31 May 2019 Study · Web & Cloud · Digital Identity · 9 min read

Detecting Certificate Misissue via CT Logs

Archived. This content is no longer updated. It remains available for reference.

Source https://technology2.a-sit.at/en/publications/detecting-certificate-misissue-via-ct-logs/

Certificate Transparency (CT) is an open framework that provides visibility of newly issued SSL/TLS certificates by enforcing Certificate Authorities (CAs) to log every certificate they issue in public, tamper-proof, append-only logs. This project aimed at exploring the viability of using CT Log entries as the sole data source to detect phishing websites certificates. The implemented system analyses certificates submitted to the Logs to build a machine learning-based classifier that predicts the phishing likelihood of newly issued certificates. The system uses features directly extracted from CT log data to successfully classify certificates into one of five different incremental certificate risk labels that range from legitimate to highly suspicious. Evaluation results demonstrate the effectiveness of the approach, with a success rate of over 90%. Results confirmed that CT is indeed a valuable source of data that can be machine-processed to mount automated alert systems. By relying solely on CT Log data, the system can deliver results in almost real-time, significantly reducing the time to detect phishing websites. The project resulted in a scientific paper accepted at the 15th EAI International Conference on Security and Privacy in Communication Networks.

1. Introduction

The web's current PKI system allows *any* trusted CA, or intermediate CA, to issue certificates for *any* subject identity. This assumption of trustworthy CAs introduces a vulnerability to attacks based on improperly issued certificates, either as a result of CA compromise, negligence, errors, or even malicious behavior. A prominent example of such a security incident is the so-called Operation Black Tulip incident with a Dutch CA named DigiNotar [2].

The green padlock symbol shown in browsers' address bars gives users a false sense of security regarding the trustworthiness of a website that employs TLS. Moreover, the popularization of free and automated TLS certificates by companies like Let's Encrypt and Cloudflare has led to a massive surge in the use of automatically-issued certificates on phishing sites (up to 49% in the third qarter of 2018 according to this PhishLab report [3]). Furthermore, when such a certificate misissuance happens (malicious or otherwise), it can take weeks or even months until the suspect certificates are detected and revoked. This window of vulnerability gives malicious actors plenty of time to do damage.

Google responded to the need of an easy and effective way to audit or monitor TLS certificates and CA operations in real-time by implementing Certificate Transparency (CT). CT is a system that publicly records ("logs") TLS certificates in centralized lists as they are issued or observed, in a manner that allows anyone to audit a certificate authority's activity and notice the issuance of suspect certificates for the domains they own.

The instant visibility of newly issued certificates can significantly reduce the amount of time needed until a malicious site or CA misconduct can be detected and proper mitigation actions are taken. In this project, we implement a system that detects phishing websites in almost real time by leveraging only CT Logs.

1.1. Certificate Transparency

The strength of the CT framework stems from the append-only, cryptographically assured nature of the logs. On a technical level, this is accomplished by relying on a Merkle Tree (i.e. a data structure made up of linked cryptographic hashes). This ensures that back-dated certificates cannot be inserted into the log, and added certificates cannot be edited or deleted afterwards.

A typical scenario of issuing and then monitoring/auditing a CT-logged certificate is shown in Figure 1:

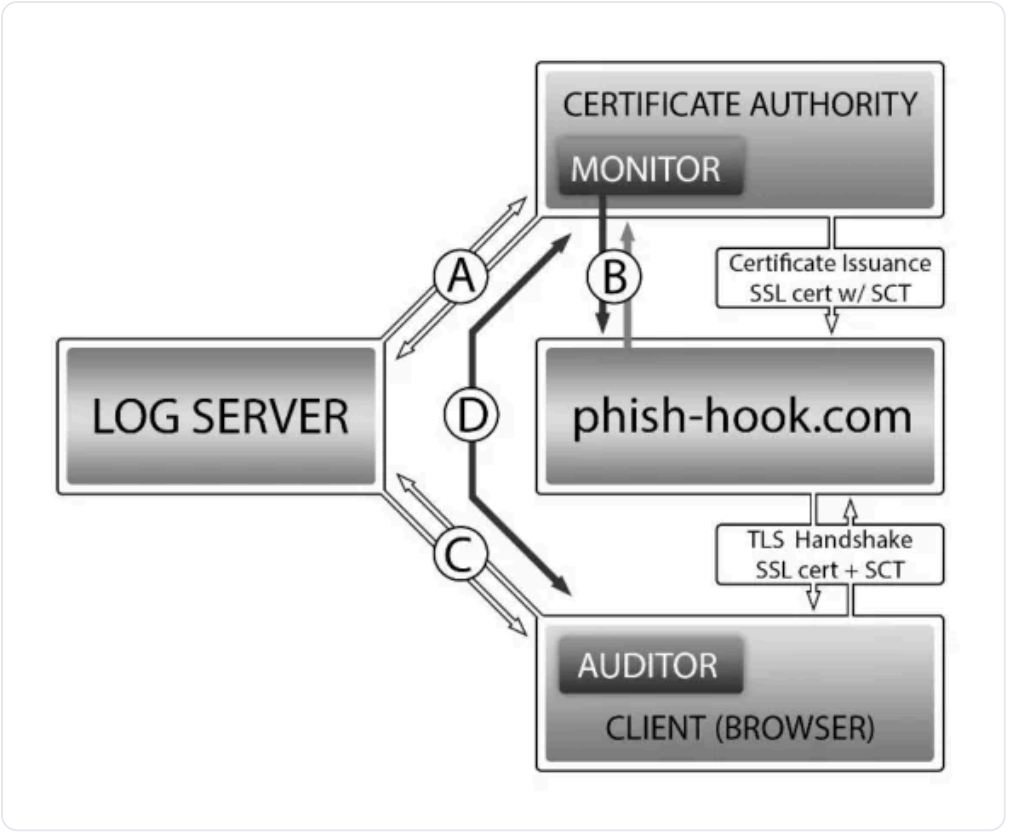


Figure 1: CT Operation Typical Scenario Upon receiving a certificate signing request (CSR) for the domain phish-hook.com, the CA prepares and submits a pre-certificate for this domain to the CT system. The CA then issues the certificate together with the SCT returned from one of the CT Log Servers. The server hosting the website phish-hook.com then delivers both the certificate and the SCT during the TLS handshake. Afterward, Monitors periodically check the logs for consistency and suspicious certificate issuance (A). The domain owner of phishhook.com queries Monitors for potentially malicious certificates submitted for their domain (B). Browsers make use of Auditors to verify that a certain certificate has been registered in the CT Logs (C). Monitors and Auditors share information to ensure the proper behavior of the logs (D). More details on how CT works are explained in [4].

1.2. Phishing Attacks

Malicious actors use several ways to trick users into believing they are visiting a website with a domain similar to one of the legitimate domains. Examples include typo-squatting attacks, homoglyph (name spoofing) attacks, or incorporating a legitimate domain as a prefix, inner part, or suffix of the new domain.

Typosquatting attacks aim at modifying the domain names by incorrectly spelling them, while homoglyphic attacks rely on character substitution using look-alike glyphs from the Unicode sets to create fake domain names that are nearly indistinguishable from real ones to the naked eye. A quick look at the confusables file [5] published by the Unicode Consortium, reveals that just for the character i in phish-hook, there exist up to 41 look-alike glyphs that can be utilized by attackers to produce misleading domain names. Also, domains can be built by incorporating the legitimate domain name into a longer domain.

Table 1 provides some examples for each of these techniques:

Table 1 Phishing Attacks Examples	**Col2**
Legitimate domain	`phish-hook.com`
Typosquatting attack	`phihs-hook.com, phish-hok.com`
Homoglyph attack	`phish-hook.com, phish-hook.com, phısh-hook.com.`
Prefix, Suffix	`www-phish-hook.com, login-phish-hook.com, www.phish-hook.com.malicious.fakedomain.name`

2. Implementation

In this project, we implemented a machine-learning-based solution for detecting phishing website certificates from CT Log entries. This phishing detection system is composed of three main components:

- Certificate Collector
- Feature Extractor
- Classifier

The CT Logs feed the Certificate Collector, which in turn passes the parsed CT Logs to the Feature Extractor component. The set of attributes generated from the Feature Extractor is finally used to train our Classifier model. New certificates streamed from the CT logs are then fed into the trained Classifier to be classified into phishing or legitimate. Figure 2 illustrates Phish-Hook's main components.

Table 2 Feature Set	Col2
Feature Name	Definition
small_lavenshtein_distance	Calculates a measure of similarity between two strings — of sub-words of the domain registered with the certificate to suspicious popular keywords (for example`phish-hook` vs.`phish_hook`)
deeply_nested_subdomains	Checks for domain names with unusually long subdomains such as`www.phish- hook.com.security.account-update.gq`
issued_from_free_CA	Checks for certificates obtained from free CAs as a potential indicator of suspiciousness
suspicious_tld	Checks for the presence of top-level domains mostly targeted by attackers in their attempt to create malicious sites

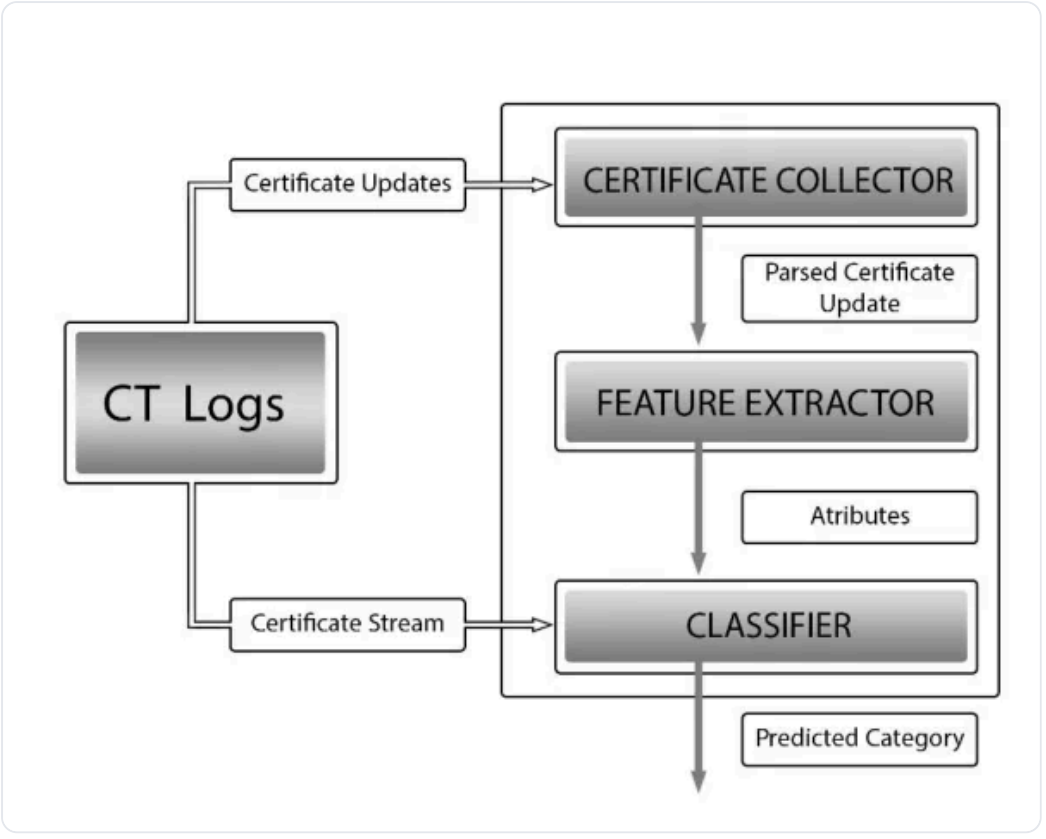


Figure 2: Phishing Detection System Components To build our own dataset, we used the CertStream open-source library to interact with the CT network and aggregate CT Log data. The set of features is directly extracted from the Log entries without requiring to download or analyze the respective certificates or website source code. These features were derived based on some of the most common techniques used for phishing. Table 2 summarizes the set of features and their definition:

inner_tld_in_subdomain	**Checks for the presence of a popular TLD in an inner sub-domain as an indicator of suspiciousness**
suspicious_keywords	Checks for the inclusion of popular keywords from famous applications of social media, commerce, or cryptocurrency in a domain name
high_shannon_entropy	Measures the degree of randomness— of the domain a certificate was issued for, targeting the detection of algorithmically generated malicious domains
hyphens_in_subdomain	Checks for the presence of multiple hyphens ('-') in the sub-domain, as both of these characters can be used to attach popular keywords of legitimate domains to generate malicious ones
phishing_likelihood_category	The resulting label, with the possible values of *legitimate, potential, likely, suspicious, and highly-suspicious*

In a nutshell, our system works as follows: We stream certificate updates from the Logs, while simultaneously labeling the data for each feature. We also employ a heuristic methodology to compute a total phishing likelihood score according to the presence or absence of a feature, or the respective computed value of a feature. We use this overall score to classify the certificate and assign the resulting feature called phishing_likelihood_category out of five different categories, namely *legitimate, potential, likely, suspicious, and highly-suspicious.*

3. Results

To evaluate the performance of Phish-Hook, we made use of the pre-classified phishing detection dataset publicly available under the UCI Machine Learning Repository [6]. This dataset consists of 11055 data points with 30 features. Part of the features corresponds directly to X.509 certificate fields, while others are derived certificates fields and website source code. Each feature takes a ternary value of [-1,0,1] representing phishing suspicious, and legitimate respectively. Unlike features, result labels can take only two values: phishing and legitimate. We model our small set of features as a subset of this set and thus make use of the pre-classified data to provide ground truth.

The following section presents the classification performance of Phish-Hook based on different classifiers. Table 3 reports the performance of classification algorithms such as k-nearest neighbor (KNN), support vector machine (SVM), decision tree classifiers (DT), multilayer perceptrons (MLP) against each other. We report accuracy, precision, recall, and F1 scores regarding the classification of phishing websites for each approach.

Table 3 Evaluation of Classification Models

Algorithm	Parameters	Accuracy	Precision	Recall	F1 score
DT	max_depth=2	91.06	91.12	91.06	91.07
DT	max_depth=5	91.42	91.46	91.42	91.39
DT	max_depth=10	89.39	89.44	89.39	89.40
SVM	kernel = 'linear' C = 0.03	91.62	91.68	91.62	91.58
SVM	kernel = 'linear' C = 0.3	91.29	91.37	91.29	91.25
SVM	kernel = 'linear' C = 1067	91.39	91.45	91.39	91.35
KNN	k=1	86.41	86.40	86.41	86.37
KNN	k=3	86.38	86.40	86.38	86.32
KNN	k=10	87.23	87.23	87.23	87.19
MLP	network_size=3×5	90.08	90.16	90.08	90.03
Col1	**network_size=5×10**	**89.55**	**89.94**	**89.55**	**89.44**
	network_size=1×100	89.06	89.63	89.06	88.92

Evaluation results are reported for various parameters tuned for each algorithm, such as maximum depth for DT, network size for MLP, and the penalty parameter C in SVM. The results demonstrate the effectiveness of our approach, with an accuracy of over 90%, while maintaining precision, recall, and F1 scores of also over 90%. Support vector classifiers (SVM) outperform other classifiers for the certificate classification task, closely followed by Decision Trees (DT) with a small difference margin.

Table 3 lists some examples of the suspicious certificates that were detected by our system:

All in all, evaluation results show that Phish-Hook can reliably classify phishing websites based solely on CT Log data in near real-time as they appear. This can significantly reduce the time it takes to detect phishing websites and consequently mitigate their impact.

2. References

[1] [Online]. Available: https://www.certificate-transparency.org/.

[2] [Online]. Available: https://www.rijksoverheid.nl/ministeries/ministerie-van-binnenlandse zaken-en-koninkrijksrelaties/documenten/rapporten/2011/09/05/diginotar-public-reportversion-1.

[3] [Online]. Available: https://info.phishlabs.com/blog/49-percent-of-phishing-sites-now-use-https.

[4] [Online]. Available: https://www.certificate-transparency.org/how-ct-works.

[5] [Online]. Available: https://unicode.org/cldr/utility/confusables.jsp.

[6] [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Website+Phishing.

```
secure.support.apple.com-orderpaymentsrefund.net
facebook.agus.web.id
groundmovies.video.youtube.free.watchanddownload.putlockers1.pw
management.centralus.control.database.windows.net
instagramkasma.com
apple.customer-support.org
thaiclouds.com
waws-prod-hk1-79cefe2a-api.p.azurewebsites.windows.net
webmail.instagrami.com.tr.ht
secure.support.apple.com-orderpaymentsrefundid.net
portal-ssl1973-2.bmix-lon-yp-5f5bc08d-ecdc-48dc-b091-29cf515f44c9.cm-
drugstars-com.composedb.com
secure.payment.appleid.payment-3dsecure.tk
sign.secure.myaccount.webaps.update-information.lockneon7212.com
autodiscover.blockchainforfinancialservices.com
www.netflix-support.factway.com.sa
portal-ssl1776-3.bmix-lon-yp-5f5bc08d-ecdc-48dc-b091-29cf515f44c9.cm-
drugstars-com.composedb.com
orders-aliexpress.roxxbg.com
appleid.apple.service-accountinformation-helpcenter404.com
*.aliexpress-shop.ga
www.linkedintube.localtubenetwork.com
twittertipscentral.com
andex-direct-nastrojka-kontekstnoj-reklamy.starobogatov.ru
www.paypal.counterpanehandmade.com
myetherwallet.com.verifysignature.mewlink.online
*.login.microsoftonline-p.comwebshell.suite.office.com.us.cas.ms
www.gmaillogin.review
```