Hunting Malicious TLS Certificates with Deep Neural Networks

Ivan Torroledo
Cyxtera Technologies
ivan.torroledo@cyxtera.com

Luis David Camacho
Cyxtera Technologies
luis.camacho@cyxtera.com

Alejandro Correa Bahnsen
Cyxtera Technologies
alejandro.correa@cyxtera.com

ABSTRACT

Encryption is widely used across the internet to secure communications and ensure that information cannot be intercepted and read by a third party. However, encryption also allows cybercriminals to hide their messages and carry out successful malware attacks while avoiding detection. Further aiding criminals is the fact that web browsers display a green lock symbol in the URL bar when a connection to a website is encrypted. This symbol gives a false sense of security to users, who are in turn more likely to fall victim to phishing attacks. The risk of encrypted traffic means that information security researchers must explore new techniques to detect, classify, and take countermeasures against malicious traffic. So far there exists no approach for TLS detection in the wild. In this paper, we propose a method for identifying malicious use of web certificates using deep neural networks. Our system uses the content of TLS certificates to successfully identify legitimate certificates as well as malicious patterns used by attackers. The results show that our system is capable of identifying malware certificates with an accuracy of 94.87% and phishing certificates with an accuracy of 88.64%.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Artificial intelligence**;

KEYWORDS

Web certificates; TLS; SSL; Malware; Phishing; Machine Learning; Network Monitoring; Deep Learning.

ACM Reference Format:
Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. 2018. Hunting Malicious TLS Certificates with Deep Neural Networks. In *11th ACM Workshop on Artificial Intelligence and Security (AISec '18), October 19, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3270101.3270105

1 INTRODUCTION

The movement of many applications and services from traditional to digital business channels has come with many benefits - and just as many problems. Digitalization has fostered the creation new vulnerabilities inherent to internet behavior, such as man-in-the-middle attacks. In the past few decades, there has been a large increase in traffic encryption, through using Transport Security Layer (TLS) certificates, to enhance businesses security. According to Cisco Systems, encryption increased almost 91% from 2015 to 2016 [1]. Furthermore, Gartner states that, if this trend continues, it is expected that more than 80% of internet traffic will be encrypted by 2019 [19].

From the perspective of the threat actor, encryption is an essential tool. In recent years, attackers have started to adapt their operations to incorporate encryption. According to a Trustwave Global Security report from 2017, almost 36% of malware detected that year had used encryption [5]. Since Firefox and Chrome browsers introduced the *secure* green padlock symbol to their URL bar, the security company Netcraft has seen a 300% increase in the number of phishing sites created and validated with a security certificate from their website [8]. Indeed, as shown in Figure 1, the percentage of phishing sites using the *https* protocol grew from 5% in the fourth quarter of 2016 to 23% in the third quarter of 2017.

How does encryption impact attackers' strategies? Encrypted traffic enables users to hide their communications, thereby avoiding a third-party who could spy, manipulate or steal transmitted information, such that it weakens the capability of attackers to use Man-in-the-Middle attacks, in counterpart, enhances the transmission of malware attacks without being detected by security systems. Finally, for phishing attacks, encryption produces benefits as a side effect rather than a direct utility. According to Forrester,

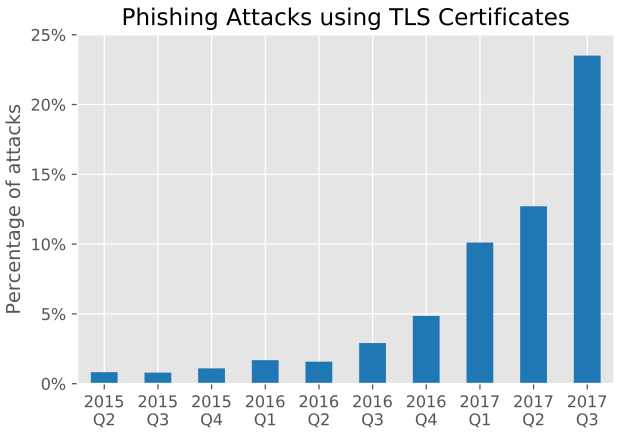
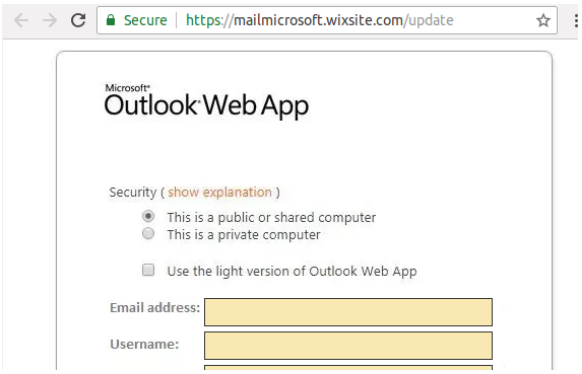


Figure 1: Evolution of phishing attacks using TLS [17].

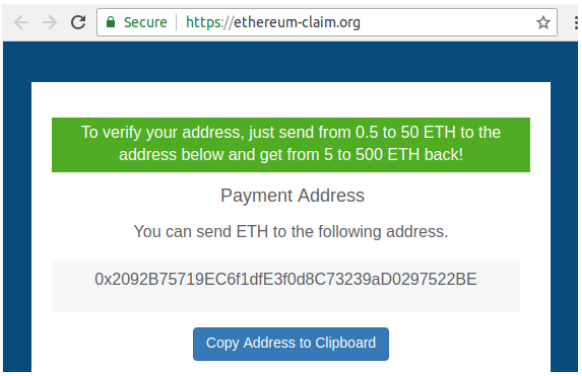
Figure 2: Ultrabank secure icon.

AISec '18, October 19, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-6004-3/18/10...$15.00
https://doi.org/10.1145/3270101.3270105



(a)



(b)

Figure 3: Examples of phishing attacks using TLS.

users have been educated to trust symbols and signs in User Interface (UI) design of most browsers [8]. In a recent survey, Forrester researchers asked users what they think the *secure* symbol and *https* flag from the URL shown in Figure 2 mean. Unsurprisingly, 82% of those who responded said they thought the website was safe, 66% said the website was trustworthy, 32% believed that the website was private. Three-quarters of respondents said they thought the symbol meant that the website is encrypted - which is the only right answer. This phenomenon, known as the *https* paradox, states that users have learned to recognize the *https* protocol as a guarantee that they are connecting to legitimate websites when, in fact, it simply means that the website has a secure channel for transmitting data. The incorrect interpretation of the current UI design of mainstream browsers and the education around it have enabled attackers to defraud users by leveraging the sense of safety and confidence provided by TLS certificates and *https*. Figure 3 shows some examples of phishing sites that appear to be trustworthy due to the *secure* symbol.

As encryption becomes a widely used feature in internet traffic, but awareness of what TLS certificates represent still lags behind, the issue of how to protect end users from cyberattack becomes increasingly relevant. In this paper, we propose a novel system to detect malicious TLS certificates using deep neural networks. The system learning process uses the content of each certificate to detect known malicious patterns used by attackers. Our results show that the system is capable of identifying malware certificates with an accuracy of 94.87%, and phishing certificates with an accuracy of 88.64%.

Table 1: Web certificate validation levels.

Validation Level	Initials	Description
Domain Validation	DV	Domain administration and email has the same domain is being issued.
Organization Validation	OV	DV, organization name legally exists and location.
Extended Validation	EV	OV, organization name legally exists and running, phone number and physical address match and buyer persona is authorized.

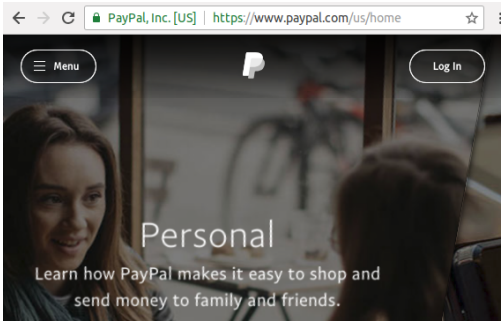
The remainder of this paper is divided as follows. In Section 2, we provide a detailed description of TLS certificates and basic theory of deep neural networks. Then, in Section 3, we describe previous approaches for detecting malicious activity using certificates. In Section 4, we present our proposed system to detect malicious certificates using deep neural networks. In Section 5, we describe the process of feature creation and most relevant models tested during the experiment. Lastly, we show our experimental results in Section 6, and the conclusions of this work are discussed in Section 7.

2 BACKGROUND

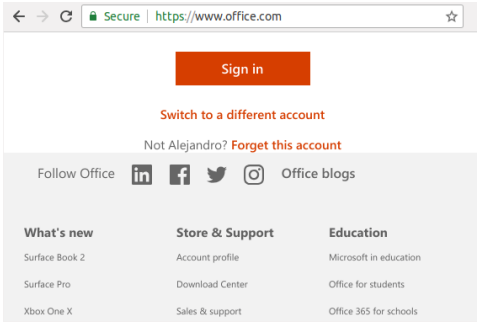
2.1 SSL and TLS certificates

A digital certificate is defined as an unencrypted file attached to a public encryption key [15], it contains organization details about the owner of the certificate and encryption keys. The certificates used to encrypt web traffic are known as SSL and TLS. The encryption system used by TLS is based on the RSA standard for symmetric encryption where two random keys are generated for one public and one private [13]. The encrypted communication process starts after generating public and private keys by the party who has generated keys. They encrypt the message using the private key and keep it. The message is then sent to the receiver along with public key. The message can only be decrypted with the public key. To send a message back, the new message is encrypted using public key and can only be decrypted with the private key. Those encrypted messages also contain a digital certificate that has identification details. This process keeps communication private between both parties, and both parties only [12].

Encryption is widely used across the internet to secure communication channels and to protect transmitted information, so it cannot be read or manipulated by a third party. Nowadays there are three standards of SSL [11] and 2 more of TLS [14, 16]. SSL certificates can be created by anyone, therefore this allows anyone to encrypt and secure any communication channel. But recipients can not just accept any certificate, as this may be interpreted as a security risk. Browsers only accept as secure certificates generated by Certificates Authorities, and these authorities are selected

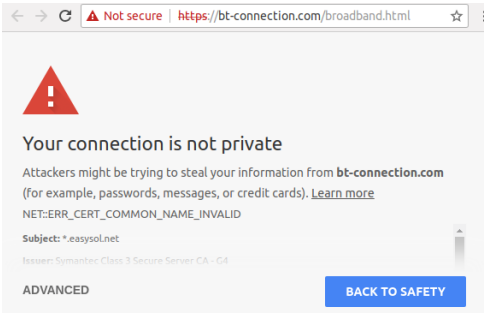


(a) Extended validation message.



(b) Domain validation and organization validation message.

(c) No validation message.



(d) Broken certificate message.

Figure 4: Chrome messages and icon by validation type.

Table 2: SSL and TLS certificate details.

Variable Name	Description
Fingerprint	Certificate SHA-1 hashed finger print
Serial Number	Certificate unique number for the issue.
Subject Principal	Subject's main field with details
Issuer Principal	Issuer's main field with details
Not Before	Date of creation
Not After	Expiration date
Extensions	Number of extensions contained in certificate
Version	Certificates SSL version number
Algorithm	Encryption algorithm used

by browser, operating systems and mobile vendors as trustworthy companies who are responsible for validating the information contained in a digital certificate. Web certificates have different validation levels, which means the authority that emits the certificate has validated organization information as real. The three different validation levels are shown in Table 1. Moreover, in Figure 4, it is shown how the Google Chrome browser displays certificates messages for different types of certificates.

Lastly, the information contained in a certificate is somewhat limited. In Table 2, the list of details contained in a certificate are shown.

2.2 Neural networks and deep learning

Artificial intelligence, deep neural networks and deep learning are some of the most promising modeling tools ever used. The current theory relies on neural networks models. Basically, we can define a standard neural network as a mathematical model made up of many simple, connected processing units called neurons, each connected with others units to produce a sequence of transformations to input data [35]. Based on neural network theory, deep neural networks are an extension of traditional theory that incorporate more complex and longer models in order to improve feature extraction and transformation of the inputted data to increase model performance [10].

In deep neural networks, there are special types of algorithms called recurrent neural networks (RNNs). RNNs are the set of neural networks that enhance and analyze data with sequential dependence as time series variables or text data. The main drawback is that RNN models are able to only learn the patterns of short dependencies, and not patterns in long-term dependencies. This means that if we were analyzing some time series data, a general RNN model will easily learn the patterns for the last day, but "forget" the general context of the last month. To mitigate this problem, S. Hochreiter and J. Schmidhuber [18] proposed a new model architecture called a long short-term memory network (LSTM). LSTM models enable effective learning processes of both short and long term dependencies by modifying the fundamental processing unit or neuron.

3 RELATED WORK

3.1 Malware detection

Traditional malware detection has been done either by manual methods or by analyzing the traffic payload using expert rules [30]. Unfortunately, those traditional methods cannot work with encrypted content. Recent work has focused on detecting malicious encrypted traffic by analyzing network connections in real time. This is done by investigating the encrypted malware communication with C2 servers, identifying the destination of such communication and then a DNS sinkhole is created to redirect the malware communication away from the C2 servers [27]. This represents a reactive approach because it must allow the malware to infect, propagate and execute its harmful action before it can be stopped. Furthermore, this approach needs to decrypt the communication in order to perform analysis of the malware's content [34].

Another approach is based on certificate and IP address pivoting to keep track of threat actor infrastructure. Classification strategy for this approach is done by the use of internet scanning and blacklisting of IP addresses and certificates, so when a new connection is coming from any blacklisted IP or uses a known malicious certificate, the connection is classified as malicious [3, 29].

As machine learning starts to become a more popular technique for encrypted traffic analysis, other work has shifted focus to connection metadata analysis. These approaches can predict when a connection is potentially harmful and keep track of threat actor infrastructure [3, 4].

Most recent work avoids the pivoting and starts with a focus only on certificates by looking at digital certificates data. For example, researchers from the security company Splunk were able to achieve a 91% accuracy by classifying certificates used in malware activities by using a support vector machines (SVM) algorithm [32].

3.2 Phishing detection

Phishing detection can be done by either proactive or reactive means. On the reactive end, we find services such as Google Safe Browsing API[1]. These types of services use a blacklist of malicious URLs to be queried. The blacklists are constructed using different techniques, including manual reporting, honeypots, or by crawling the web in search of known phishing characteristics [22, 36, 39, 40].

Proactive methods to detect phishing attacks involve analyzing the characteristics of a webpage in real time in order to assess the potential risk of that webpage. Risk assessment is done through a classification model [2]. Some of the machine learning methods that have been used to detect phishing include: support vector machines [21], streaming analytics [24], gradient boosting [25, 26], latent Dirichlet allocation [31], random forests [37], online incremental learning [23], and neural networks [28]. Several of these methods employ an array of website characteristics, which means that in order to evaluate a site, they first have to be analyzed before the algorithm can be used. This adds a significant amount of time to the evaluation process [6, 38]. Using URLs instead of content analysis reduces the evaluation time because only a limited portion of text is analyzed.

Lately, the application of machine learning techniques for URL classification has been gaining attention. Several studies proposing the use of classification algorithms to detect phishing URLs have come to the light in recent years [20, 26, 37]. These studies are mainly focused on creating features through expert knowledge and lexical analysis of the URL. The most recent approaches to detecting phishing websites are using deep neural networks to identify hidden patterns left by the attackers [7, 9, 33].

Prior works proved that both malware and phishing attackers leave clues behind in their certificate information that can be used to track them back to their infrastructure, and also classify the certificate as malicious or legitimate. However, no work shows whether the same sort of clues can be found and used to classify certificates for phishing.

4 REAL-TIME MALICIOUS CERTIFICATE DETECTION

Current browser strategies to validate web certificates are very simple and rely mostly on check if a certificate is self-signed and the expiration dates. However, domain validation certificates, which are the simplest validation type, still send a safeness message to the end user. Attackers usually use self-signed certificates as well as free generated certificates, because they are quick and cheap to generate. However, by using this sort of certificate, attackers expose their intentions, leaving them vulnerable to detection, tracking and blacklisting. To summarize, by detection of certificate abuse in real time, it is possible to close the time frame for attackers to get profit from attack, reducing their success rate.

In this section we describe our proposed algorithm to detect malicious web certificates using deep neural networks. First, we explain our process to extract useful information from a web certificate itself, and then we present how we use that information in a deep learning algorithm to detect malware and phishing certificates.

4.1 Feature engineering

From our analysis and feature creation process we extracted the most important indicators that could differentiate a malicious certificate from a legitimate certificate. We focused on what information is contained or implicit in a certificate to make it trustworthy, keeping in mind that certificates with less information are more suspicious.

It is expected that attackers will not spend time or money to buy and validate certificates as it may reduce their revenue and expose their intentions. Based on this assumption and insight from our company's SOC, we created features that show whether a certificate is likely to be legitimate. While performing our data analysis, we observed some patterns that lead us to create four categories of features. We noticed that malware and phishing certificates are almost always missing several fields of information. From this, we created a set of boolean features indicating the information contained in each certificate - this category is called *Boolean*. From

[1] https://safebrowsing.google.com/

Table 3: Features created.

Feature Name	Description	Category
SubjectCommonNameIp	Indicates if CN is an IP address instead of domain	Boolean
Is_extended_validated	Indicates if certificate is extended validated	Boolean
Is_organization_validated	Indicates if certificate is organization validated	Boolean
Is_domian_validated	Indicates certificate is domain validated	Boolean
SubjectHasOrganization	Indicates if subject principal has O field	Boolean
IssuerHasOrganization	Indicates if issuer principal has O field	Boolean
SubjectHasCompany	Indicates if subject principal has CO field	Boolean
IssuerHasCompany	Indicates if issuer principal has CO field	Boolean
SubjectHasState	Indicates if subject principal has ST field	Boolean
IssuerHasState	Indicates if issuer principal has ST field	Boolean
SubjectHasLocation	Indicates if subject principal has L field	Boolean
IssuerHasLocation	Indicates if issuer principal has L field	Boolean
Subject_onlyCN	Indicates if subject principal has only CN field	Boolean
Subject_is_com	Indicates if subject CN is a ".com" domain	Boolean
Issuer_is_com	Indicates if issuer CN is a ".com" domain	Boolean
HasSubjectCommonName	Indicates if CN is present in subject principal	Boolean
HasIssuerCommonName	Indicates if CN is present in issuer principal	Boolean
Subject_eq_Issuer	Boolean indicating if Subject Principal = Issuer Principal	Boolean
SubjectElements	Number of details present in subject principal	Splunk
IssuerElements	Number of details present in issuer principal	Splunk
SubjectLength	Number of characters of whole subject principal string	Splunk
IssuerLength	Number of characters of whole issuer principal string	Splunk
ExtensionNumber	Number of extensions contained in the certificate	Splunk
Selfsigned	Indicates if certificate is self signed	SOC
Is_free	Indicates if the certificate is free generated	SOC
DaysValidity	Calculated days between not before and not after days	SOC
Ranking_C	Calculated ranking of domain based on domain ranking	SOC
SubjectCommonName	Calculated character entropy in the subject CN	text
Euclidian_Subject_Subjects	Calculated euclidean distance of subject among all subjects	Text
Euclidian_Subject_English	Calculated euclidean distance of subject characters among English characters	Text
Euclidian_Issuer_Issuers	Calculated euclidean distance of issuer among all issuers	Text
Euclidian_Issuer_English	Calculated euclidean distance of issuer characters among English characters	Text
Ks_stats_Subject_Subjects	Kolmogorov-Smirnov statistics for subject in subjects	Text
Ks_stats_Subject_English	Kolmogorov-Smirnov statistic for subject in English characters	Text
Ks_stats_Issuer_Issuers	Kolmogorov-Smirnov statistics for issuers in issuers	Text
Ks_stats_Issuer_English	Kolmogorov-Smirnov statistic for issuer in English characters	Text
Kl_dist_Subject_Subjects	Kullback-Leiber Divergence for subject in subjects	Text
Kl_dist_Subject_English	Kullback-Leiber Divergence for subject in English characters	Text
Kl_dist_Issuer_Issuers	Kullback-Leiber Divergence for issuer in Issuers	Text
Kl_dist_Issuer_English	Kullback-Leiber Divergence for issuer in English characters	Text

our SOC agents' experience, we learned that a useful way to detect malicious intent is to check the domain ranking and differentiate when a certificate is self-signed and acquired for free, and to check the validity period. This category is called *SOC*. We also noticed that some information fields were repeated across the malware and phishing data set. From these, we created text analysis features that allow us to understand the issuer and subject information in a category we called *Text*. Finally we added the features used by Splunk [32], in a category called *Splunk*. In Table 3, the features are explained in detail.

4.2 Deep neural network

We utilized a deep neural network as the main algorithm to predict if a certificate is used for malicious purposes. The algorithm is composed of two different parts.

First we analyzed the text contained in the *subject principal* and *issuer principal* fields of the web certificate. To achieve this, the text contained in these features turned into a matrix representation by using *one hot encoding* technique based on the alphabet, such that X and Y take the form:

X features with shape $N \times S \times V$,

Y label with shape $N \times V$,

where N is the number of certificates analyzed, S is the maximum number of characters of each *issuer* and *subject* encoded and V the number of different characters in the vocabulary.

In this mathematical representation, each row represents a character and is filled with zeros except where the row matches a character in the alphabet. At the end, we calculate embedding and pass each feature as input to a LSTM layer that creates a vector size of 32, representing both the *subject principal* and the *issuer principal*.

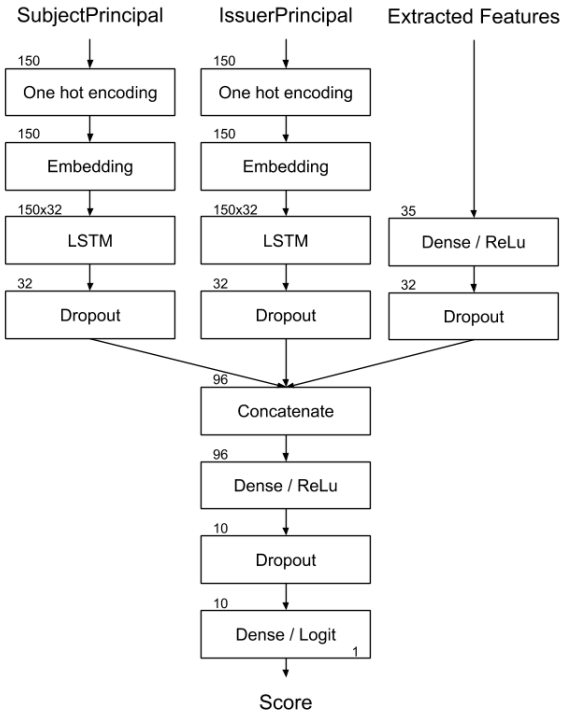


Figure 5: Neural network architecture to classify malicious certificates.

Second, we created an additional neural network that receives as an input the 35 features extracted from the web certificate, see Table 3, and returns a 32 vector representation of that data. Finally, the output of all three networks is concatenated into a vector of size 96, and additional layers of the neural network are added until the results are reduced to a single probability of being malicious. An architectural diagram of the network is shown in Figure 5.

5 EXPERIMENTAL SETUP

5.1 Data

To train our classification models, a dataset of legitimate, phishing and malware certificates is created. The phishing certificates come from Vaderetro[2] an internal feed that gave us confirmed phishing certificates. We also extracted malware certificates from abuse.ch[3] project and censys.io[4], they gave us blacklisted certificates and pem files. Finally, legitimate certificates came from Alexa top one million[5] rank who provided us with those website certificates. Our dataset has a total of 5,000 phishing certificates, 3,000 malware certificates and 1,000,000 legitimate certificates.

[2]https://isitphishing.org/
[3]https://sslbl.abuse.ch/
[4]https://censys.io/
[5]https://www.alexa.com/

Table 4: Most common CN found in certificates.

Malware		Phishing	
Domain Name	%	Domain Name	%
No CN	30.8%	incapsula.com	1.8%
example.com	8.5%	localhost	1.4%
localhost	6.0%	No CN	1.1%
domain.com	4.6%	Parallels Panel	0.7%
www.example.com	1.1%	localhost.localdomain	0.5%

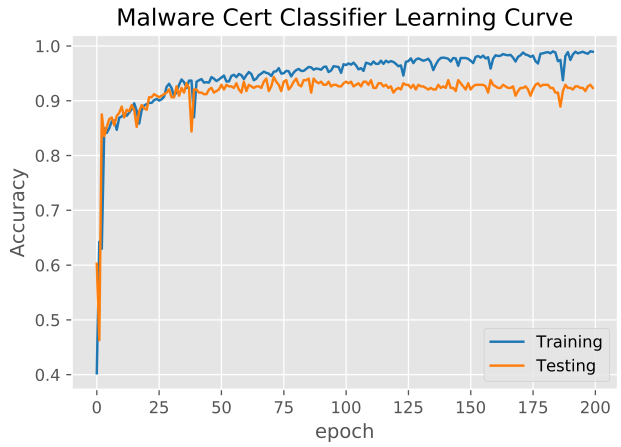
By looking deep into the data we noticed some obvious patterns. In cases of malware, we found the same certificate fingerprint coming from different IP addresses, or certificates with a CN like *example.com* or *www.example.com* and all of these certificates were self-signed. In cases of phishing, we found slightly different patterns, like the use of obviously fake CNs, but many of them also used incomplete location information, for example, Springfield with no state. To test our hypothesis about validation levels, we checked certificate validation levels and validity days to find out how many certificates of each source where validated or free. In Table 4, an example of most common CN found across certificates used for phishing activity, and certificates used for malware activity.

We then cleaned our dataset to make sure we had unique certificates. The cleaning process left 1,000,000 legitimate-use certificates, 900 malware certificates and 2,000 phishing certificates, all of which were unique. After cleaning the dataset we saw a dramatic reduction of malware and phishing because of the reused number of certificates, however, the same patterns were found after feature creation was maintained. As shown in Table 5, 44.3% of legitimate businesses use validated certificates, meanwhile just 10% and 9% of phishing and malware campaigns use validation, respectively. In particular, the two most strong validation types (EV and OV) are mainly used by legitimate businesses, because they imply higher cost and time to be obtained. The remarkable thing is that we found 0.01% fingerprints with extended validation and 0.6% with organization validation used for phishing attacks, which can be interpreted as phishing compromised domains.

5.2 Experimental design

From tabulated data we started creating features mentioned in Section 4.1. As our dataset was unbalanced, we created a random balanced dataset by selecting random legitimate certificates to match the phishing and malware certificates. In order to evaluate the performance of the models, we used a five-fold cross-validation strategy. This process consists of splitting the data in five folds, then training the data using four folds and the remaining one is used for validation of the model. This process is repeated five times, each time using a different fold for validation. In the end, all the performance metrics on the validation folds were averaged. In this way the variance could be analyzed and we could get a better estimate of the model's performance.

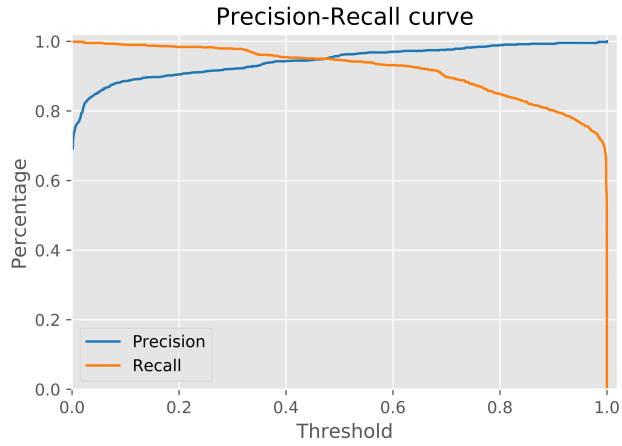
The evaluation of the performance is done using standard classification evaluation measures. Using a confusion matrix, the following measures are calculated: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, $Recall = \frac{TP}{TP+FN}$, $Precision = \frac{TP}{TP+FP}$ and $F_1\text{-}Score = 2\frac{P \cdot R}{P+R}$,



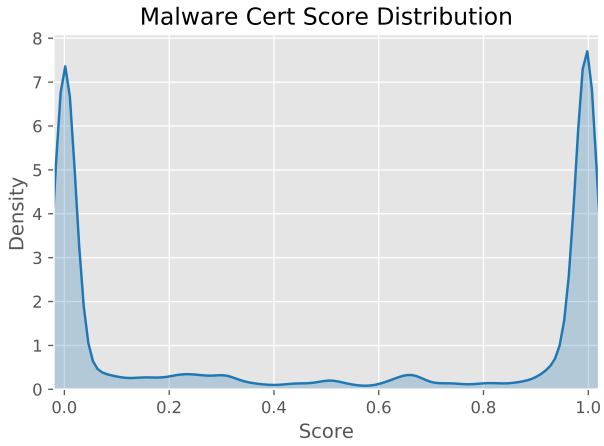
(a) Learning curve malware certificate classifier.

	Precision	Recall	Accuracy	F1-Score
count	5.0000	5.0000	5.0000	5.0000
mean	0.9501	0.9474	0.9487	0.9485
std	0.0092	0.0230	0.0090	0.0099
min	0.9402	0.9128	0.9402	0.9373
25%	0.9441	0.9368	0.9430	0.9422
50%	0.9477	0.9558	0.9460	0.9479
75%	0.9551	0.9602	0.9516	0.9521
max	0.9632	0.9714	0.9630	0.9632

(b) Results 5-fold cross-validation malware certificate classifier.



(c) Precision-Recall curve of the malware certificate classifier.



(d) Distribution of the malware certificate classifier score.

Figure 6: Results of the malware certificate deep neural network classifier.

Table 5: Validation certificates by certificate category.

	DV	OV	EV	No Validation
Legitimate	32.6%	4.0%	7.7%	55.7%
Phishing	9.0%	0.6%	0.01%	90.0%
Malware	9.7%	0.0%	0.0%	91.0%

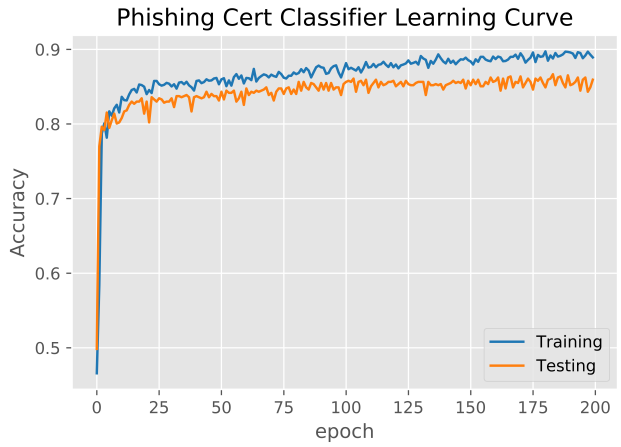
where P, R, TP, FN, TN and FP are the precision score, recall score, the numbers of true positives, false negatives, true negatives and false positives, respectively. We define as positive the malicious certificates and negative the legitimate/ham ones.

6 RESULTS

We used a database of more than a million legitimate web certificates, 900 certificates used for hosting malware, and 2,000 certificates used for phishing. We trained two different algorithms, one to detect malware certificates, and the other to detect phishing certificates, using a deep neural network as described in Section 4.2. To classify the certificates this model produces a score from 0 to 1, where 1 means a 100% probability of the certificate being used for malicious purposes, either malware or phishing. By taking a threshold equal to 0.5, we classified each certificate as malicious or not, according to its score.

6.1 Malware certificate detection

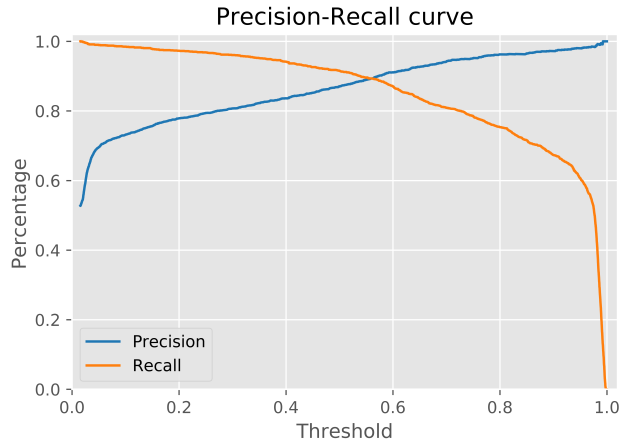
In our first experiment we selected the certificates used by malware and trained the deep neural network to identify its patterns. We evaluated the algorithm using a five-fold cross validation strategy. We allowed the algorithms to run for 200 epochs and after that, and we did not see any additional increase in performance, as can be seen in Figure 6a. Then we evaluated the performance of the algorithm in each fold. The results are shown in Table 6b. It was observed that the algorithm had an accuracy of a 94.87%, correctly



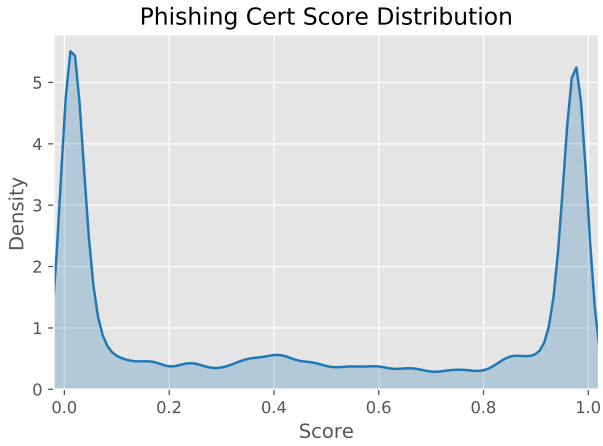
(a) Learning curve phishing certificate classifier.

	Precision	Recall	Accuracy	F1-Score
count	5.0000	5.0000	5.0000	5.0000
mean	0.8963	0.8748	0.8864	0.8852
std	0.0165	0.0188	0.0110	0.0082
min	0.8676	0.8512	0.8752	0.8760
25%	0.8979	0.8617	0.8796	0.8794
50%	0.9022	0.8780	0.8811	0.8831
75%	0.9060	0.8838	0.8957	0.8926
max	0.9077	0.8992	0.9003	0.8947

(b) Results 5-fold cross-validation phishing certificate classifier.



(c) Precision-Recall curve of the phishing certificate classifier.



(d) Distribution of the phishing certificate classifier score.

Figure 7: Results of the phishing certificate deep neural network classifier.

identifying 94.74% of the malware certificates, with a precision of 95.01%. Furthermore, the results showed a very stable model, as the variation of the accuracy is less than one percent across the five folds.

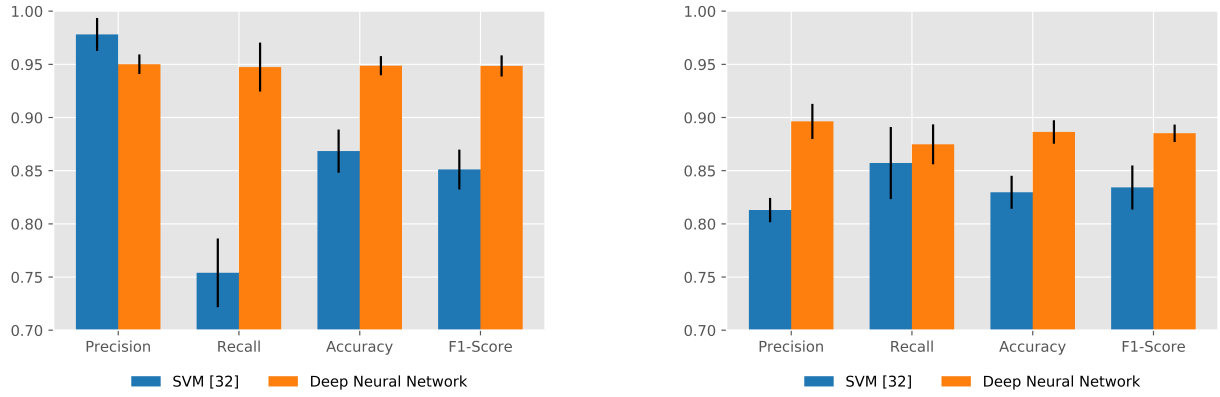
The precision recall curve is shown in Figure 6c. We can see that the algorithm could be used in different scenarios, such as using a threshold of 0.1 the algorithm could detect over 99% of the malware certificates without having that many false positives. Furthermore, the distribution of the scores are presented in Figure 6d, we can see that most of certificates are classified either with high or low probabilities.

6.2 Phishing certificate detection

We trained the algorithms using the phishing certificates database. We defined 200 epochs to optimize the parameters of the deep neural network. In Figure 7a we see the learning curve of the algorithm. Although it may be suggested that adding additional epochs may increase the performance, the gap between training and testing errors started to grow a lot, so we decided to stop on those iterations.

Afterwards, we evaluated the performance of the algorithms. The results are shown in Table 7b. The deep neural network had an accuracy of 88.64% with a standard deviation of 1.1%. This tells us that the performance of the algorithm its very stable. Moreover, the algorithm detected 87.48% of malicious phishing certificates with a precision of 89.63%.

The precision recall curve is shown in Figure 7c. This curve allowed us to make decisions regarding which score threshold to use in order to manage the false positives or false negatives. In particular, the model could detect almost 94% of malicious certificates while having a precision of 80% using a threshold equal to 0.15. Figure 7d shows the score distribution.



(a) Malware classifier.

(b) Phishing classifier.

Figure 8: Comparison of algorithms performance using support vector machines and deep neural networks.

6.3 Comparison with previous models

Lastly, we compared the results of the deep neural networks to those obtained using Splunk's support vector machines algorithm. The main difference between these algorithms is that the deep neural network is able to use the text information contained in the certificate *subject* and *issuer* in a more effective way than traditional machine algorithms, like SVM, due to its long short term memory layers.

In Figure 8, we can compare the results of Splunk's SVM model to those of our deep neural network algorithm for both malware and phishing certificates. In the malware certificates algorithm, the deep neural network outperformed the support vector machines model by more than 7% in accuracy. Similarly, in the phishing certificates model, the deep neural networks outperformed Splunk's model by more than 5% in accuracy. Furthermore, the improvements of the results are also observed in the recall, precision, and F1-Score statistics.

7 CONCLUSIONS

We used a database of 1,000,000 legitimate, 5,000 phishing and 3,000 malware certificates obtained by crawling the internet. Using our company's SOC experience and a deep analysis of certificate information, we created more than 30 features. We also proposed a novel approach using deep learning algorithms and recurrent neural networks to more accurately detect malware and phishing web certificates. Using these algorithms, we improved the feature engineering process by allowing the model to automatically uncover the hidden patterns in the malicious web certificates. This new proposed algorithm is able to leverage detection by more effectively analyzing text data, in addition to the other features we included. Using a deep learning model, we were able to outperform Splunk's results. In the case of malware, our model achieved an accuracy of 94.87%, a 7% improvement over Splunk's results. In the case of phishing, our model achieved 88.64% accuracy, a 5% improvement over Splunk. The high success rate of the classification in both cases demonstrates the strength of the proposed model.

The accuracy difference between both phishing and malware results was due to the certificates used for phishing attacks having details very similar to ones used for legitimate businesses. For example, certificates with fake data still had country, common name and location details. This is because phishers tend to use compromised domains, meaning that legitimate business certificates are sometimes taken and used for phishing attacks.

This paper shows an alternative approach to previous results in latest literature. This approach is able to outperform those results by introducing a more detailed feature engineering process, coupled with the use of deep neural networks to leverage certificate text data. We expect the algorithm to be used widely within the cybersecurity community, as cybercriminals are increasingly relying on using encrypted communication. Capturing data hidden in digital certificates should help in the fight against threat actors.

REFERENCES

[1] 2018. *Encrypted Traffic Analytics (White Paper).* Technical Report. Cisco Systems.
[2] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. 2007. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit on - eCrime '07.* 60–69. https://doi.org/10.1145/1299015.1299021
[3] Blake Anderson and David McGrew. 2016. Identifying Encrypted Malware Traffic with Contextual Flow Data. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (AISec '16).* ACM, New York, NY, USA, 35–46. https://doi.org/10.1145/2996758.2996768
[4] Blake Anderson and David McGrew. 2017. Machine Learning for Encrypted Malware Traffic Classification: Accounting for Noisy Labels and Non-Stationarity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17).* ACM, New York, NY, USA, 1723–1732. https://doi.org/10.1145/3097983.3098163
[5] James Antonakos and Anat Davidi. 2017. *2017 Trustwave Global Security Report.* Technical Report. Trustwave.
[6] Calvin Ardi and John Heidemann. 2015. *Poster: Lightweight Content-based Phishing Detection.* Technical Report ISI-TR-2015-698. USC/Information Sciences Institute. http://www.isi.edu/%7ejohnh/PAPERS/Ardi15a.html
[7] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González. 2017. Classifying phishing URLs using recurrent neural networks. In *2017 APWG Symposium on Electronic Crime Research (eCrime).* 1–8. https://doi.org/10.1109/ECRIME.2017.7945048
[8] Chris Bailey. 2018. The Crisis from Encrypted Phishing Sites. In *2018 APWG Symposium on Electronic Crime Research (eCrime).* IEEE.

[9] Alejandro Correa Bahnsen, Ivan Torroledo, David Camacho, and Sergio Villegas. 2018. DeepPhish: Simulating Malicious AI. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 1–8.

[10] Li Deng and Dong Yu. 2014. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* 7, 3-4 (June 2014), 197–387. https://doi.org/10.1561/2000000039

[11] Internet Engineering Task Force. 2011. The Secure Sockets Layer (SSL) Protocol Version 3.0. Retrieved June 18, 2018 from https://tools.ietf.org/html/rfc6101

[12] Internet Engineering Task Force. 2016. Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS). Retrieved June 18, 2018 from https://tools.ietf.org/html/rfc7919

[13] Network Working Group. 2003. Public-Key Cryptography Standards (PKCS) 1: RSA Cryptography Specifications Version 2.1. Retrieved June 18, 2018 from https://tools.ietf.org/html/rfc3447

[14] Network Working Group. 2008. The Transport Layer Security (TLS) Protocol Version 1.2. Retrieved June 18, 2018 from https://tools.ietf.org/html/rfc5246

[15] Network Working Group. 2010. Transport Layer Security (TLS) Authorization Extensions. Retrieved June 18, 2018 from https://tools.ietf.org/html/rfc5878

[16] Network Working Group. 2018. The Transport Layer Security (TLS) Protocol Version 1.3 draft-ietf-tls-tls13-28. Retrieved June 18, 2018 from https://tools.ietf.org/html/draft-ietf-tls-tls13-28

[17] Crane Hassold. 2017. A Quarter of Phishing Attacks are Now Hosted on HTTPS Domains: Why? https://info.phishlabs.com/blog/quarter-phishing-attacks-hosted-https-domains. [Online; accessed 19-June-2017].

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[19] Jeremy D'Hoinn Lawrence Orans, Adam Hils and Eric Ahlm. 2017. *Predicts 2017: Network and Gateway Security.* Technical Report. Gartner.

[20] Anh Le, Athina Markopoulou, and Michalis Faloutsos. 2011. PhishDef: URL Names Say It All. In *INFOCOM, 2011 Proceedings IEEE*. https://doi.org/10.1109/INFCOM.2011.5934995 arXiv:1009.2275

[21] Gaston L'Huillier, Alejandro Hevia, Richard Weber, and Sebastian Rios. 2010. Latent semantic analysis and keyword extraction for phishing classification. In *International Conference on Intelligence and Security Informatics*. 129–131.

[22] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. ACM, New York, NY, USA, 1245–1254. https://doi.org/10.1145/1557019.1557153

[23] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. Identifying Suspicious URLs: An Application of Large-scale Online Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 681–688. https://doi.org/10.1145/1553374.1553462

[24] Samuel Marchal, Jerome Francois, Radu State, and Thomas Engel. 2014. PhishStorm: Detecting Phishing With Streaming Analytics. *IEEE Transactions on Network and Service Management* 11, 4 (2014), 458–471. https://doi.org/10.1109/TNSM.2014.2377295

[25] Samuel Marchal, Kalle Saari, Nidhi Singh, and N. Asokan. 2015. Know Your Phish: Novel Techniques for Detecting Phishing Sites and their Targets. (oct 2015). arXiv:1510.06501 http://arxiv.org/abs/1510.06501

[26] Samuel Marchal, Kalle Saari, Nidhi Singh, and N. Asokan. 2016. Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets. In *International Conference on Distributed Computing Systems*. 323–333. https://doi.org/10.1109/ICDCS.2016.10 arXiv:1510.06501

[27] D. McGrew and B. Anderson. 2016. Enhanced telemetry for encrypted threat analytics. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. 1–6. https://doi.org/10.1109/ICNP.2016.7785325

[28] Rami M. Mohammad, Fadi Thabtah, and Lee McCluskey. 2014. Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications* 25, 2 (01 Aug 2014), 443–458. https://doi.org/10.1007/s00521-013-1490-z

[29] Mark Parson. 2017. Using TLS certificates to track activity groups. Retrieved June 18, 2018 from https://www.slideshare.net/MSbluehat/bluehat-v17-using-tls-certificates-to-track-activity-groups

[30] M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui. 2010. Network Traffic Analysis and Intrusion Detection Using Packet Sniffer. In *2010 Second International Conference on Communication Software and Networks*. 313–317. https://doi.org/10.1109/ICCSN.2010.104

[31] Venkatesh Ramanathan and Harry Wechsler. 2013. Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation. *Computers & Security* 34 (2013), 123–139. https://doi.org/10.1016/j.cose.2012.12.002

[32] Dave Herrald Ryan Kovar. 2018. The "Hidden Empires" of Malware. Retrieved June 18, 2018 from https://www.sans.org/summit-archives/file/summit-archive-1517253771.pdf

[33] Joshua Saxe and Konstantin Berlin. 2017. eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys. *CoRR* abs/1702.08568 (2017). arXiv:1702.08568 http://arxiv.org/abs/1702.08568

[34] Sourabh Saxena. 2016. Demystifying Malware Traffic. (August 2016), 1735–80.

[35] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85 – 117. https://doi.org/10.1016/j.neunet.2014.09.003

[36] Javier Vargas, Alejandro Correa Bahnsen, Sergio Villegas, and Daniel Ingevaldson. 2016. Knowing your enemies: Leveraging data analysis to expose phishing patterns against a major US financial institution. In *2016 APWG Symposium on Electronic Crime Research (eCrime)*. 52–61. https://doi.org/10.1109/ECRIME.2016.7487942

[37] Rakesh Verma and Keith Dyer. 2015. On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (CODASPY '15)*. ACM, New York, NY, USA, 111–122. https://doi.org/10.1145/2699026.2699115

[38] Ge Wang, He Liu, Sebastian Becerra, Kai Wang, Serge Belongie, Hovav Shacham, and Stefan Savage. 2011. *Verilogo: Proactive Phishing Detection via Logo Recognition.* Technical Report CS2011-0969. UC San Diego.

[39] Colin Whittaker, Brian Ryner, and Marria Nazif. 2010. Large-Scale Automatic Classification of Phishing Pages. In *NDSS '10*. http://www.isoc.org/isoc/conferences/ndss/10/pdf/08.pdf

[40] Jian Zhang, Phillip Porras, and Johannes Ullrich. 2008. Highly Predictive Blacklisting. In *17th USENIX Security Symposium*. 107–122. http://www.usenix.org/event/sec08/tech/full{_}papers/zhang/zhang{_}html/