# Poster: Strengthening User Verification Using Browser Fingerprint and Cyber Deception

Xiaoyun Li
Beijing University of Posts and
Telecommunications
Beijing, China
lxy691219491@bupt.edu.cn

Xiang Cui
Beijing University of Posts and
Telecommunications
Beijing, China
cuixiang@iie.ac.cn

Chaoge Liu
Institute of Information Engineering,
Chinese Academy of Sciences
Beijing, China
liuchaoge@iie.ac.cn

***Abstract*—Certificate authentication has already been widely used, especially in OA system. However, a stolen client certificate is still valid, which represents a potential security risk. Given browsers are important tools when employees log in OA system, in this poster, we introduce browser fingerprint and web shadow service to strengthen user verification. Firstly, we collect information by browser and select stable attributes to form browser fingerprint. Moreover, we generate client certificate together with browser fingerprint. Furthermore, we compare the browser fingerprint in the certificate with the one generated real time in login process. And then we judge if two fingerprints originate from the same employee's browser. If not, we forward the traffic to a web shadow service of OA system to keep further observing. Once we identify attackers, we track them based on information collected during login process.**

***Keywords—user verification; browser fingerprint; cyber deception; shadow service***

## I. Introduction

Many targeted attacks focus on Office Automation (OA) system, which contains rich and valuable information. To prevent attackers from illegally logging in, some organizations use client certificates to verify employees' identity in OA system. However, a stolen client certificate will still take effect, and this may cause a potential security threat.

Browser fingerprint, proposed by Eckersley [1], is a combination of information collected from user's browser. It is often used to identify and track users. However, some features may change frequently over time. Fortunately, Antoine Vastel studied the stability of features used to generate fingerprints. [2] In this poster, we select a new feature set to generate browser fingerprint according to stability. In addition, we proposed a novel approach to strengthen user verification by adding browser fingerprint to the client certificate and the verification process.

Web shadow service, proposed by Lin [3], is developed by multifarious cyber deception technology. The shadow adds various fake sensitive data and deception elements on a clone of the original website. It constructs a deception environment to confuse attackers and help analyze the attack intent. In our approach, if two fingerprints are inconsistent, we forward the traffic to the deception environment and keep observing the user's operation with the help of web shadow service. Furthermore, once the user makes some suspicious operations, we recognize him as attackers and track him based on information collected during login process.

## II. Generation of Client Certificate

### A. *Collection and Generation of Browser Fingerprint*

To add browser fingerprint to client certificate, we need to select the attributes carefully. On the one hand, the features used to form fingerprint should be stable so that there is no need for users to update the certificate frequently. On the other hand, the generation process should be secret so that attackers cannot forge the same fingerprint easily.

In the process of information collection, we gather all attributes described in Table I and transferred encrypted results to server, where we use some of them to generate browser fingerprint. Table I is from FP-STALKER, and in their paper, they use it to show statistical analysis of attribute stability. [2] The last column presents that local storage remains stable for 320.2 days in 95% of the browser instances, which means it is quite stable. And user agent changes every 39.7 days for 50% of the browser instances, which shows that it is unstable.

TABLE I. Browser Fingerprint Attribute Stability

| | | ***Percentile(days)*** | | |
|---|---|---|---|---|
| **Attribute** | ***Trigger*** | ***50th*** | ***90th*** | ***95th*** |
| Resolution | Context | Never | 3.1 | 1.8 |
| User agent | Automatic | 39.7 | 13.0 | 8.4 |
| Plugins | Automatic/User | 44.1 | 12.2 | 8.7 |
| Fonts | Automatic | Never | 11.8 | 5.4 |
| Headers | Automatic | 308.0 | 34.1 | 14.9 |
| Canvas | Automatic | 290.0 | 35.3 | 17.2 |
| Major browser version | Automatic | 52.2 | 33.3 | 23.5 |
| Timezone | Context | 206.3 | 53.8 | 26.8 |
| Renderer | Automatic | Never | 81.2 | 30.3 |
| Vendor | Automatic | Never | 107.9 | 48.6 |
| Language | User | Never | 215.1 | 56.7 |
| Dnt | User | Never | 171.4 | 57.0 |
| Encoding | Automatic | Never | 106.1 | 60.5 |
| Accept | Automatic | Never | 163.8 | 109.5 |
| Local storage | User | Never | Never | 320.2 |
| Platform | Automatic | Never | Never | Never |
| Cookies | User | Never | Never | Never |

In our approach, considering the convenience of user, we select several stable attributes in Table I to generate browser fingerprint, including vendor, language, dnt, encoding, accept, local storage, platform and cookies. Since these attributes are relatively stable, users don't need to update their certificate frequently. And other attributes can also be helpful to track attackers.

### B. *Generation of client certificate*

OA systems usually act as Certificate Authority (CA) and issue self-signed certificates for users. Therefore, we design the generation of Client Certificate as Fig.1.

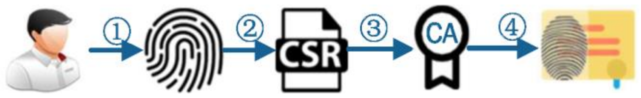

Fig.1 generation of client certificate

① Users generate browser fingerprint on their browsers.

② When generating a client Certificate Signing Request to CA, users add their browser fingerprint as a part of it.

③ When CA receives the client Certificate Signing Request, CA verifies the identity of users offline.

④ After users' identities have been verified, CA issue the client certificate for them.

If users change browser, they need to request a new client certificate with new browser fingerprint.

## III. Approach

### A. *Verification process*

① During TLS Handshake, the server verifies the user's identity by client certificate, and remembers the browser fingerprint written on the certificate as FP1. If the verification is successful, the server will return login page.

② The JavaScript script on the login page collects information mentioned in Table I and sends results encrypted to server. To be safe, the code of the script need to be obfuscated. And then server generates real time browser fingerprint (as FP2) using attributes mentioned above.

③ The server will check username and password with records in database, and meanwhile, compare FP1 with FP2.

④ If FP1 equals FP2 and the username and the password are correct, the server will allow user access to the OA system.

⑤ If FP1 is inconsistent with FP2, it proves that this is a suspicious login behavior, which means the certificate may be stolen. Firstly, the server immediately sends an alert to the admin and forwards the traffic to the web shadow service of OA system. And then the admin monitors user's behavior and judge the identification of the user. If user only carries on routine operation, it proves that he is a normal employee with changed fingerprint. Otherwise, there is a high possibility that the target is an attacker. And then admin can analyze the attack intent and track the attacker with FP2 and information collected from the attacker's browser.

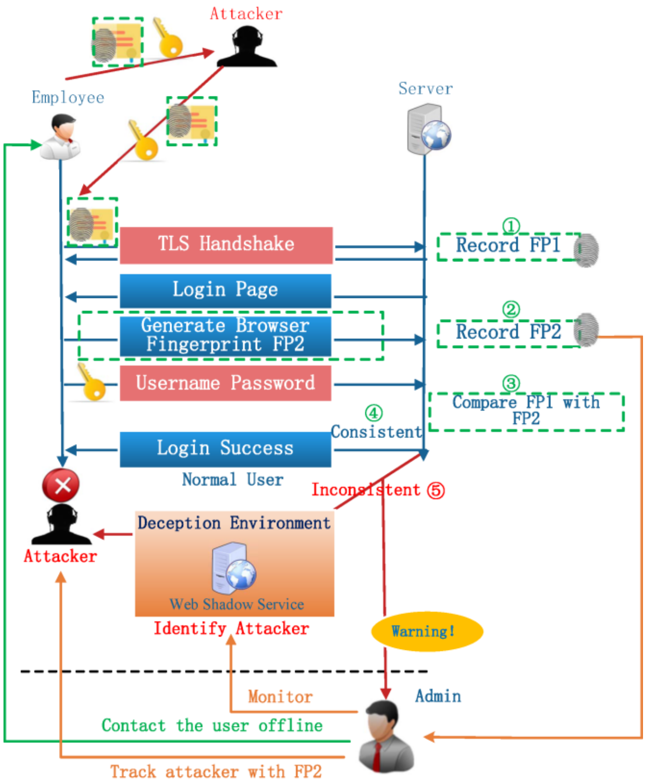

Fig.2 verification process

## IV. Conclusion

In this poster, we propose an approach to strengthen user verification based on browser fingerprint and cyber deception. We focus on choosing attributes to generate fingerprint and adding it to client certificate. And then we add browser fingerprint and deception environment to verification process. Benefit from this, we can not only realize user authentication, but also identify attackers and track them.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments for improving this paper. This work is supported by the Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences and Beijing Key Laboratory of Network security and Protection Technology.

## References

[1] Eckersley P. How unique is your web browser? [C]//International Symposium on Privacy Enhancing Technologies Symposium. Springer Berlin Heidelberg, 2010: 1-18.

[2] A. Vastel, P. Laperdrix, W. Rudametkin, R. Rouvoy, "FP-STALKER: Tracking Browser Fingerprint Evolutions"[C]// IEEE S&P 2018-39th IEEE Symposium on Security and Privacy. IEEE, 2018: 1-14.

[3] Lin J, Liu C, Cui X, Jia Z. POSTER: A Website Protection Framework Against Targeted Attacks based on Cyber Deception. 38th IEEE Symposium on Security and Privacy.