Exploring the Security of Certificate Transparency in the Wild

Bingyu Li[1], Fengjun Li[2], Ziqiang Ma[3]

[1] School of Cyber Science and Technology, Beihang University, China
[2] Department of Electrical Engineering and Computer Science, the University of Kansas, USA
[3] School of Information Engineering, Ningxia University, China
libingyu@buaa.edu.cn, fli@ku.edu, maziqiang@nxu.edu.cn

Abstract. Certificate Transparency (CT) is proposed to detect fraudulent certificates and improve the accountability of CAs. CT as an open auditing and monitoring system is based on the idea that all CA-issued certificates are logged in a publicly accessible log server, and that CT-compliant browsers only accept publicly recorded certificates. The purpose of CT is to make all TLS server certificates issued by the CA publicly visible; once a fraudulent certificate is publicly published, it can be discovered by the domain name owner. In practical, the CT can achieve its intended purpose only when the three components of the CT cooperate and work correctly and effectively. In this paper, we study the interaction of the CT framework among log servers, monitors, auditors, CAs, domain owners (or websites), and browsers, and then analyze the security impact of each component on other components of the CT framework. Compared with traditional PKI systems, the CT framework does not rely on a single trusted party, but as a distributed system that distributes trust guarantees to many CAs, Log Servers, Auditors, and Monitors. We explore the security of CT framework in practice from multiple perspectives, and find that each component has many security vulnerabilities. Thus, the attackers might first exploit the vulnerability to disable the CT and then launch an attack using fraudulent certificates. The overall security guarantees of CT are jeopardized due to the weak protections of any components.

Keywords: Certificate Transparency (CT) · Fraudulent Certificate · Trust Management

1 Introduction

Public Key Infrastructure (PKI) uses certificates to establish and transmit trust in the Internet [8]. The Certificate Authority (CA) is responsible for issuing a certificate, which is used for blinding users' identity and public key. So it is usually assumed that the CA is completely reliable. The browser credibly obtains the public key of the website via certificates used for authentication, confidentiality, data integrity and non-repudiation etc. Therefore, it must ensure

the validity of certificates when deploying PKI system in information system. Otherwise, it cannot find the fraudulent certificate, even though the browser strictly validates the certificates. By 2020, there are more than 2.3 billion valid certificates in the Internet [38].

However, in recent years, a series of security incidents [7, 12, 16, 22, 29, 40–42] have shown that accredited CAs may issue fraudulent certificates due to administrative oversight or attacks. The attacker uses fraudulent certificates to bind the key pair to the domain name that does not belong to him. Thus, it can launch malicious websites, MitM/impersonation attacks without any warning against targets such as mainstream websites, national core devices or user networks. Numerous fraudulent certificates weaken the trust provided by PKI system and result in serious threat to the application and promotion of PKI.

Traditional PKI system lacks the mechanism of finding fraudulent certificate. The fraudulent certificate usually takes a long time to be detected (from weeks to months). In addition, browsers' trust in accredited CAs are undifferentiated, and any of the CA's security problems may harm the entire Internet ecosystem. Therefore, the attack surface of fraudulent certificates on the network is long-term and extensive.

Aiming at the security threat caused by fraudulent certificates, the Certificate Transparency (CT) scheme [24] is proposed to timely detect the fraudulent certificates and enhance the accountability of CAs. CT as an open auditing and monitoring system, the basic idea is to record all certificates issued by the CA in a publicly accessible log server, and clients (e.g., browsers) only accept publicly issued certificates. Its purpose is to make all TLS server certificates issued by the CA publicly visible and subject to public monitoring and auditing. Once a fraudulent certificate is published via CT, it can be detected by the domain owner. Therefore, CT introduces the following three new components: (a) Log server, used to record certificates submitted by the CA or domain owner, etc.; (b) Auditor, verify that the log server behavior is correct; (c) Monitor, obtain all certificates recorded in the log regularly to help find suspicious (fraudulent) certificates.

In recent years, it has become a consensus to introduce the CT mechanism into PKI system in the industry. CT has been supported by CAs, websites, browsers and TLS software, including Chrome [19], Apple platforms [3], Mozilla Firefox/NSS [30], OpenSSL [33], Nginx [31], Microsoft AD Certificate Service and Azure Key Vault [28].

The purpose of CT scheme is to quickly detect fraudulent certificates. Compared with the traditional PKI system, the CT framework does not rely on a single trusted party, but as a distributed system, it distributes trust security to CAs, log servers, Auditors and Monitors [11, 23, 24]. The CT log is only responsible for recoding the valid certificates submitted by CAs and issuing SCTs, and does not check whether the certificate is authorized by the domain owner. A certificate is submitted and recorded in multiple logs based on the browser's CT policy to obtain multiple SCT. When the browser establishes a TLS handshake

with the website, it will check the SCTs; only certificates that meet the CT policy and are issued by a trusted CA will be accepted.

The fraudulent certificate issued by trusted CA can also be verified by browser after being submitted to CT log and obtaining SCTs. Therefore, CT itself can not prevent CA from issuing fraudulent certificate. Instead, it relies on Monitor to regularly obtain and check all the certificate recorded in the log to help detect fraudulent certificates. Any stakeholder (i.e., domain owner or trusted third party) can act as monitor. In addition, through SCTs and STHs, Auditor regularly checks whether CT Log meets to consistency and existence, realizes the behavior audit of the log, ensures the log to run correctly, and always provides the real and effective certificate records.

In an ideal state, CT components and each links achieve the security via redundant and digital signature [24,25]. First of all, Log server guarantees append-only based on Merkle hash tree. Log server, CA and domain owner depend on the digital signatures of certificate and SCTs, and the public keys of the signers are publicly known or pre-installed in the verifiers. Secondly, the behavior consistency of Log server is audited by Auditor and Monitor. The interaction security between Log server and other components, including browser, Auditor and Monitor, is designed with the fault tolerance of redundant auditors. These interactions also rely on digital signatures, including SCT or/and STH signatures. And the public key used to verify signature is publicly known. Auditor and Monitor provide security services to browsers and domain owners through mutual interaction and redundant to help detect fraudulent certificates or incorrect behavior of the log servers. In summary, among these components, the public keys of the signers are publicly known and it is assumed that at least one of the numerous Auditors and Monitors is secure and reliable. Therefore, they will seldom suffer from MitM attack exploiting fraudulent certificate.

In practice, only the three components of CT work correctly and effectively, the CT can achieve the expected goal. For example, if the log server does not append the certificate to the public log within the maximal merge delay (MMD), and Auditor fails to detect the incorrect behavior of the log in time. Alternatively, Monitor may not reliably detect fraudulent certificates from the log server in a timely manner, etc. These reasons may be lead to the attacker could exploit fraudulent certificate to launch MitM or impersonation attacks, without triggering any alert in CT. The longer the fraudulent certificates stay undetected in the system (or CT logs), the more the damage they may cause to the PKI ecosystem. Therefore, these factors such as the correctness of CT log behavior, the quality of certificate monitoring server provided by Monitor, and the granularity and timeliness of audit log, will affect the overall security enhancement by the CT framework.

In this paper, we investigate the security configuration of the components in the CT framework and their interaction in practical. After comprehensively analyzing the security configurations of these components, we find that, compared with the security design, these components are not significantly more immune to the security vulnerabilities. Therefore, the attacker could first launch MitM

and/or DDoS attacks on one or more of the CT components to manipulate the certificate monitor results or audit results, or to invalidate the CT mechanism. Then, when a fraudulent certificate was exploited in the MitM attacks on any ordinary website which supports CT, the domain owner still can not detect this fraudulent certificate because the attackers would conceal the certificate in the manipulated search result, or force the browser not to perform CT policy checks. Note that, in this attack scenario as explained above, the log server, monitor and auditor have malicious behavior due to their own vulnerabilities or defects, and the domain owner and browser will accept the fraudulent certificate without receiving any warning from the CT mechanism.

Contribution. We shed light on the security design of each component of the CT framework, and disclose that if any of the components are not well protected and configured, the attackers could still exploit fraudulent certificates to launch MitM attacks on an ordinary website, without trigger any alerts in the CT framework. Then, we actually analyzed the various components of CT deployed on the Internet, including the log server, Monitor, and Auditor, and find that any one of them could have various security issues. So the overall security guarantees of CT is jeopardized due to the weak protections of any components.

The remainder is organized as follows. The CT framework and its deployment are described in Section 2. The security design of each component in the CT framework are presented in Section 3. Section 4 analyzes the deployment defects and security threats of CT components on the Internet. Section 5 surveys the related works and Section 6 draws the conclusions.

2 The component of Certificate Transparency

In this section, we illustrate the CT framework and its deployment in practice.

2.1 The CT Framework

In the traditional PKI system, the website applies for a certificate from CA, and in the process of TLS handshake, the browser verifies the certificate. If the browser trust the root CA that issued the certificate and the signature of CA is valid, the certificate will be accepted.

CT scheme is proposed to resist fraudulent certificate attack which binds a domain name to a key pair held by MitM attacker. As shown in Figure 1, compared with the traditional PKI system, the CT framework introduces new component and enhances the functions of the components of the traditional PKI system, so that the CT can achieve the expected purpose.

CA. Compared with the CA in traditional PKI system, a CA supports the CT by adding the following steps. After signing the certificate, the CA submits it to the log server to obtain the SCT. Then, in the TLS handshake phase, the SCT is sent to the website along with the certificate. Alternatively, SCT can be embedded in the certificate as an extension: before signing the certificate, the CA creates a pre-certificate that binds the same data but contains a special

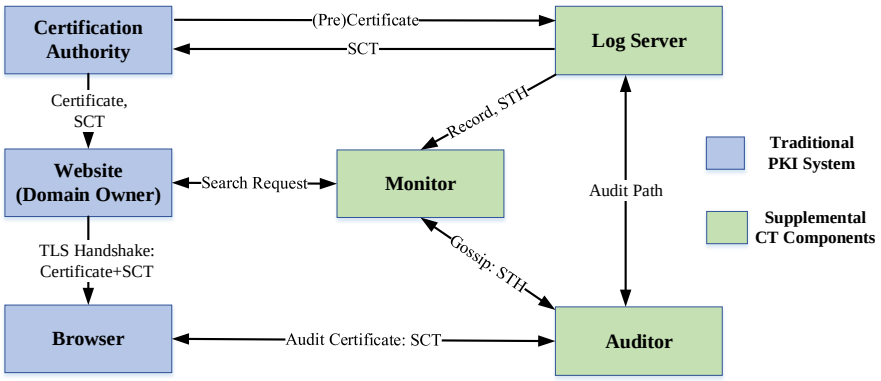

Fig. 1. The Framework of Certificate Transparency

extension indicator. Then, the CA submits the pre-certificate to obtain the SCT and embeds it into the final certificate. According to the CT policy, a certificate may be submitted to multiple Log server to obtain multiple SCT.

Website. Sometimes, it needs to submit its own certificate to the Log server to obtain SCT. In the process of TLS handshake, SCT as an extension is embedded into the certificate or TLS extension and sent to website along with the certificate. Finally, domain owner needs to periodically query all certificates issued for its domain from Monitor, so as to monitor suspicious certificates and detect the fraudulent certificate.

Browser. Compared with the browser in traditional PKI systems, a CT-enabled browser is enhanced with the following functions. In TLS handshakes, a CT-compliant browser verifies the certificate and SCTs (i.e., the signatures by the CA and the log servers). The public keys of approved log servers are preinstalled in browsers by manufacturers. A browser rejects a certificate without enough valid SCTs. After the TLS handshake, the browser periodically sends every SCT to an auditor, to check whether it corresponds to some certificate entry in public logs.

Log Server. Log server is responsible for receiving the certificate and returning SCT as the voucher. It promises to record the certificate into public log within the specified maximum time and make it publicly visible. The certificate recorded in the log is recorded in the form of Merkle hash tree. Log server signs the root node of Merkle hash tree regularly, which is named the signature tree head (STH). This structure makes the certificate record only addable which is convenient for auditing.

There logs are publicly visible, and anyone can act as Monitor to obtain certificate from these logs, monitor suspicious certificate and find fraudulent certificate. Any entity can request STHs to audit the behavior of Log server: (a)

Monitor and Auditor ensure that Log server provides the same view for different entities through exchanging the STHs periodically; (b) Via comparing the two STHs, Monitor or Auditor can check whether Log server is only addable, that is, any particular version of the log is superset of any previous version. Moreover, an auditor requests the audit path from the log server, which is the shortest list of additional nodes in the Merkle tree to compute the root node [24], to check whether an SCT corresponds to a particular entry in the log.

Auditor. As a lightweight software component, Auditor is used to ensure the correctness of the Log server behavior. Auditor can be a standalone service, a TLS client or Monitor component. Through comparing the two STH, Auditor verifies whether the log conforms to only add attribute. Auditor also verifies that each SCT corresponds to a record in the log to ensure that the SCT has been recorded in log via verifying the auditor path.

Monitor. Monitor is responsible for regularly and continuously monitoring suspicious certificates in public log. Monitor obtains all the records from the monitored log set, parses the certificate and checks the certificate of interest. The domain owner can play the Monitor role to monitor and query the certificate of interest; or the third-party monitor completes to process the certificates in public log and provides users with certificate query and monitoring services.

2.2 CT in Practice

CT has been widely deployed on the Internet [36]. In July 2020, we created a list of 93 accessible logs by collecting the information from the log list maintained by Google [20], and the websites of CA companies, third-party monitors and auditors. Then, using the get-roots command of log servers [24], we obtain the list of root CAs accepted by each log. In total, these logs support 607 unique CAs. Many mainstream CA companies (e.g., DigiCert, Comodo, GlobalSign, StarCom, GeoTrust, GoDaddy and Let's Encrypt) have supported the CT scheme [6, 35]. We collect the STHs archived in SSLMate [34], and find that by July 2020, there are over 8.196 billion certificates in these 93 public logs.

CT has been widely used in browsers and TLS applications, including Chrome [19] Apple platform [3], Mozilla Firefox browser [30], OpenSSL [33], Nginx [31], Microsoft AD Certificate Service and Azure Key Vault [28], etc. As of February 2018, at least 60% of HTTPs communication support the CT strategy [36]. In addition, since Jun 2018, Chrome browser and Apple platform began to enforce CT check. Certificates used by the server that do not meet the CT policy will no longer be accept by the browser and platform. The public keys of these approved logs are pre-installed in the browsers and operating systems (OSes). By July 2020, there are 41 approved logs in Chrome [19], and 59 in Apple platform [3]. This will further promote the deployment and application of CT.

Although CT allows the domain owner to act as the Montor [24], the heavy demand of processing and storage will largely prevent the ordinary domain owner from implementing the certificate monitoring and query function alone [10]. At the same time, there are maturely deployed third-party monitor servers on the

Internet. They can obtain records from logs, decode certificate and provide certificate query and monitoring services for users. To our best knowledge, there are 6 third-party monitors on the Internet, namely crt.sh, SSLMate, Censys, Google Monitor, Facebook Monitor, and Entrust CT Search Tool.

Some CT Auditor such as Edgecombe [13] and Merkle Town [6] are also deployed on the Internet. They audit the running state of Log server by verifying STH and SCT. SSLMate also implements part of function of Auditor and executes Gossip verification along with Edgecombe. CT-over-DNS scheme [17] helps browser implement CT audit function and will not disclose their personal browsing history. This scheme has been implemented in Chrome.

3 The Security Design of CT

In this section we analyze the safety security of CT framework and the interaction between the components.

It should be noted that in this study, we adopt the same threat model and hypothesis as the CT scheme [18, 24]. It mainly includes: (a) The CA organization may be attacked or cheated by attackers, and it results in providing the fraudulent and false certificate to the Log server; (b) The correctness behavior of Log server is ensured by sufficient, redundant and actually deployed Auditor and Monitors.

3.1 Log

As the core of CT framework, Log server is responsible for recording all accepted certificate information, returning the digital signature commitments of SCT et.al, providing public assessable interface, supporting Monitor of the individuals or organizations to extract certificate and accepting Auditor of the third part to audit the security of log server behavior. Therefore, the parameter configuration, external strategy and running quality of Log server will affect many components of CT and mass CT-enable devices.

For example, (a) The key of Log server. Public and private key of Log server is mainly used for digital signature of STH, SCT. The public key is used to verify signature by CA organization, Monitor, Auditor and client through preset or public available manner. Therefore, once the public and private key of the Log server needs to be updated due to leak and expiry, it may affect these components. The number of CA organization, Monitor and Auditor deployed in the Internet is limited (hundreds or thousands). These components can be updated in a safe and controllable way. For example, offline out of band mode, manual update. The massive client devices are deployed all over the world, and different platforms adopt different update methods. For example, Chrome browser implements to online update the list of approved log servers through regularly push. However, in the early Apple platform, the accepted log server list is preset into the platform source code which can not be updated independently. This may cause potential security threats to users who do not update the version of Apple

operating system in time. The issued SCT for issuing fraudulent certificate via the leaked log server private key will still be authenticated by the client.

(b) List of accepted CA. Each log server maintains a list of acceptable CA. Only certificates issued by CA on the list are allowed to be recorded on the log server. The CA list accepted by log server will directly influence CA organization, website and mass clients. For example, some publicly trusted CA may not be accepted by any log server, which means that CT can not cover all certificates deployed in the network. On the one hand, some certificates are invisible, and fraudulent certificates issued by these CA can be verified by CT-unable browser and Monitor can never monitor the fraudulent certificate. On the other hand, the legitimate certificate issued by these CA cannot be trusted by the CA and verified by CT-enable browser.

(c) Running strategy. With the wide application and deployed of CT, more and more certificates are submitted to log server. This results in that a large amount of certificate information is recorded in the log server (for example, by July 2020, 93 log server has recorded a total of 8.196 billion certificate information, and 82.6 million new certificates are added every day), which increases the long term operation burden of log server operators. In order to solve this problem, researchers propose a partitioned log server based on the validity of certificate to limit the range of received certificates. For example, in Google argon 2020, only accepting these certificates whose validity period is within 2020 enables the operator have the right to shut down the server after the end of 2020 without affecting the use of users. For early non-partitioned log servers, operators plan to freeze these servers within a limit time and no longer receive new certificate information. However, this also leads to some issues: as mentioned above, the early system of Apple platform directly wrote the accepted log list into the system source code, so it is unable to update the log list online. Once the accepted log server stops working, CA can only submit the newly issued certificate to other log servers, which can not be verified by Apple platform and affect the normal use of legal certificates. Therefore, at present, operators plan to reduce the cost of maintenance through decreasing the scope of accepted CA and gradually reducing the growth of log. In addition, the massive and expired certificate information recorded in the log server also adds extra burden to the third-part monitoring audit institutions such as Monitor and Auditor, which influences the word efficiency.

3.2 Monitor

Monitor plays a key role in monitoring fraudulent certificate. The service quality of Monitor certificate monitoring method will directly determine the effectiveness of CT, and further affect the promotion and deployment of CT. In practice, there are many factors that determine the service quality of Monitor, including monitoring strategy, interface rule, and so on. The evaluation is also reflected in man aspects, including timeliness, reliability and security. Defects or vulnerabilities of any factor or feature may lead to that existence fraudulent certificate

cannot be detected by Monitor in time, and then invisible to the legitimate domain name owner, and used to launch attacks by attackers.

If there is a vulnerability of flaw in monitor's implementation, the attacker will use the vulnerability to evade Monitor to monitor the fraudulent certificate. If there is a fraudulent certificate which is issued by the public trust CA, meets the CT policy, is verified by the browser, but cannot be detected by the monitor and is invisible to the legitimate domain name owner, the attacker can use the fraudulent certificate to launch a middleman or identity impersonation attack on the target. The security and reliability problems of Monitor will directly affect the security effect of CT framework: in TLS/HTTPS ecosystem, certificates conforming to CT policy should be more trustworthy.

To implement monitor technology in the certificate transparency system is essentially to establish a fraudulent monitoring system to ensure that all valid certificate set related to the monitored domain can be safely, reliably and timely fed back to the legitimate domain name owner. To achieve the CT target, Monitor certificate monitoring scheme should meet the following requirements: (a) it can timely and reliably monitor all valid certificate set related to target domain name; (b) it can that monitor all legitimate domain name owners can be safely and completely fed back the monitoring result; (c) it should have certain fault tolerance, comprehensive and fast security measurement mans and be able to identify and repair faults and resist malicious attacks.

In the actual deployment, there are many challenges referring to many operations to achieve this goal. For example, (a) The huge amount of data (millions per day) recorded in log which is rapidly growing brings challenges for full coverage [26]; (b) the format of certificate and domain name is diversified and the binding relationship is complex, including a variety of special characters, which increases the difficulty and unpredictability for correct analysis [26]; (c) the association and storage mode among different certificate information introduces uncertainty for complete query and monitoring. If any process of the above fails, Monitor scheme may not be able to provide the expected services.

Some studied have shown that CT Monitor, which provides certificate query and fraudulent certificate monitoring service on the Internet, has obvious defects in terms of reliability and timeliness and exists hidden danger of being attacked so that it can not provide users timely and complete certificate set of monitored domain name. In addition, these monitors do not achieve perfect TLS/HTTPS configuration. Compared with ordinary domain name websites, they do not achieve obvious security enhancement and the generated potential TLS MitM vulnerability will seriously threaten the overall security of CT framework.

3.3 Auditor

Auditor plays a key role in auditing log sever behavior. The service quality of Auditor will directly determine the reliability of CT. In practice, there are many factors that determine the service quality of Auditor, including deployment mode and location, execution mode, coverage scope and cycle, robustness and specific

cost. Any factor of feature exists defects or problems, it may lead to that the log server with problematic behavior is unable to be detected by the Auditor in time and then there may be fraudulent certificate which is used to launch attacks by attackers.

If there are loopholes and defects in the implementation of Auditor, the attackers will exploit the vulnerability to avoid the detection of malicious log server. If there is a fraudulent certificate which is issued by publicly trust CA and has malicious log server to issue SCT to satisfy the CT policy, and the browser passes the verification, but it is not recorded in the log server and the Auditor does not detect the exception, so it can not be find by Monitor in time and is invisible to the legitimate domain name owner, then the attacker can use the fraudulent certificate to initiate the middleman or the identity impersonation attacks for target. The safety and reliability problems of the Auditor directly affect the safety effect of CT framework.

The implementation of Auditor technology in certificate transparency system is essentially to establish a malicious log server monitoring system in the Internet to ensure that the monitored log server meets the consistency and existence proof for external. In the actual deployment, Auditor has many challenges: (a) Privacy. Auditor can be integrated into the browser client as an additional function, which can alleviate the privacy leakage of users, but it will increase the burden of client; if the Auditor provides services as an independent component, it will lead to the privacy disclosure of the client. (b) Security. When Auditor performs consistency and existence check, it can only select part of certificate for verification and interaction STH, SCT information of part of third-party Auditor and Monitor which cannot cover all the inspection targets. In the process of interaction, it may be attacked by middleman and results in the tampering of verification results; or malicious log server cooperating with a third-part cheats the Auditor. (c) Performance. When a client establishes a TLS connection with the server, it has high requirement for time, usually within milliseconds. If the client requests the Auditor to perform existence check when executes certificate verification, it needs to connect with a third party to obtain the existence check path from the log server. If the extra cost is too large, it will seriously affect the performance of client network connection and then affect the promotion and deployment of Auditor.

3.4 Browser

As the certificate verifier, the browser needs to be as the main body to participate to check whether the certificate meets the CT policy. These checks include the validity, quantity and existence proof of SCT signature, so as to alleviate the security threat brought by the fraudulent certificate. Therefore, as the beneficiary of CT framework, the support strategy of browser and other client directly influences the promotion and deployment of CT. In actual deployment, the following aspects of browser will affect the deployment and security of CT.

(a) Trust root. Including trusted root CA list and log server list. The former determines which CA must follow the CT policy, while the latter defines at

least which certificate issued by CA submits to which log server. In addition, as mentioned earlier, the update way of trust list will also affect many components, including the running status of log server, the accepted list of CA, and so on.

(b) STC check strategy. The browser's requirements for the number and source of SCTS will affect the scope of the logging server where the CA submits the certificate. This may further potentially affect the range of monitoring log server list of Monitor. On the one hand, the requirement of SCT transmission from browser will influence massive website servers to application and deployment wat of CT. Finally, the SCT check manner of browser also affects the deployment quality of CT. If Chrome and others cannot successfully update the trust list of log server, they will adopt "soft error" processing method to pause CT examination. This can reduce the false positive generated by using the old trust list to perform CT check and avoid affecting the user experience. However, it also seriously affects the user security: the adversary can launch a continuous middleman attack to prevent the browser from updating the trust list of the local log server, force the browser to downgrade the CT check and then use the fraudulent certificate to launch further attacks.

(c) Auditor. The browser as the main body of certificate verification, its checking strategy for the existence of certificate will also directly influence Auditor and other CT audit institutions, including the verification method, frequency and policy. This can affect the check coverage and security quality of log server from Auditor. In addition, this may potentially affect the TLS security connection performance of the client, disclose the current network access information of user and cause the privacy disclosure of the user.

3.5 Website and CA

Website and CA are the security enhancement targets of CT mechanism. In practice, any website may be attacked by fraudulent certificate and any CA may be forced to issue fraudulent certificate. This is consistent with the treat model and assumptions of CT. The support strategy of website and CA for CT will also directly influence the deployment of CT, including the selection of log server and transmission mode of supported SCT.

4 Certificate Transparency on the Internet

In this section, we study the security of the CT component on the Internet, to analyze the strength of CT in practice.

In practice, there are hundreds of log servers and multiple Monitor and Auditor deployed on the Internet. They run independently, backup and cooperate with each other to ensure the effective work of CT mechanism. As shown above, in the actual deployment, each component face with many challenges. There are many factors affecting it from external and different components also affect with each other. Any problem of them will lead to security problems of the whole

framework of CT and ultimately affect the application effect of CT. At present, the main problems in the actual deployment of CT components are as follows:

Log server. The certificate information recorded by the log server increases rapidly, and the query and audit requirements also increase day by day. These increments all increase the cost of storage, calculation, network bandwidth and other operation and maintenance of log server. In addition, the accepted CA list of log server is different from that trusted by mainstream platforms, which makes it impossible for CT framework to fully cover the CA and realize the ecological supervision of TLS certificate by CT.

Monitor. The existing research shows that there are many problems in mainstream monitors, including the correct resolution of certificate, timely and efficient processing of massive certificates. When providing external services, Monitor has some problems in terms of reliability, timeliness and security of external interaction.

Auditor. At present, there is no third-party Auditor organization that can perform the full functions of the Auditor in the Internet. Some third-party organizations can perform consistency check of some log server; Chrome implements the function of checking the existence of certificate, but it is not enabled by default. This leads to the lack of Auditor function of CT mechanism in practice.

The problems existing in the practical application of CT components make CT possible to suffer from various attacks, which seriously endanger the overall security of CT.

Downgrade attack: These vulnerabilities, including that CT does not cover CA completely, browser does not strictly implement CT policy and various monitoring vulnerabilities exist in Auditor and Monitor, make adversaries use them to maliciously construct fraudulent certificate and evade the inspection of CT mechanism to launch attacks.

MitM attack: tampering with the Monitor monitoring and the audit results of Auditor verification, hiding fraudulent certificate for the domain name owners and browsers.

Denial of service attack: this kind of attack is launched on log server, Monitor and Auditor, which makes them unable to provide timely and normal services to the outside and affects CT examination.

5 Related Work

CT deployment. The deployments of CT in the Internet are investigated from various perspectives. Stark et al. [36] completed a comprehensive study of CT deployment across the Internet, including compliance, user experience, and potential risk. Nykvist et al. [32] studied the adoption of CT in Alexa Top-1M websites and evaluated the performance of SCT delivery methods. Scheitle et al. [35] analyzed the server-side deployment of CT, and discussed the subdomain information leakage caused by the certificates in public logs. B. Li et al.

conducted systematic in-depth research and analysis on CT Monitor from the perspectives of reliability [26] and TLS/HTTPS configurations [25] respectively. Gustafsson et al. [21] characterized 11 public logs and highlight the differences of certificates they record. Amann et al. [2] finished a large-scale study on the adoption of various TLS/HTTPS security enhancements, including CT, HPKP, HSTS, CAA, SCSV downgrade prevention and DANE.

CT extensions. Following the basic CT framework, several designs were proposed to improve the security and/or performance. Matsumoto et al. [27] studied the incentives of parities in the PKI system to deploy log-based enhancement schemes, and proposed the deployment status filters to detect the deployment status of a domain against the downgrade attacks. Dowling et al. [11] defined four security properties of logging schemes, and formally prove that CT implements these security properties. An efficient gossip protocol was proposed to detect several types of log inconsistencies [5]. Eskandarian et al. [14] proposed to audit a CT log without exposing user privacy by zero-knowledge proofs, and with the support of non-public subdomains by commitments with binding and hiding properties. Dahlberg et al. [10] proposed a verifiable light-weight monitoring, which enabled users to verify the correctness of the certificate notification from monitors. Tomescu et al. [37] introduced an append-only authenticated dictionary to construct logs, to provide efficient append-only proofs and lookup proofs.

TLS Certificate on the Internet. The certificates in public logs help to understand the TLS/HTTPS ecosystem. Gasser et al. [15] used the certificates in CT logs to investigate the violations of the baseline requirements for the certificate issuance [4]. Cui et al. [9] analyzed multiple attributes of forged certificates in the wild, such as preferences, causes, and attributes. Aertsen et al. [1] exploited the data obtained from several CT logs to study the certificate services of Let's Encrypt adopted in different organizations, hosts and domains. VanderSloot et al. [39] attempted to present a complete view of the certificates in the wild, by integrating the certificates in logs with data from passive measurement, active scanning, and search engines.

6 Conclusion

Certificate Transparency (CT) is proposed to detect fraudulent certificates and improve the accountability of CAs. In this paper, We analyzed the overall CT framework and its components, and find that, in order to achieve the design goal of CT, CT components themselves need to formulate reasonable strategies and implement them correctly, and each component must ensure that it is more secure and reliable than regular TLS sessions when exchanging information. Then, we analyzed the security of each component of the CT framework in practical deployment and its impact on other components. The analysis results show that each component faces various challenges in the implementation process, and its own strategies and implementation methods can influence other components to

different degrees. If the CT components cannot be unified, coordinated and reasonably planned and deployed, then an attacker may attack any component, making CT unable to achieve its intended purpose, to conceal the fraudulent certificates exploited in the MitM attacks on the target website. Therefore, the overall security guarantees of CT is jeopardized due to the weak protections of any components.

References

1. Aertsen, M., Korczynski, M., Moura, G., Tajalizadehkhoob, S., van den Berg, J.: No domain left behind: Is Let's Encrypt democratizing encryption? In: 2nd Applied Networking Research Workshop (ANRW). pp. 48–54 (2017)
2. Amann, J., Gasser, O., Scheitle, Q., Brent, L., Carle, G., Holz, R.: Mission accomplished? HTTPS security after DigiNotar. In: 17th Internet Measurement Conference (IMC). pp. 325–340 (2017)
3. Apple Inc: Certificate transparency in Apple (2018), https://support.apple.com/en-us/HT205280
4. CA/Browser Forum: Baseline requirements for the issuance and management of publicly-trusted certificates, version 1.6.1 (2018), https://cabforum.org/baseline-requirements-documents/
5. Chuat, L., Szalachowski, P., Perrig, A., Laurie, B., Messeri, E.: Efficient gossip protocols for verifying the consistency of certificate logs. In: 3rd IEEE Conference on Communications and Network Security (CNS). pp. 415–423 (2015)
6. Cloudflare Inc: Explore the certificate transparency ecosystem (2018), https://ct.cloudflare.com/
7. Comodo Group Inc: Comodo report of incident (2011), https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html
8. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.T.: IETF RFC 5280 - Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile (2008)
9. Cui, M., Cao, Z., Xiong, G.: How is the forged certificates in the wild: Practice on large-scale SSL usage measurement and analysis. In: 18th International Conference on Computational Science (ICCS). pp. 654–667 (2018)
10. Dahlberg, R., Pulls, T.: Verifiable light-weight monitoring for certificate transparency logs. arXiv ePrint (2017), https://arxiv.org/abs/1711.03952
11. Dowling, B., Günther, F., Herath, U., Stebila, D.: Secure logging schemes and certificate transparency. In: 21st European Symposium on Research in Computer Security (ESORICS). pp. 140–158 (2016)
12. Eckersley, P.: A Syrian man-in-the-middle attack against Facebook (2011), https://www.eff.org/deeplinks/2011/05/syrian-man-middle-against-facebook
13. Edgecombe, G.: Certificate transparency monitor (2018), https://ct.grahamedgecombe.com/
14. Eskandarian, S., Messeri, E., Bonneau, J., Boneh, D.: Certificate transparency with privacy. In: 17th International Symposium on Privacy Enhancing Technologies (PETS). pp. 329–344 (2017)
15. Gasser, O., Hof, B., Helm, M., Korczynski, M., Holz, R., Carle, G.: In log we trust: Revealing poor security practices with certificate transparency logs and Internet measurements. In: 19th International Conference on Passive and Active Measurement, PAM. pp. 173–185 (2018)

16. GlobalSign: Security incident report (2011), https://www.globalsign.com/resources/globalsign-security-incident-report.pdf
17. Google Inc: Certificate transparency over dns (2016), https://github.com/google/certificate-transparency-rfcs/blob/master/dns/draft-ct-over-dns.md
18. Google Inc: Certificate transparency (2018), http://www.certificate-transparency.org/
19. Google Inc: Certificate transparency in Chrome (2018), https://groups.google.com/a/chromium.org/forum/#!msg/ct-policy/wHILiYf31DE/iMFmpMEkAQAJ
20. Google Inc: Known logs (2018), http://www.certificate-transparency.org/known-logs
21. Gustafsson, J., Overier, G., Arlitt, M.F., Carlsson, N.: A first look at the CT landscape: Certificate transparency logs in practice. In: 18th International Conference on Passive and Active Measurement (PAM). pp. 87–99 (2017)
22. Heather Adkins: An update on attempted man-in-the-middle attacks (2011), https://security.googleblog.com/2011/08/update-on-attempted-man-in-middle.html
23. Kent, S.: IETF Draft - Attack and Threat Model for Certificate Transparency (2019)
24. Laurie, B., Langley, A., Käsper, E.: IETF RFC 6962 - Certificate transparency (2013)
25. Li, B., Chu, D., Lin, J., Cai, Q., Wang, C., Meng, L.: The weakest link of certificate transparency: Exploring the tls/https configurations of third-party monitors. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). pp. 216–223. IEEE (2019)
26. Li, B., Lin, J., Li, F., Wang, Q., Li, Q., Jing, J., Wang, C.: Certificate transparency in the wild: Exploring the reliability of monitors. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2505–2520 (2019)
27. Matsumoto, S., Szalachowski, P., Perrig, A.: Deployment challenges in log-based PKI enhancements. In: 8th European Workshop on System Security (EuroSec). pp. 1–7 (2015)
28. Microsoft Inc: Certificate transparency in Microsoft (2018), https://blogs.msdn.microsoft.com/azuresecurity/2018/04/25/certificate-transparency/
29. Morton, B.: More Google fraudulent certificates (2014), https://www.entrust.com/google-fraudulent-certificates/
30. Mozilla: Certificate transparency in Mozilla (2018), https://wiki.mozilla.org/PKI:CT
31. Nginx: Certificate transparency in Nginx (2018), http://www.certificate-transparency.org/resources-for-site-owners/nginx
32. Nykvist, C., Sjöström, L., Gustafsson, J., Carlsson, N.: Server-side adoption of certificate transparency. In: 19th International Conference on Passive and Active Measurement (PAM). pp. 186–199 (2018)
33. OpenSSL: Certificate transparency in OpenSSL (2018), https://www.openssl.org/docs/man1.1.0/crypto/ct.html
34. Opsmate Inc: Certificate transparency log growth (2018), https://sslmate.com/labs/ct_growth/

35. Scheitle, Q., Gasser, O., Nolte, T., Amann, J., Brent, L., Carle, G., Holz, R., Schmidt, T.C., Wählisch, M.: The rise of certificate transparency and its implications on the Internet ecosystem. In: 18th Internet Measurement Conference (IMC). pp. 343–349 (2018)
36. Stark, E., Sleevi, R., Muminovic, R., O'Brien, D., Messeri, E., Felt, A.P., McMillion, B., Tabriz, P.: Does certificate transparency break the web? measuring adoption and error rate. In: 40th IEEE Symposium on Security and Privacy (S&P) (2019)
37. Tomescu, A., Bhupatiraju, V., Papadopoulos, D., Papamanthou, C., Triandopoulos, N., Devadas, S.: Transparency logs via append-only authenticated dictionaries. IACR Cryptology ePrint Archive (2018), https://eprint.iacr.org/2018/721
38. University of Michigan: Censys (2018), https://censys.io/
39. VanderSloot, B., Amann, J., Bernhard, M., Durumeric, Z., Bailey, M., Halderman, J.A.: Towards a complete view of the certificate ecosystem. In: 16th Internet Measurement Conference (IMC). pp. 543–549 (2016)
40. VASCO Data Security International Inc: DigiNotar reports security incident (2011), https://www.vasco.com/about-vasco/press/2011/news_diginotar_reports_security_incident.html
41. Wikipedia: Flame (malware) (2017), https://en.wikipedia.org/wiki/Flame_(malware)
42. Wilson, K.: Distrusting new CNNIC certificates (2015), https://blog.mozilla.org/security/2015/04/02/distrusting-new-cnnic-certificates/