded servers.
- -Higher FP rate than blacklists. -High computational cost.
- Time consuming.
- Costly.
- Huge number of rules.

CONCLUSION:

In this paper, we analyze the features of phishing websites and present two types of feature for web phishing detection: original feature and interaction feature. Then we introduce DBN to detect phishing websites and discuss the detection model and algorithm for DBN. We train DBN and get the appropriate parameters for detection in the small data set. In the end, we use the big data set to test DBN and TPR is approximately 90%.

REFERENCES:

1. https://en.wikipedia.org/wiki/Web_service.

2. O. Adam, Y. C. Lee, and A. Y. Zomaya, “Stochastic resource provisioning for containerized multi-tier web services in clouds,” IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 7, pp. 2060–2073, 2017.

View at: Publisher Site | Google Scholar

3. T. Bujlow, V. Carela-Espanol, J. Sole-Pareta, and P. Barlet-Ros, “A survey on web tracking: Mechanisms, implications, and defenses,” Proceedings of the IEEE, vol. 105, no. 8, pp. 1476–1510, 2017.

View at: Publisher Site | Google Scholar

4. H.-C. Huang, Z.-K. Zhang, H.-W. Cheng, and S. W. Shieh, “Web application security: Threats, countermeasures, and pitfalls,” The Computer Journal, vol. 50, no. 6, pp. 81–85, 2017.

View at: Publisher Site | Google Scholar

5. https://en.wikipedia.org/wiki/WeChat.