Detecting Ambiguous Phishing Certificates using Machine Learning

Sajad Homayoun
DTU Compute
Technical University of Denmark
Kongens Lyngby, Denmark
sajho@dtu.dk

Kaspar Hageman
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
kh@es.aau.dk

Sam Afzal-Houshmand
DTU Compute
Technical University of Denmark
Kongens Lyngby, Denmark
saaf@dtu.dk

Christian D. Jensen
DTU Compute
Technical University of Denmark
Kongens Lyngby, Denmark
cdje@dtu.dk

Jens M. Pedersen
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
jens@es.aau.dk

***Abstract*—Recent phishing attacks have started to migrate to HTTP over TLS (HTTPS), making a phishing web page appear safe to the user's browser despite its malicious purpose. This paper benefits from both digital certificates and domains related data features to propose machine learning-based solutions to predict digital certificates involved in HTTPS as phishing or benign certificates. In contrast to previous works that consider this a binary classification problem, we take into account that a certificate can be partially benign and phishy simultaneously. We propose a multi-class classifier and a regressor to classify these ambiguous certificates, in addition to benign and phishing certificates, where the 'phishyness' of a certificate is expressed as a value between 0 and 1 for the regressor. We apply our method to a set of certificates obtained from certificate transparency logs and show that we can classify them with high performance. We extend our validation by evaluating the performance of the model over time, showing that our model generalizes over time on our training data set.**

***Index Terms*—Digital Certificate, Phishing, Machine Learning, Feature Extraction**

I. Introduction

Phishing is still one of the most common attacks which has a plethora of reported cases on a daily basis [1]. In these attacks, victims are persuaded in disclosing sensitive information by an attacker that impersonates a legitimate organization or person. Modern and more advanced attackers have begun to use phishing websites offering legitimate digital certificates as a way to persuade the users into diverging their information willingly, by making the websites look more legitimate through visible padlock icons in browsers. Although phishing websites with legitimate certificates can be more effective than their unencrypted counterparts, they have the drawback from the attacker's perspective that they have are visible in publicly available certificate transparency (CT) logs.The inclusion of certificates in these logs is a requirement to be a trusted website in most modern mainstream browsers [2]. Monitoring CT logs for newly-issued certificates may enable the detection of malicious phishing websites using legitimate certificates in the earlier stages of the attack. In fact, prior research has proposed automated solutions for detecting digital certificates being involved in phishing attacks [3]–[6]. Most of previous research has tackled this as a binary classification problem, with certificates being considered either *malicious* or *benign*. Identifying a high-quality ground truth is challenging, because the label of a certificate tends to based on the label of the domain name or website that serves a given certificate. Not only is the nature of the domain not always clear-cut benign or malicious [3], [7], but there may also be conflicting domain labels for a given certificate [8].

In this paper, we propose a novel machine learning technique to address the issues with this task in the state of the art. In contrast to prior work, our solution considers the classification of certificates to be a multi-class classification problem, and is capable of finding certificates that fit neither the traditional benign or malicious labels, but are rather considered ambiguous or 'conflicted'. More specifically, the contributions in this paper are summarized as follows:

- we present a novel labeling mechanism that takes into account the individual labels of the domains that are being covered by a certificate,
- our study is not limited to benign and phishing certificates but also considers certificates covering both benign and phishing domains,
- We combine all features described in prior works to build a pool of features for machine learning tasks,
- we study how our features can predict a *phishyness score* for each certificates,
- we validate our methods using a time-based cross validation scheme and show that our classifier and regression models achieve a high performance.

This paper is divided as follows. Section II gives brief background on digital certificates and Section III reviews some related works. Section IV describes our data collection

pipeline extracting certificate/domain related features. Section V & VI presents our classification and regression scenarios, respectively. Section VII evaluates our scenarios using a time-based approach. We discuss the achievements and limitations in Section VIII, and conclude the paper in Section IX.

II. Background

The Transport Layer Security (TLS) protocol suite provides encryption functionality that other networking protocols such as HTTP and IMAP. Besides encrypting the communication between a client and server, TLS guarantees authenticity by operating under a public key infrastructure (PKI). When establishing a connection, a server provides a digital certificate (as defined in the X.509 format), which consists of a public key (whose private key is only known to the server operator), a set of identities (usually one or more domain names) and a signature from a trusted third-party. These third parties, or Certificate Authorities (CA), only sign new certificates after successfully validating that the requestor of the certificate can demonstrate ownership of all identities to be included in the certificate. Clients can verify the authenticity of a certificate by validating that the signature was created using a private key that the client inherently trusts, as part of the certificate root store installed on their system. Originally, a certificate contained a single *subject* field, which indicated for what domain name the certificate was valid. An extension was later introduced to support multiple domain names to be included, named the `Subject Alternative Name` extension, and the domain names embedded in a certificate are therefore commonly refered to as SANs.

In 2011, two CAs issued certificates for high-profile domain names for malicious actors, which allowed these actors to perform large-scale impersonation attacks. As a response, the Certificate Transparency (CT) framework was developed with the intention of monitoring the certificate issuance behavior of the CAs. In this framework, CAs are encouraged to submit newly-issued certificates to CT logs, publicly-available repositories of certificates. Browser vendors started to require certificates to be included in these logs, and as a results the CT logs capture nearly all certificates that are issued worldwide.

III. Related Work

Artificial intelligence based approaches for classifying malicious certificates have been studied during the past few years. Inspired by classifiers for URL detection, there have been works on classifiers that utilize the information that can be gained from the certificates itself which is encompassed in the work presented in this paper. Mohammad *et al.* provide a dataset that illustrates how using the certificate issuer from HTTPS information can be used to high avail which has bled over to multitudes of other works [9]. It ought to be noted that certificate information often needs additional information extraction of viable features due to the limited information offering as compared to URL databases [10].

Across works in the area of detecting phishing websites based on certificates, there is a recurring theme of feature engineering. Mishari *et al.* proposed a selection of certificate features from phishing websites to be used for training a random forest, decision tree and nearest neighbor to detect phishing website which denoted an accuracy above 85% [11]. A more holistic real-time version with a similar approach and results was proposed by Dong *et al.* where the framework of feature extractor, classifier and decision process was included with phishing information used to train the same models in addition to naive bayes tree, logistic regression, decision table and k-nearest neighbor [12]. A deep neural network version was also proposed for classification by Dong *et al.* which denoted an accuracy above 95% [13]. Relying on deep learning, Torroledo *et al.* use a long short-term memory (LSTM) based model for detection [6]. Recently, Drichel *et al.* propose a pipeline where they could easily test a lot of these classifiers using CT log data which may help in classifier selection process [3].

The security research community tends to rely on information from domain names to provide a label for certificates that cover the given domain names. Typically, a list of popular domains serves as a ground truth for benign domains, and lists of phishing domains or URLs (e.g., PhishTank [14]) as benign domains. Certificates that are served by these two respective domain group can as a result be labeled as benign and phishing as well. Hageman *et al.* showed that labeling certificates in such as fashion may result in mislabeling [8]. Content Delivery Networks (CDNs) such as CloudFlare and Incapsula that provide HTTPs based protection services issue certificates for sets of domains originating from different owners. In case this set of domains include both malicious and benign domains, labeling these certificates is a challenge.

Even though a significant effort was made by the security community towards the detection of certificates involved in malicious activity, to the best of our knowledge no one has recognized it as an ambiguous problem that should not be tackled as a binary classification problem.

IV. Data Collection

To train and validate our approach, we rely on a vast number of labeled certificates. First, we describe how we extract a relevant label and feature space from a certificate, and then explain how we obtained a large dataset of certificates.

A. Feature extraction and labeling

Figure 1 shows the feature extraction and label extraction process. For each unlabeled certificate, and a set of labeled domains, it produces a feature space and a label for the certificates. This set of labeled domains is prepared in advance and contains both benign (i.e., domains that are – with high confidence – have not been part of a phishing attack) and phishing domains (i.e., domains that have been observed as part of a phishing attack). The resulting feature space is a combination of features extracted from the certificate itself, and aggregated features extracted from the list of SANs that are covered by the certificate. For the feature extraction components, we rely on a combination of features that have

been used in prior research [3]–[5] or are derived from insights from other work [8] resulting in 107 features. Due to page limit, we have uploaded a list of our features at our page on the internet[1]. The first 58 features are extracted from the X.509 certificates themselves, and include features such as the signing parameters (e.g., key size, signature algorithm), extensions (i.e., the presence and content of certain X.509 extensions) and the composition of the subject field. The remaining 49 features are extracted from the lexical properties of SANs covered by the certificate, such as the presence of particular keywords or features related to the characters composition and diversity in the string. Each certificate covers a variable number of SANs, and the feature space of these individual SANs are condensed in a fixed-length feature space. We rely on simple statistical functions (i.e., min, max, mean, median) to summarize, and express the diversity of, the numerical domain features and compute a ratio for condensing binary domain features. The dataset contains a significant number of duplicate samples, which we filter out. It is not uncommon for certificates to be renewed after they expire, covering the same SANs and being signed by the same CA with the same parameters, resulting in all our extracted features to remain identical[2].

The label of a certificate is inferred from the collection of the labels from the SANs. Depending on which model is being trained, the label is one of three classes (benign, phishy or conflicted) or a continuous "phishyness" score. In the first case, a certificate is benign or phishy when the list of SANs is only composed of benign or phishing domains respectively (and may include unlabeled domains as well). A certificate covering both at least one benign and phishy domain is considered conflicted, as it is not trivial to claim the maliciousness of the certificate. In the latter case, we use a function of all SANs covered by the certificate for computing a phishyness core. This score (s_i) is expressed as a ratio of the number of phishing domains (p_i) and phishing domains, benign domains (b_i) and unlabeled domains (u_i):

$$s_i = \frac{p_i}{p_i + b_i + u_i} \tag{1}$$

Note that the label extraction is only done during the training process, and not during the operations of the framework.

B. Dataset

To train and validate our machine learning models, we collected a vast collection of certificates. This requires a set of certificate for which a ground-truth is known. We can rely on known phishing and benign domains and infer the ground-truth of certificates issued for those domains. Under the assumption that a highly-popular domain is inherently abused for phishing attacks (e.g., `youtube.com`), we rely on the top one million most popular domains from the Tranco list [15] as a ground truth of benign domains. For establishing a ground truth of phishing entities, we collect phishing URLs from the eCrime Exchange (ECX) platform [16], a platform in which various anti-phishing organizations share newly identified phishing URLs with one another. The root domains from those URLs (e.g., `example.org` from `https://www.example.co.uk?help`) form the basis of our set of phishing domains. There is an overlap between this set of domains and the benign domains, since some popular domains host some user-generated phishing content, such as Google Forms and Facebook. As such, we remove any of the top 1 million domains from the ECX domains to form our ground truth of phishing domains. Furthermore, we take into account that the nature of a phishing domain can change over time (e.g., a domain may have been registered for benign purposes for years, after which it was registered by phisher and used for hosting phishing content). It is common for phishing domains to only be abused for several days [17]. In the label aggregation phase of Figure 1, a domain is only considered "phishy" in the context of a particular certificate, if the validity period overlaps with the identification of a phishing URL associated with the domain in the ECX platform.

Similar to [3], we rely on certificates from the CT logs as a basis for our ground truth. From both of our domain sets, we sampled 10,000 domains each, and collected all certificates that cover any of these 20,000 resulting domain names [8]. As a result, our dataset contains not only the certificates that are currently deployed on web servers – a common method for related work to retrieve their certificate data from [5] –, but also historical data and certificates used for non-HTTPS related services, such as mail servers. The certificates were collected from the Censys search engine [18], which provides extra information to these certificates, most notably the validation status of three root stores (Microsoft, Mozilla's NSS and Apple). We discard all certificates that, according to Censys, do not have any valid certificate chain to any of the three root stores.

Table I: Collected dataset after removing duplicates.

Samples Type	Number of Samples
Benign Certificates	213,353
Phishing Certificates	46,256
Conflicted Certificates	15,578

V. Scenario 1: Distinguishing Between Phishing and Benign Certificates

Figure 2 depicts the underpinnings of a multi-class classification training and testing phase. The training phase comprises two steps: (1) feature engineering and preprocessing and (2) training the multi-class classifier. The former receives benign (B), phishing (P), and conflicted (C) certificates and prepares data format for the training algorithm and filters features with low variance or constant values. This step is also responsible for normalizing the values of different features. The latter outputs a trained classifier to separate between phishing, benign, and conflicted certificates. The training step is not limited to

[1]https://phish-certs.github.io/

[2]The only distinction between these certificates are the timestamps when they become active and expire.

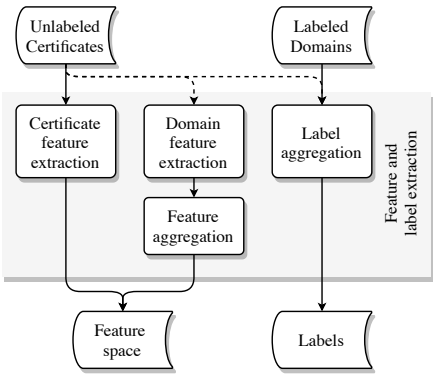

Figure 1: The feature engineering and preprocessing pipeline that turns unlabeled certificates and labeled domains in a labeled feature space. The dashed lines between the certificate dataset and the processing modules represent the set of SANs covered by the certificate.

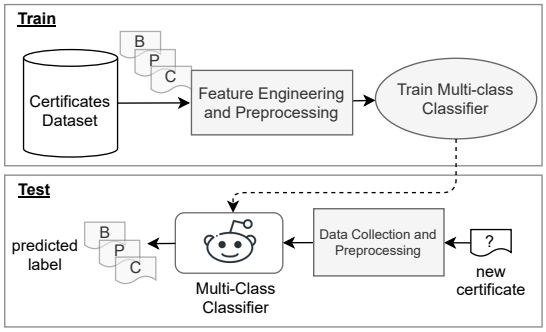

Figure 2: Multi-class classification to distinguish between Phishing, Benign and Conflicted certificates.

certain multi-class classification algorithms. In the test phase, for classifying new certificates, we first have a data collection and preprocessing step to extract certificate related and domain related features for feeding the multi-class classifier. Then the classifier outputs a label for the new certificate.

A. Experimental Results

To show the performance of our extracted features in separating phishing, benign and conflicted certificates, we selected five well-known classification algorithms namely *stochastic gradient descent (SGD)*, *k-nearest neighbors (kNN)*, *random forest (RF)*, *decision tree (DT)* and *support vector machines (SVM)*. We used *VarianceThreshold* from *sklearn* [19] for features selection with $threshold = (.9 \times (1 - .9))$. We used simple classification algorithms to show the robustness of our algorithms in classifying phishing certificates. To avoid the complexity of parameter tuning for each algorithm, we used default parameters given by *sklearn* Python library. As kNN needs k as a required parameter, we set k as the square root of the number of training samples ($k = \sqrt{N}$). We use F1 Score as our comparison metric because it expresses the precision and recall in a single metric [20]. Table II compares our trained classifiers with different metrics for 5-fold cross

Table II: F1 Score of the classifiers under 5-fold evaluation.

Fold	SGD	KNN	DT	RF	SVM
1	0.78	0.74	0.96	0.99	0.84
2	0.76	0.74	0.97	0.99	0.84
3	0.75	0.74	0.97	0.99	0.83
4	0.79	0.74	0.97	0.98	0.84
5	0.76	0.74	0.97	0.99	0.84
avg.	0.77	0.74	0.97	**0.99**	0.84

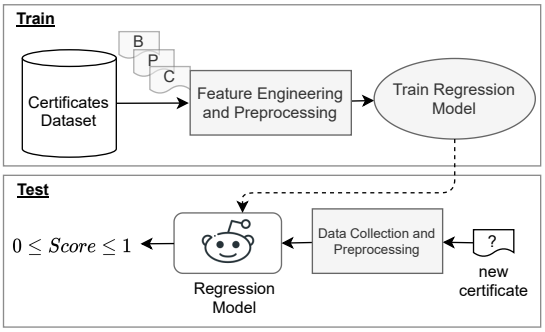

Figure 3: Regression model to predict phishyness score between 0 and 1 for each certificate.

validation. Our evaluation shows that our proposed features for detecting phishing certificates can help to build high quality classifiers. The RF classifiers generated higher performance in comparison to other classifiers.

VI. Scenario 2: Predicting Phishyness Scores

Figure 3 explains training and testing phases of a regression model to predict a phishyness score for each certificate. Similar to *Scenario 1*, this scenario has one step for preprocessing and one step for training the regression model. In the preprocessing step we calculate a phishyness score for each certificate based on equation (1). This step is also responsible for normalizing the values of different features as a data preparation task. The output would be a phishyness score for each certificate which can be between 0 and 1. In the test phase, for predicting a phishyness score for new certificates, the data collection and preprocessing step extracts all required features (certificate related and domain related features), and then the trained regression model outputs a score.

A. Experimental Results

To show the performance of our extracted features, we applied different regression algorithms on our dataset. To avoid parameter tuning of different algorithms, we applied each algorithm using default parameter set from *sklearn* [19]. Table III compare the results achieved by different algorithms such as *lasso regression*, *ridge regression*, *ElasticNet*, *random forest regressor (RFR)*, *decision tree regressor (DTR)* and *bagging regressor (BR)*. We use *Root Mean Square Error (RMSE)* given by $\sqrt[2]{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ as it is commonly used in regression analysis to verify experimental results. RMSE is the standard deviation of the residuals, which is errors between real value

Table III: RMSE of regression models under 5-fold evaluation.

Fold	Lasso	Ridge	ElasticNet	RFR	DTR	BR
1	0.37	0.2	0.37	0.04	0.06	0.05
2	0.37	0.2	0.37	0.04	0.06	0.05
3	0.37	0.2	0.37	0.04	0.06	0.04
4	0.37	0.2	0.37	0.04	0.06	0.04
5	0.37	0.2	0.37	0.04	0.06	0.05
avg.	0.37	0.20	0.37	**0.04**	0.06	0.05

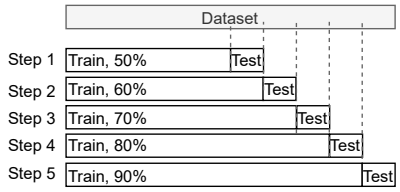

Figure 4: Time-based cross validation with 10% test data.

(y) and predicted value ($\hat{y}$) for all samples (n). *RFR* achieved the best RMSE equal to 0.04.

VII. Validation

The CT framework was created in 2011 and as a result, the CT logs consist of certificates spanning almost a decade. In cross validation, both the training and validation sets included mixed samples from different periods in time. As such, this validation does not evaluate how well the model generalizes to changes in the TLS ecosystem over time. An example of a major shift was the introduction of *Let's Encrypt*, the first certificate authority that issued certificates for free fully automated, which suddenly enabled small websites to serve their content over HTTPS.

We perform a time-based cross validation to evaluate the generalization of our models over time. In this evaluation, we take the first 50% of our certificates, as defined by their validity date, and produce the performance metrics over the next 10% of certificates. We repeat this process by taking the first 60, 70, 80 and 90% of certificates and test on the next 10% (see Figure 4). Tables IV show the F1 scores of time-based validation for the classifiers. Comparing Tables II & IV proves that our classifiers should be retrained on need data as there is a decrease in the F1 Score for classifying recent certificates. Our trained random forest classifier could achieve an F1 score of 0.84, while it could get to 0.99 in our 5-fold cross validation (II). The kNN model gave the worst results in both 5-fold and time-based cross validations, which shows that our feature space and dataset need more complex classification to separate phishing, benign and conflicted samples.

Table V describes RMSE of each studied regression model. Our time-based evaluation shows that the RFR and BR as ensemble-based algorithms have achieved RMSE of 0.05, which is best among our regression algorithms. The lasso and ElasticNet regression algorithms achieved the worst RMSE, which is 0.33. We are interested in a more in-depth look into the errors that RFR as our best model produces. As such, we calculate the error level (i.e., the absolute difference between

Table IV: F1 Score the classifiers under time-based evaluation.

Step	SGD	KNN	DT	RF	SVM
1	0.81	0.60	0.79	0.85	0.66
2	0.75	0.59	0.84	0.85	0.62
3	0.72	0.59	0.77	0.84	0.65
4	0.72	0.58	0.79	0.81	0.61
5	0.74	0.59	0.81	0.83	0.65
avg.	0.75	0.59	0.80	**0.84**	0.63

Table V: RMSE of the regression models under time-based evaluation.

Step	Lasso	Ridge	ElasticNet	RFR	DTR	BR
1	0.37	0.21	0.37	0.05	0.07	0.05
2	0.34	0.18	0.34	0.04	0.06	0.05
3	0.31	0.17	0.31	0.04	0.06	0.04
4	0.34	0.20	0.34	0.05	0.08	0.06
5	0.30	0.20	0.30	0.05	0.08	0.06
avg.	0.33	0.19	0.33	**0.05**	0.07	**0.05**

the predicted value and the real value) for each sample. We plot these figure in a cumulative distribution function in Figure 5. The figure illustrates for instance that 93.5% of samples has an error level of smaller than 0.01, which we believe gives an acceptable approximation of the actual value.

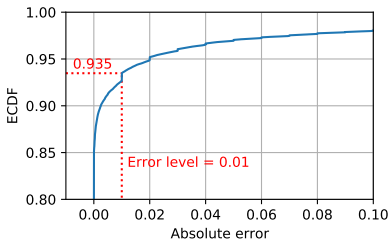

Figure 5: ECDF of the error level of real vs. predicted errors for all test samples for time-based cross validation for all test samples of all steps at Figure 4 (n=123,835).

VIII. Discussion

Our results have shown that our proposed approach can classify with a high performance, and generalizes well over time on historical data. We deliberately selected more classical and simpler classifier and regression models over more novel models, such as deep neural networks, to show that the selected features are powerful and avoid parameter settings.

a) Adverserial robustness: A major challenge for employing machine learning models is an adversarial environment, and we should consider the robustness of the model against behavioral changes of phishers trying to evade our model. Evasion here is the ability of an attacker to modify the feature space of a certificate to circumvent a 'phishing' label to be produced by the classifier. A number of features are domain name independent, and are controlled by the CA rather than the phisher (who merely requests the issuance of the certificate), and can therefore only be influenced by the

attacker by requesting their certificate from a different CA, which may have monetary consequences. The domain-related features in the feature space are derived from *all* SANs covered by the certificate. These features are changed by a phisher by requesting certificates for different sets or, or even individual, domain names, which has once again a monetary impact and may also impact the hosting infrastructure that phishers resort to. As future work, we consider evaluating the performance of our proposed approach in the absence of domain-related features or CA related features to emulate eliminating features due to the evasion strategies of phishers.

b) Ground truth challenges: As described in Section III, our work is not the first attempt at certificate classification for various security purposes. However, to the best of our knowledge we are the first to acknowledge that there can be conflicts in the label for a certificate. As a result, it is difficult to compare our results with prior work in a fair representative manner.

By monitoring CT logs, we are merely observing snapshots of a domain, and miss the changes of the domain afterwards. Phishers are known to compromise (i.e., hack) existing benign websites, re-purposing them for malicious purposes. As such, we may be labeling certificates as phishing or conflicted due to a domains being reported as phishing domains months after the certificate was issued, even though the certificates at time of issuance should have been labeled differently. We devised a method to express the maliciousness of a certificate based on the composition of labels of the domains covered by the certificate. Even though this method provides a continuous scale to put certificates on, one should be careful with relying on the results for any automated decision making. Blocking traffic to web servers serving these certificates may block benign traffic and disproportionately hit organisations that provide security services. We believe that the results can instead be highly valuable as a warning signal for security researchers or regulators to follow up on manually.

IX. Conclusion

In this paper, we proposed a novel method to automatically classify digital certificates as benign, as used in phishing attacks, or assign it an conflicted label. Our work is motivated by prior work on phishing certificate classification and on the existence of ambiguous certificates that are associated with both benign and phishing domains. We consider both a (1) multi-class classification problem, where certificates can be benign, phishy or conflicted, and (2) a regression problem where certificates have a phishyness score. By training different machine learning models in both scenarios, we show a highly performant system. Furthermore, we evaluate the resulting classifiers and regressors using time-based cross validation to show that our approach generalizes decently over time.

X. Acknowledgment

This work was supported by Innovation Fund Denmark (IFD) under the SecDNS project. The authors acknowledge researchers at CSIS Security Group A/S (https://csis.com/) for their constructive recommendations on this project.

References

[1] FBI Internet Crime Complaint Center, "Internet crime report 2020," 2020. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf

[2] B. Laurie, "Certificate transparency," *Commun. ACM*, vol. 57, no. 10, p. 40–46, Sep. 2014. [Online]. Available: https://doi.org/10.1145/2659897

[3] A. Drichel, V. Drury, J. von Brandt, and U. Meyer, "Finding phish in a haystack: A pipeline for phishing classification on certificate transparency logs," in *The 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–12.

[4] E. Fasllija, H. F. Enişer, and B. Prünster, "Phish-hook: Detecting phishing certificates using certificate transparency logs," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2019, pp. 320–334.

[5] J. Li, Z. Zhang, and C. Guo, "Machine learning-based malicious X.509 certificates' detection," *Applied Sciences*, vol. 11, no. 5, p. 2164, 2021.

[6] I. Torroledo, L. D. Camacho, and A. C. Bahnsen, "Hunting malicious TLS certificates with deep neural networks," in *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 64–73. [Online]. Available: https://doi.org/10.1145/3270101.3270105

[7] S. Maroofi, M. Korczynski, C. Hesselman, B. Ampeau, and A. Duda, "COMAR: Classification of compromised versus maliciously registered domains," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, Sep. 2020. [Online]. Available: https://doi.org/10.1109/eurosp48549.2020.00045

[8] K. Hageman, E. Kidmose, R. R. Hansen, and J. M. Pedersen, "Can a TLS certificate be phishy?" in *18th International Conference on Security and Cryptography, SECRYPT 2021*. SCITEPRESS Digital Library, 2021, pp. 38–49.

[9] R. Mohammad, F. Thabtah, and T. Mccluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, pp. 443–458, 08 2013.

[10] U. Meyer and V. Drury, "Certified phishing: Taking a look at public key certificates of phishing websites," pp. 211–223, Aug. 2019. [Online]. Available: https://www.usenix.org/conference/soups2019/presentation/drury

[11] M. Mishari, E. De Cristofaro, K. Eldefrawy, and G. Tsudik, "Harvesting SSL certificate data to mitigate web-fraud," *CoRR*, vol. abs/0909.3688, 01 2009.

[12] Z. Dong, A. Kapadia, J. Blythe, and L. J. Camp, "Beyond the lock icon: real-time detection of phishing websites using public key certificates," in *2015 APWG Symposium on Electronic Crime Research (eCrime)*, 2015, pp. 1–12.

[13] Z. Dong, K. Kane, and L. Camp, "Detection of rogue certificates from trusted certificate authorities using deep neural networks," *ACM Transactions on Privacy and Security (TOPS)*, vol. 19, p. 5, 09 2016.

[14] Cisco Talos Intelligence Group, "Phishtank," n.d. [Online]. Available: https://phishtank.org/

[15] V. L. Pochat, T. V. Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019. [Online]. Available: https://doi.org/10.14722/ndss.2019.23386

[16] Anti-Phishing Working Group, "The APWG eCrime Exchange (eCX)," https://apwg.org/ecx/, n.d., accessed: 30-09-2021.

[17] M. Wullink, M. Muller, M. Davids, G. C. M. Moura, and C. Hesselman, "Entrada: enabling DNS big data applications," in *2016 APWG Symposium on Electronic Crime Research (eCrime)*, 2016, pp. 1–11.

[18] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, Oct. 2015, pp. 542–553. [Online]. Available: https://doi.org/10.1145/2810103.2813703

[19] "scikit-learn: machine learning in python — scikit-learn 1.0 documentation," https://scikit-learn.org/stable/, (Accessed on 10/15/2021).

[20] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K.-K. R. Choo, and D. E. Newton, "Drthis: Deep ransomware threat hunting and intelligence system at the fog layer," *Future Generation Computer Systems*, vol. 90, pp. 94 – 104, 2019.