Server Notaries for TLS/SSL Trust Model

An Alternative Approach to the Current CA Model

Emre Yüce [1], Ali Aydın Selçuk [2]

INTRODUCTION

Internet users mostly rely on TLS/SSL for establishing secure web connection. Websites send a certificate to be used through this communication and browsers leverage the Web PKI trust model based on certificate authorities in order to verify the received certificate. However, recently published vulnerabilities in the current Web PKI trust model have initiated discussions on alternative methods for verifying or tracking the certificates published.

Current Web PKI Trust Model

CA/Browser trust model is built upon CAs and client side application trust stores. CAs constitute a great deal in this construction since any certificate issued by a CA will be trusted by each client. This is the result of the fact that each root CA is equally trusted by the browsers.

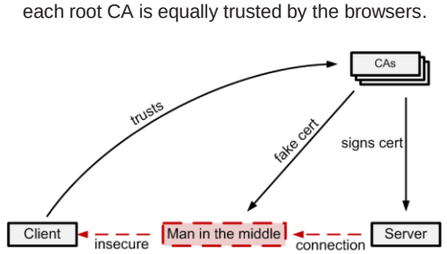

Figure 1: MITM attack model

Issues with the Current Model

- Compromised CAs issuing fraudulent certificates, uncompromised CAs issuing fraudulent certificates by mistake or otherwise. (MITM attacks)
- Every CA can issue a certificate for any domain name, no auditing.
- Lack of trust agility:
  - Whom to trust? For how long?
- No easy way for a node to use self signed certificates.

PROPOSED SOLUTIONS

Notary Based Solutions

- **Perspectives** (Wendlandt et al., USENIX 2008), **Convergence** (Moxie, BlackHat 2011).
- **Notaries** are publicly available servers.
- Notaries observe a server's public key via **multiple network vantage points** (detecting localized attacks).
- Notaries keep the record of the server's key over time (recognizing short-lived attacks).
- Requires one or more public keys (public keys of notaries) to bootstrap trust.
- For binding keys to hosts notary based solutions use **automated network probing**.
- Notary based solutions give **control over trust decisions to the user**.

Logging Based Solutions

Certificate Transparency (Langley et al.), which is a publicly auditable log system whose aim is to detect forged certificates, consists of the following components.

- Log Servers**:** Cryptographically assured, publicly auditable, append-only records of certificates.
- Monitors: Servers that periodically contact all of the log servers and watch for suspicious certificates (e.g. certificates that have CA capabilities).
- Auditors are components deployed on clients which verify that log servers are behaving correctly and verify that a particular certificate appears in a log by querying log servers periodically.

Pinning Based Solutions

- **Chrome Pinning**: Solved several Web PKI issues but not feasible for the whole Internet.
- **Trust Assertions for Certificate Keys – TACK:** Clients pin to a server-chosen signing key (a.k.a TACK signing key or TSK), which signs the server's TLS/SSL keys. Using TSK provides pinning flexibility.

OUR APPROACH

Observations

Notary based methods are successful at detecting attacks close to the client.

- Weak if a global/regional attack (governmental, country-wide etc.) exists.
- Our aim is to detect a wide-area MITM.

Google Chrome pinning does something similar:

- Google checks how its services' certificates are seen worldwide.
- Able to detect misissued/fake certificates but **not able to localize the adversary** if exists.
- Focus on localizing the attack after a successful detection.

Threat Model

In our scenario we consider an adversary who has access to all communication between the server and the client. The adversary's aim is to eavesdrop and tamper with this communication by executing MITM attacks against TLS/SSL.

- Adversary can achieve a MITM attack by obtaining a **forged certificate** for the server's domain that is signed by some trusted CA **or** by using an **untrusted (e.g. self-signed)** certificate.
- The server observes fake certificate(s) **through all notaries** (i.e. a MITM attack close to the server e.g. in the same network) **or through some of the notaries** (e.g. country-wide MITM attacks) depending on the location of the adversary.

Our method does not consider attacks exploiting TLS/SSL implementation or configuration errors. Also it is assumed that the server is not compromised.

Components

Server notaries consists of a TLS/SSL server, a number of pre-deployed notaries and an adversary. We assume that the server has already obtained the current list of active notaries and their public keys.

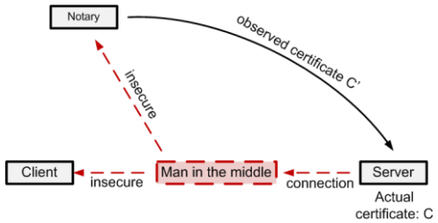

Figure 2: Server Notaries Model

Our proposal: Server Notaries

Currently there exist several proposals to detect MITM attacks against TLS/SSL as mentioned above. However none of these proposals give the server the control to observe its own certificate from different network vantage points. We propose the **server notaries** method to detect MITM attacks launched close to the server. In other words, this method enables observing server's TLS/SSL certificate and detecting MITM attack attempts launched between the server and any of the pre-deployed notaries.

Deploying server notaries; TLS/SSL servers may query their own certificates leveraging notaries.

- The server periodically sends queries to the notaries.
- The server will observe how its certificate is observed by notaries.
- So the server will be able to detect MITM attacks close to itself.

Detecting & Locating the Adversary

- Probability of **detecting** the adversary is equal to the probability of observing a fake certificate.
- Probability of **locating** the adversary is equal to the probability of observing a fake certificate at a node and observing the genuine certificate at the previous node.

Discussions

Depending on the forged certificates in the wild and security breaches observed, the possible location of the adversary with respect to server and client is (1) **close to the client**, (2) **somewhere in the middle** or (3) **close to the server** as presented in **Figure 3**.

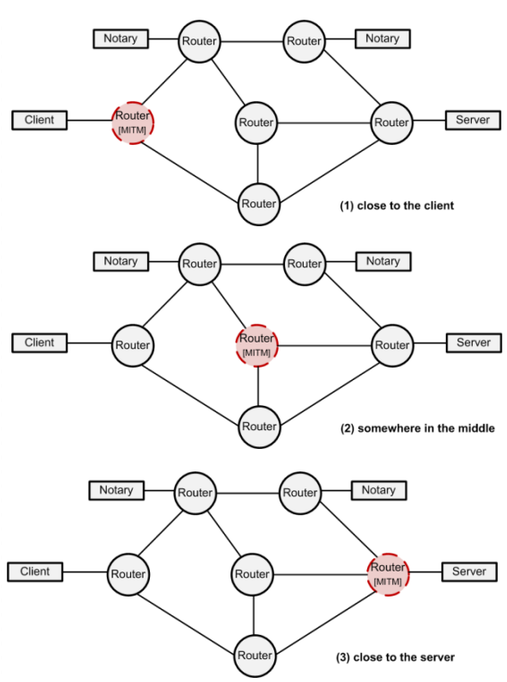

Figure 3: Possible Locations of Adversary

Server notaries method is an effective detection method for cases (2) and (3). However for the case (1) our method is less effective unless a notary is deployed in the vicinity of the client. This case does not constitute a security breach if a notary based solution is deployed by the client. Therefore the server notaries method complements notary based solutions such as Perspectives or Convergence which are vulnerable to scenario (2) and (3) through the first connection or certificate updates.

CONCLUSION

Currently we are developing a model to measure how effective the server notaries method is for detecting and locating an adversary. For this purpose we are developing simulation scenarios based on actual BGP data.

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

[1] Middle East Technical University Institute of Applied Mathematics Department of Cryptography

[2] TOBB University Of Economics & Technology Department of Computer Engineering