SoK: A Comprehensive Evaluation of 2FA-based Schemes in the Face of Active Concurrent Attacks from User Terminal

Ahmed Tanvir Mahdad
Texas A&M University
College Station, TX, USA
mahdad@tamu.edu

Nitesh Saxena
Texas A&M University
College Station, TX, USA
nsaxena@tamu.edu

ABSTRACT

Malware-infected terminals pose a pervasive threat to authentication systems. As password-only authentication cannot adequately protect against malware on terminals, the literature proposes several authentication methods claiming to provide security in the presence of significant security threats, including infected terminals. Most methods incorporate a password-independent factor in the authentication process to mitigate these threats. According to the community view in the literature, 2FA-oriented methods appear to be secure in the presence of malware on the authentication terminal. In this work, we systematize these 2FA-based academic schemes' threat models and authentication procedures to examine how they ensure security at every step of the authentication process. Additionally, we present an active concurrent attack framework named *CSI* (Concurrent Session Injection) and have done a comprehensive analysis of studied academic authentication systems against it. Furthermore, we systematize secure authentication systems from the literature that claim to provide protection against user terminal malware and concurrent attacks and point out their potential vulnerabilities. Our research emphasizes the significance of taking proper security measures against such threats and creates the opportunity to design more secure authentication systems in future research.

CCS CONCEPTS

• **Security and privacy**;

KEYWORDS

2FA, Authentication, Concurrent Attack, SoK

ACM Reference Format:
Ahmed Tanvir Mahdad and Nitesh Saxena. 2023. SoK: A Comprehensive Evaluation of 2FA-based Schemes in the Face of Active Concurrent Attacks from User Terminal . In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23), May 29-June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3558482.3590183

WiSec '23, May 29-June 1, 2023, Guildford, United Kingdom

ACM ISBN 978-1-4503-9859-6/23/05…$15.00
https://doi.org/10.1145/3558482.3590183

1 INTRODUCTION

Authentication is the process of verifying a user's identity in order to grant access to a sensitive resource. The user must present their identification credential to the verifying party, who will then grant authorization. Authentication schemes can use a variety of credentials, including passwords, one-time PINs, and fingerprints, which all involve the three factors of authentication: *"something the user knows"*, *"something the user has"*. and *"something the user is"*.

Authentication systems can be designed as single-factor, multi-factor, and passwordless systems. Password-only single-factor authentication is the most popular and widely used authentication scheme, having known security concerns. Researchers pointed out several security problems of password-only authentication [26, 38, 59, 61, 70]. They have also indicated usability problems, especially secure password management of multiple accounts [6, 9, 30, 36, 55]. Multi-factor authentication comes into the scene to address security concerns related to password-only authentication.

Two-Factor Authentication (2FA) is a type of multi-factor authentication that requires the use of a password along with another factor for authentication. Commonly used 2FA methods include One-Time Pin (OTP) [19], push notification authentication [49], and security keys [31]. While 2FA provides an added layer of security, there have been concerns raised about its security vulnerabilities [2, 15, 27, 28, 39, 50]. Additionally, some studies have shown that the added step of using 2FA can impact the user experience and usability [1, 11, 13, 14, 29, 43, 46, 47, 65, 66]. As a result, the search for a more secure and usable authentication system continues, with alternative single-factor, multi-factor, and passwordless methods being introduced in recent years.

However, adversaries use various attack methods to compromise these secure authentication systems, such as man-in-the-middle attacks [8], session hijacking [21], social engineering [67], phishing [62], and duplicating One-Time Pins or compromising 2FA devices. Researchers are working to protect against these threats and identify common security problems [7, 15, 27] in 2FA deployments. Their goal is to build a robust 2FA system that can defend against these attacks. To achieve this, it is important to address the security issues and vulnerabilities in future 2FA deployments.

In this work, we focus on two key vulnerabilities in the authentication system: "Malware on Terminal" and "Active Concurrent Attack", which refer to situations where an attacker is able to access user's terminals and launch an attack in real-time. If the user terminal is compromised, it becomes easier for the attacker to obtain the user's password. Multi-factor authentication systems are designed to provide an extra layer of security in that case, and it is generally believed that in order to compromise a 2FA-based system,

Table 1: Advantages of CSI attack over other known attacks

Attack Feature	CSI	Session Hijacking	Man-in-the-Machine
Fresh independent session	✓	✗	✓
Full control on attacker's session	✓	✗	✓
Stealthy	✓	✓	✗
Cross service attack	✓	✗	✗
Limited activity on terminal	✓	✗	✗

✓– The attack supports this feature. ✗– The attack does not support this feature.

an attacker would need to compromise both the user's terminal and the 2FA device, as described by Bonneau et al. [5].

Based on the above discussion, the research question we aim to answer is **whether secure academic authentication schemes can provide sufficient security compared to password-only authentication when the user's terminal is compromised**. We will address this question through a systematic analysis of relevant factors and data from previous literature.

User terminal malware infections can result in session hijacking attacks, which allow attackers to modify user requests during an ongoing session. However, it is important to note that such attacks are not dependent on the use of an extra factor of authentication and are therefore irrelevant to the threat model of 2FA. Furthermore, according to Bellare-Rogaway [4], session hijacking is considered a relay attack and is not a valid attack against any cryptographic protocol. Therefore, while session hijacking is a powerful and unavoidable attack, it is not a significant concern in the context of evaluating the security of 2FA or other authentication schemes that use additional factors beyond just passwords.

Another type of attack that can compromise the security of a 2FA system is the Man-in-the-Machine attack [7]. In this attack, the attacker intercepts the service response during authentication and sends it to a USB device. When the user inserts a 2FA device (such as a security key) and presses the button to approve the request, the attacker's previous request is approved instead of the user's request. Such attacks illustrate the need for further research and innovation in the field of 2FA security.

Cybercriminals have developed sophisticated malware to gain access to users' sensitive accounts, such as banking accounts. They infect users' terminals by sending malicious links through email or tricking them into installing malicious software. After that, they can launch a concurrent attack when the user intends to start an activity on the target account, such as initiating a money transfer using online banking. Recently, security experts uncovered a similar attack that caused significant financial losses in many organizations [58]. The well-known Zeus malware [12] uses a similar technique to target online banking accounts. However, they need to compromise 2FA devices separately to defeat 2FA schemes.

.

Our Work: Systematization of Academic Authentication Systems: The objective of this paper is to evaluate academic secure authentication schemes proposed over the last 15 years by systematizing their threat models and workflows. Specifically, we examine a practical attack framework known as ***Concurrent Session Injection (CSI)***, which allows for active concurrent attacks to be generated from the user terminal. We analyze the resilience of all authentication schemes in the presence of this threat and identify potential vulnerabilities. Notably, we find that CSI can compromise even systems designed to address malware and concurrent attacks.

Our Contributions: Our contributions to this work are three-fold:

(1) ***Systematization of Threat Models and Workflows of Wide range of Authentication Schemes***: In this work, we analyze Academic Authentication Systems proposed over the last 15 years, systematically examining their threat models and workflows. The studied schemes include both multi-factor and passwordless options and incorporate diverse devices such as smartwatches, smartphones, and BLE devices, as well as various verification methods such as automated and human-assisted verification. The study also evaluates authentication systems with trusted execution environments implemented on the user terminal and 2FA devices. Overall, this work provides a comprehensive evaluation of academic authentication schemes, enabling a better understanding of their effectiveness and potential vulnerabilities.

(2) ***Detailed Evaluation of potential Vulnerable Features in the presence of CSI Attack***: We investigate the effectiveness of an active concurrent attack framework *CSI*, which can launch a concurrent attack from a user terminal. We also identify potentially vulnerable features in existing authentication schemes that can serve as entry points for *CSI* attacks. To achieve this, we perform a thorough evaluation of various authentication schemes and analyze their workflows to identify potential weaknesses that can be exploited by *CSI*. Additionally, we consider authentication schemes that actively address CSI-related threats, such as malware in the terminal and concurrent attacks, and evaluate their effectiveness in protecting against these threats.

(3) ***Implementation of proof-of-concept CSI attack***: To demonstrate the feasibility of the *CSI* attack, we implemented a proof-of-concept attack. Given the diverse range of authentication schemes we evaluated, it is neither practical nor necessary to implement a prototype for each system and demonstrate attacks on them. Instead, we focus on demonstrating a typical and fundamental attack workflow using a common attack prototype. Throughout our analysis, we logically and analytically explain how the attack can be used to defeat each authentication scheme, taking into account the specific workflow and methodology used by each system. By doing so, we aim to provide insights into the fundamental weaknesses in existing authentication schemes that can be exploited by attackers, and highlight the need for more robust security measures to counteract *CSI* attacks.

Attack Demonstration: We demonstrate the fundamental attack workflow at: https://sites.google.com/view/csi-attack-demo/home

2 PRIMAR ON STUDIED SCHEMES

2.1 Evaluation Criteria

The majority of the authentication systems we studied involved a secondary device, such as a mobile phone, in their workflow. To evaluate these systems, we analyzed their workflows and extracted key characteristics that we deemed relevant to the security of the authentication process. In the later sections of this paper, we use

Table 2: Evaluation of "2FA - Interacting With User Terminal" schemes

Scheme/Author's Name	Interaction With Terminal	Method of Interaction	Secured Input In Terminal	Terminal Verification	Method of Terminal Verification	Device Verification	Method of Device Verification
Aloul et al. [3]	✓	OTP	No	✗	N/A	✓	phone needs to enrolled
WebOTP [18]	✓	OTP	No	✗	N/A	✓	phone needs to be authenticated
BrightPass [40]	✓	OTP	Yes, OTP typing in virtual keyboard	✗	N/A	✗	N/A
Khan et al. [25]	✓	OTP	No	✗	N/A	✓	IMEI, IMSI, UUID
Cheng et al. [10]	✓	OTP	No	✗	N/A	✓	phone needs to be enrolled
TrustOTP [56]	✓	OTP	No	✗	N/A	✓	phone needs to be enrolled

✓- Yes, ✗- No

these characteristics as evaluation criteria to assess the effectiveness of each authentication system in protecting against *CSI* attacks.

Credential Typing in Terminal: Most of the authentication systems we studied require users to enter authentication credentials, such as usernames, passwords, and one-time passwords (OTPs), into the user terminal (e.g., laptop).

Terminal Verification: Some 2FA and passwordless schemes utilize techniques to verify the authentication terminal, such as QR codes or displaying the same PIN on both the terminal and the 2FA device. Throughout this paper, we refer to this process as "Terminal Verification" to denote this important step.

2FA device verification: 2FA and passwordless schemes sometimes verify if the user is interacting using pre-registered device using various techniques, such as verifying unique code. Throughout the paper, we denote it as "2FA Device Verification".

2.2 Authentication Schemes

2FA - Interacting With User Terminal: This category of authentication systems requires users to interact (e.g., type a One Time PIN) with the user terminal in the 2FA authentication workflow. We present these authentication system's characteristics according to our evaluation criteria in Table 2. We notice that almost all of them require typing OTP in the terminal using the terminal's primary input method (e.g., keyboard) except Brightpass [40], which provides a virtual keyboard to type the OTP. The main difference between these systems lies in their OTP generation algorithms.

Cheng et al. [10] used "Rubbing Encryption Algorithm" to encrypt OTP before communicating it with the user. The user has to use a hardware token to decrypt the encrypted OTP. The work only ensures the security of the delivery channel of OTP. Khan et al. [25] have used both hardware features (IMEI, IMSI) and software features (UUID, android ID) to generate a unique OTP after successfully registering the mobile device.

Brightpass [40] generates OTPs by combining a known PIN (known to user) with a temporary PIN (sent by service). OTPs are only visible in mobile phone browsers using WebOTP [18]. TrustOTP [56] generates OTPs in a Trusted Execution Environment (TEE) [48] of the smartphone, providing protection against mobile malware and unwanted programs. In contrast, Cheng et al. [10] encrypt the OTP using the "Rubbing Encryption Algorithm" and require the user to use a hardware token to decrypt it.

None of these schemes use terminal verification. Khan et al. [25] use the registered device's unique features, such as IMEI and IMSI, to construct OTP, which helps the service to verify the device. For other schemes, services verify the 2FA device from enrollment information. The summary of characteristics of these authentication systems is presented in Table 2.

User Assisted Verification Schemes: These authentication systems rely on multi-factor authentication systems that include one or more secondary devices (e.g., phones, BLE devices). It verifies the terminal and 2FA device with the user's assistance (e.g., unique ID comparison, QR code scan, visual inspection) in their authentication workflow. The summary of primary characteristics of these authentication systems is listed in Table 4.

Several authentication schemes, including MP-Auth [34] and oPass [57], incorporate alternative password collection systems to protect against keylogger-based malware on the user's terminal. These systems generally initiate the authentication process from the user's terminal and collect passwords from a secondary device, such as a mobile phone. During possession factor verification, none of these schemes interact with the authentication terminal, except for imageOTP [32]. With imageOTP, the user inputs the OTP extracted from an image instead of using a username and password.

Most of these authentication schemes require user assistance for both terminal and 2FA device (e.g., smartphone) verification. Terminal verification is typically accomplished through methods such as QR code collection (2FMA-Netbank [45]), unique ID verification (MP-Auth [34]), and image matching (2FIM [32], imageOTP [16]). 2FA device verification is achieved through various means, including phone number verification (oPass [57], 2FIM [32], 2FMA-Netbank [45], SV-2FA [17]), IMEI verification (2FMA-Netbank [45]), unique user ID verification (2FIM [32]), and HTML cookie-based device identification (Device-aware 2FA [22])).

These authentication schemes require user participation to complete the authentication process. Some require the user to enter passwords on their mobile device [34, 57], while others rely on the user to click on a specific image [16, 32]. The 2FMA-Netbank [45] scheme uses a QR code displayed on the user terminal to scan signed and encrypted random values. Device-aware 2FA [22] verifies the 2FA device and user's identity by having the user click on a link sent by the service. In SV-2FA [17], the user calls a One-Time Number and verifies their voice to complete authentication.

Automated Verification Schemes: These authentication schemes rely on environment features, such as ambient audio, or establish a secure channel, such as a secure Bluetooth connection (as in 2FA-PP [60]), to verify the terminal and 2FA device. In most cases, users only need to provide their username and password (except for QuickAuth [69], which does not require a password), and the terminal and device take care of the rest when they are in proximity.

Table 3: Evaluation of "User-assisted Verification" Schemes

Scheme/Author's Name	Interaction With Terminal	Method of Interaction	Secured Input In Terminal	Terminal Verification	Method of Terminal Verification	Device Verification	Method of Device Verification
MP-Auth [34]	✗	N/A	No. Collect password from mobile device	✓	User assisted ID verification	✗	N/A
oPass [57]	✗	N/A	No. Collect password from mobile device	✗	N/A	✓	verified by phone number
2FIM [32]	✗	N/A	No	✓	Image matching with smartphone app	✓	identified by User ID or phone number
ImageOTP [16]	✓	OTP	Username and password not required	✓	Image matching with smartphone app	✓	identified by phone number
2FMA-NetBank [45]	✗	N/A	No	✓	QR Code	✓	identified by smartphone's IMEI
SV-2FA [17]	✗	N/A	No	✗	N/A	✓	identified by phone number
Device-aware 2FA [22]	✗	N/A	No	✗	N/A	✓	Identified by device information in HTML Cookies

✓- Yes, ✗- No

Almost all of them verified terminal or 2FA devices automatically and without the user's active involvement. Sound-proof [24] verifies the terminal by comparing ambient sound from the environment. However, it can be vulnerable to an environment-imitating attack, which is later addressed by this work [53]. To address this attack, similar zero-effort (i.e., no user involvement in verifying possession factor) authentication systems (e.g., Ambient Audio Authentication, Luo et al. [33], Listening Watch [54], T2FA [68], QuickAuth [69]) uses encrypted audio as comparison factor of ambient sounds.

These authentication systems are also emphasized on 2FA device (e.g., smartphones, smartwatches) verification. In most of the schemes (Wi-Auth [51], SoundAuth [64], Sound-Proof [24], and 2FA-PP [60]), the 2FA device is required to be pre-enrolled and can be identified by phone or application-specific public key. Watermelon 2FA [37] also required the 2FA device to be pre-enrolled for push notification. In addition to pre-enrollment, T2FA [68] verifies the device by Physical Unclonable Function (PUF). QuickAuth [69] server checks if the user is logged into the smartphone application.

In addition to using ambient or encoded ambient sounds, these authentication systems use near-ultrasound [64], inaudible OFDM modulated acoustic signals [20], and fine-grained Channel State Information (CSI) (Wi-Auth [51]) to verify terminal and 2FA devices and complete authentication. One of them also proposed a passwordless authentication system (QuickAuth [69]) as the 2FA device already logged in with the user's credentials.

2.3 Threat Model Analysis

2FA - Interacting With User Terminal: From Table 5, we can see that most of the authentication schemes in this group only support password theft/leakage (Aloul et al. [3], BrightPass [40], Khan et al. [25], TrustOTP [56]). As these schemes are primarily secure OTP generation schemes that users need to enter on the user terminal, they facilitate an extra layer over the passwords. WebOTP [18] also claimed protection against phishing attacks in their scheme. In addition to that, the scheme proposed by Cheng et al. [10] provides protection against a specific kind of man-in-the-middle attack (i.e., man-in-the-middle seed attack). As discussed earlier, TrustOTP [56] generates the OTP in secure hardware (i.e., Trusted Execution Environment (TEE)). As such, they consider mobile OS compromise in their threat model, in addition to password leaks.

User-assisted Verification Schemes: As previously mentioned, these authentication systems require user interaction to verify the identity of the user terminal and 2FA device. While most of the schemes include a human-in-the-loop in their verification procedure, they also claim to provide protection against various types of attacks such as phishing (MP-Auth [34], oPass [57], ImageOTP [16], SV-2FA [17]) and social engineering (ImageOTP [16], Device-aware 2FA [22]). Especially, three of the schemes (MP-Auth [34], oPass [57], SV-2FA [17]) in this category include malware as part of their attacker model, taking into account the possibility of malware on the user terminal. MP-Auth [34] and oPass [57] require users to enter their passwords using 2FA devices (such as smartphones) which can help protect against terminal malware (e.g., keyloggers).

Two of the schemes we reviewed, oPass [57], and SV-2FA [17], consider the possibility of compromising the 2FA devices in their threat model. SV-2FA, for instance, requires the user to call a one-time phone number where their voice is verified. Even if an attacker compromises the device, the assumption is that they cannot replicate the user's voice. oPass, on the other hand, requires a long-term password on the phone, which the authors suggest could be a protective measure in case of device theft or compromise. However, MP-Auth [34] excludes 2FA device (i.e., smartphone) theft or compromise from their threat model. SV-2FA [17] considers man-in-the-middle attacks as a potential threat but argues that their use of two different communication paths (e.g., SMS) makes the system resistant to such attacks. Another scheme, 2 Factor Image Matching (2FIM) [32], claims to protect against concurrent attacks by generating different images as challenges for each session.

Automated Verification Schemes: Authentication systems that rely on the proximity of 2FA devices require that these devices are not compromised or stolen to ensure security. As such, most of the authentication systems in this category that we studied (such as Proximity-proof [20], 2FA-PP [60], Sound-Proof [24], SoundAuth [64], Wi-Auth [51], Listening Watch [54], QuickAuth [69]) did not include 2FA device compromise/theft in their threat model. Other schemes did not specify whether or not they considered device compromise in their attacker model.

As the user terminals are also involved in the automatic verifications, some of the schemes in this category (Sound-Proof [24], SoundAuth [64], Listening Watch [54]) exclude malware in the

Table 4: Evaluation of "Automated Verification" Schemes

Scheme/Author's Name	Interaction With Terminal	Method of Interaction	Secured Input In Terminal	Terminal Verification	Method of Terminal Verification	Device Verification	Method of Device Verification
Proximity-proof [20]	✗	N/A	No	✓	Inaudible OFDM-modulated acoustic signal	✓	pre-enrollment of 2FA devices
2FA-PP [60]	✗	N/A	No	✓	Bluetooth MAC address and timed challenge	✓	pre-enrollment of the device which requires generating asymmetric key pairs for any communication
Sound-Proof [24]	✗	N/A	No	✓	Similarity score from ambient sound	✓	pre-enrollment which saves public key of phone application
Watermelon 2FA [37]	✗	N/A	No	✓	Browser recorded audio played by phone	✓	push notification pre-enrollment
SoundAuth, Wang et al. [64]	✗	N/A	No	✗	N/A	✓	identified by pre-enrolled smartphone's public key
Ambient Audio Authentication, Luo et al. [33]	✗	N/A	No	✓	BER comparison with the phone's recorded sound	✓	BER comparison with terminal's recorded sound
Wi-Auth [51]	✗	N/A	No	✓	CSI (Channel State Information) similarity check	✓	identified by pre-enrolled phone's public key
Listening Watch [54]	✗	N/A	No	✓	by decoding recorded encoded audio	✓	phone pre-enrolled for push notification and smartwatch paired with phone
T2FA, Zhang et al. [68]	✗	N/A	No	✓	by comparing terminal's recorded ambient sound	✓	By comparing physical unclonable function (PUF)
QuickAuth, Zhu et al. [69]	✗	N/A	No	✓	By comparing encrypted ambient sound with registered phone	✓	validating that user is logged in the smartphone

✓- Yes, ✗- No

terminal from their threat models. However, 2FA-PP [60] included malware in the user terminal in their attacker model and claimed to ensure security even in the presence of sniffing malware (e.g., keyloggers). They did not include MitB (Man-in-the-Browser) attacks as they claimed that they could hijack the user's session even if the authentication system is secure.

These schemes provide security in case of password leakage since they enable automatic validation of possession factor devices. Therefore, most of them include password leakage in their threat model. However, QuickAuth [69] does not require passwords in their authentication workflow. Some schemes, such as Watermelon 2FA [37] and Ambient Audio Authentication [33], do not explicitly address password leakage in their attacker model.

Automatic verification requires the 2FA device to be in proximity to the user terminal to establish a secure channel or compare ambient environment features, leaving these authentication systems vulnerable to co-located attacks where a nearby attacker can impersonate users. Most systems (Proximity-proof [20], Ambient Audio Authentication [33], Wi-Auth [51], Listening Watch [54], QuickAuth [69]) in this category include co-located attacks in their threat model. Sound-Proof [24] does not consider this threat, and others do not mention it. As the 2FA device needs to be nearby, remote attacks would not be successful in most cases. SoundAuth [64], Wi-Auth [51], and Listening Watch [54] include remote attacks in their threat model.

Many authentication systems rely on environmental features (such as ambient sound recording and comparison), making them vulnerable to environment imitation attacks. While earlier schemes like Sound-Proof [24] did not address this risk, newer schemes like Listening Watch [54], and QuickAuth [69] collect encrypted ambient sounds, and SoundAuth [64] uses ultra-sound to mitigate this threat. These schemes also explicitly consider environment imitation attacks in their threat models.

2.4 Overall Analysis

Analysis of the threat models in Table 5 reveals that the majority of schemes considered prevalent cyber attacks such as phishing and password leakage. Automated verification schemes were particularly addressed co-located attacks but generally did not include 2FA device theft or compromise in their threat model. Meanwhile, only a small number of authentication schemes addressed man-in-the-middle attacks in their attacker model.

Several authentication schemes (including MP-Auth [34], oPass [57], SV-2FA [17], and 2FA-PP [60]) included malware on the user terminal in their threat model, while other schemes (such as Sound-Proof [24], SoundAuth [64], and Listening Watch [54]) excluded it. Concurrent attacks were considered in only one of the schemes (2FIM [32]), while others did not mention this threat. Later in the paper, we will demonstrate how our concurrent attacks framework *CSI* can exploit vulnerabilities in these authentication schemes.

3 THE CSI ATTACK FRAMEWORK

3.1 Threat Model

Attackers have the ability to copy and run a malicious program with keylogger and hidden browser session invocation capabilities in the user terminal (no installation or administrator privilege required). They can also deceive the users into installing a benign extension on the browser containing malicious code. Using these attack components, the attacker can block or redirect any URL to a malicious site. It is important to note that this attack framework works only on the user terminal and does not compromise or control any external device or service.

Table 5: Threat Model Analysis

Threat	2FA- Interacting with Terminal						User-assisted Verification Schemes							Automated Verification Schemes								
	Aloul et al. [3]	WebOTP [18]	BrightPass [40]	Khan et al. [25]	Cheng et al. [10]	TrustOTP [56]	MP-Auth [34]	oPass [57]	2FIM [32]	ImageOTP [16]	2FMA-NetBank [45]	SV-2FA [17]	Device-aware 2FA [22]	Proximity-proof [20]	2FA-PP [60]	Sound-Proof [24]	Watermelon 2FA [37]	SoundAuth [64]	Luo et al. [33]	Wi-Auth [51]	Listening Watch [54]	T2FA [68]	QuickAuth [69]
Man on the Middle Attack	-	-	-	-	●	-	-	-	-	-	-	●	-	●	-	○	-	-	-	○	-	-	-
Co-located Attack	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	○	-	-	●	●	●	-	●
Phishing	-	●	-	-	-	-	●	●	-	●	-	●	-	○	●	○	-	-	-	-	-	-	-
Password Leak	●	-	●	●	-	●	-	●	-	-	●	-	-	●	●	●	-	●	-	●	●	●	●
Environment Imitating	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	-	-	●	-
Remote Attack	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	●	●	-	-
Malware on Terminal	-	-	-	-	-	-	●	●	-	-	-	●	-	-	●	○	-	○	-	-	○	-	-
Compromising 2FA device	-	-	-	-	-	●	○	●	-	-	-	●	-	○	○	○	-	○	-	○	○	-	○
Social Engineering	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-	-	-	-	-	-
Concurrent Attack	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-

●- Claimed to protected by scheme. ○- Threat model excludes this attack. - - Not mentioned

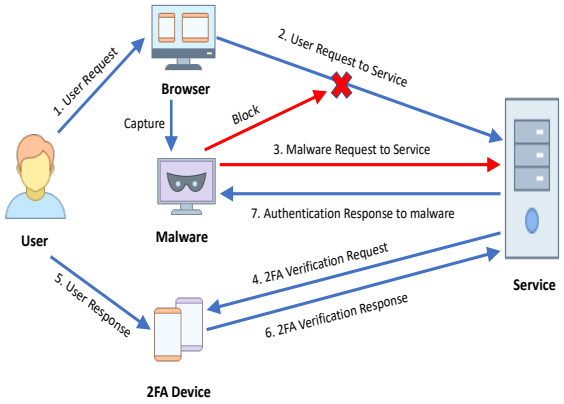

Figure 1: A step-by-step workflow of CSI attack on 2FA

3.2 Attack Framework

Fundamental capabilities of *CSI* attack can be described from two phases of authentication systems.

3.2.1 ***Initialization Phase:*** *CSI* has three significant capabilities during authentication initialization phase.

1. Keylogging: *CSI* can record any keypress of the user to steal authentication credentials (e.g., username, password). It can also identify the attack start time when the targeted user starts the authentication process.

2. Active Concurrent Attack: The *CSI* framework enables an Active Concurrent Attack that uses a hidden browser session to intercept authentication credentials and steal sensitive data during the user's authentication attempt. The browser session operates invisibly in the background and possesses nearly all the capabilities of a regular browser. Attackers can use previously obtained authentication credentials to compromise user accounts.

3. Browser Redirection and Request Block: Another capability of *CSI* is it can block the user's request and immediately redirect them to a similar-looking altered site temporarily.

3.2.2 ***Terminal Verification Phase:*** *CSI* has two key capabilities in the Terminal Verification Phase.

1. Keylogging: The *CSI* framework can intercept and capture any input provided by the user on the verification page of the authentication process, such as an OTP code. This captured information can then be used to complete the attacker's authentication attempt.

2. User View Manipulation: *CSI* can deceive users by manipulating their view of the screen, which can include fake pages or image overlays. This user view manipulation can be used to conceal information and trick the user into performing actions that benefit the attacker, such as approving notifications or providing a PIN.

An overview of a sample attack is illustrated in Figure 1.

3.3 CSI vs. Other Known Attacks

CSI is a distinct type of attack that differs from other known attacks. While a *Session Hijacking* attack is a man-in-the-middle attack that modifies transactions during an existing user session, *CSI* can launch independent concurrent session that can cause even more damage. Unlike active phishing attacks, *CSI* doesn't collect cookies and session parameters.

According to Bellare and Rogaway [4], the security of a cryptographic protocol could be compromised if an adversary's attempt is considered a trusted attempt rather than a relay attack. In the case of session hijacking, since it is a relay attack, it is not a valid attack to be considered during the security analysis of a protocol. However, in contrast, the *CSI* attack sends an independent request that is considered to be a trusted attempt by authentication services, making it a significant threat to the security of the system.

Another known attack is the *Man-in-the-Machine* attack, which captures client data objects from the browser and continuously sends a request to the USB, assuming that the malicious request will be approved first. *CSI* doesn't record any client data objects but instead sends a single concurrent request to the service using the user's authentication credentials, making it a more stealthy attack. Table 1 lists the advantages of *CSI* over session hijacking and the *Man-in-the-Machine* attack.

Table 6: Evaluation of potentially vulnerable features of studied authentication schemes

Category	Scheme Name / Author Name	Initialization		Terminal Verification					Vulnerability against CSI
		Credential Typing	Browser Involvement	Interaction with Terminal	Visual Verification	QR Code	Playing / recording audio	Other Method	
2FA - interacting with terminal	Aloul et al. [3]	✓	✓	✓	✗	✗	✗	✗	✓
	WebOTP [18]	✓	✓	✓	✗	✗	✗	✗	✓
	Brightpass [40]	✓	✓	✓	✗	✗	✗	✗	✓
	Khan et al.[25]	✓	✓	✓	✗	✗	✗	✗	✓
	cheng et al.[10]	✓	✓	✓	✗	✗	✗	✗	✓
	Trust OTP [56]	✓	✓	✓	✗	✗	✗	✗	✓
User Assisted Verification Schemes	MP-Auth [34]	✗	✓	✗	✓	✗	✗	✗	✓
	oPass [57]	✗	✓	✗	✗	✗	✗	✗	✓
	2FIM [32]	✓	✓	✗	✓	✗	✗	✗	✓
	ImageOTP [16]	✓	✓	✓	✓	✗	✗	✗	✓
	2FMA-Netbank [45]	✓	✓	✓	✗	✓	✗	✗	✓
	SV-2FA [17]	✓	✓	✗	✗	✗	✗	✓	✓
	Device-aware 2FA [22]	✓	✓	✗	✗	✗	✗	✓	✓
Automated Verification Schemes	Proximity-proof [20]	✓	✓	✗	✗	✗	✓	✗	✓
	2FA-PP [60]	✓	✓	✗	✗	✗	✗	✓	✓
	Sound-proof [24]	✓	✓	✗	✗	✗	✓	✗	✓
	Watermelon 2FA [37]	✓	✓	✗	✗	✗	✓	✗	✓
	SoundAuth [64]	✓	✓	✗	✗	✗	✓	✗	✓
	Luo et al [33]	✓	✓	✗	✗	✗	✓	✗	✓
	Wi-auth [51]	✓	✓	✗	✗	✗	✗	✓	✓
	Listening Watch [54]	✓	✓	✗	✗	✗	✓	✗	✓
	T2FA [68]	✓	✓	✗	✗	✗	✓	✗	✓
	QuickAuth [69]	✗	✓	✗	✗	✗	✓	✗	✓

✓- Yes. ✗- Not present.

3.4 Implementation of Attack Prototype

The *CSI* attack consists of three primary components:

Key and Mouse Event Monitor: We implement a keylogger and mouse event monitor to detect when the user intends to log into a target service. In addition, it can display a deceptive overlay on top of any information displayed in the browser or operating system.

Headless Browser: We utilize headless browser implementations and leverage their capabilities to design an attack that can automate user activities from the background, performing tasks that are typically performed through a standard browser.

Browser Extension: Malicious extension monitors and blocks authentication requests, redirects the user to an attacker-controlled page which displays misleading information to deceive them.

We use custom libraries to evade signature-based detection by antivirus programs. Our attack sends only one request during authentication and avoids network overload. The hidden browser session and browser extension redirect capability also make it stealthy.

4 ATTACK ANALYSIS OF STUDIED SCHEMES

4.1 Potentially Vulnerable Features

Our *CSI* attack can simultaneously capture authentication credentials, launch a hidden browser session, block the user's legitimate session, draw an overlay on the screen, and redirect the user to a tampered site. Leveraging browser-specific features via chromedriver, *CSI* can bypass 2FA by sending a single request to the device. The attack unfolds in two primary phases, where it exploits potentially vulnerable features to achieve its objectives.

Phase 1: Initialization: In this phase, we examine whether users are required to enter their authentication credentials in a terminal. If so, our analysis shows that keyloggers embedded in our *CSI* attack can easily capture these credentials for use in a future automated attack. Furthermore, if an authentication system has a web interface and is designed to be initialized by a browser, then *CSI* can attack it in two steps. First, it can block the user's legitimate request to the server. Then, it can initiate a hidden, concurrent browser session to make requests instead of the user. We have thoroughly analyzed every scheme studied and recorded the presence of these initialization features in Table 6, providing a comprehensive overview of the potential risks associated with each authentication system.

Phase 2: Terminal Verification: In comparison to traditional authentication systems, these academic systems typically incorporate a high level of security measures to protect against unauthorized access. As part of ensuring security, they use terminal verification to check if the request is generated from the user's legitimate session or the attacker's concurrent or remote session. We categorized terminal verification methods into five types: (1) *Interaction with the terminal*, where users enter a unique code, such as a one-time password; (2) *Visual Verification*, where users visually verify a unique identifier; (3) *QR Code*, where users can scan a code with a legitimate mobile app; (4) *Playing and Recording Audio*, where both the terminal and the 2FA device record environmental audio or play encrypted audio; and (5) *Other Methods*, where schemes compare other factors, such as the channel state information of Wi-Fi. To provide a summary of our findings, we have listed the terminal verification status of each system in Table 6,

4.2 Analysis of Vulnerability

2FA- Interacting with Terminal: As discussed earlier, these authentication systems primarily depend on one-time passwords to be entered in the terminal. Our studied authentication systems in this category primarily focused on secure OTP generation (e.g., TrustOTP [56]) and communication (Khan et al. [25]). However, as they have to type this OTP on the terminal, capturing by *CSI* is straightforward in this case.

We have listed the potentially vulnerable features of each scheme in Table 6, revealing that all of them allow users to enter their credentials in the terminal, involve the browser in sending authentication requests, and require users to type in their OTP, rendering them vulnerable to *CSI* attacks. It is worth noting that simply interacting

with a potentially unsafe terminal, such as typing a credential, could be enough to make a user vulnerable to attacks similar to *CSI*.

User Assisted Verification Schemes: As previously discussed, the authentication schemes in this category require active user involvement in the terminal verification process. These schemes employ various methods, such as visual inspection (MP-Auth [34], ImageOTP [16], 2FIM [32]), QR code scanning (2FMA-NetBank [45]), entering an OTP in the terminal, and other methods (e.g., SV-2FA [17] verifies using a one-time phone number). These steps in the authentication workflow help users confirm that they are approving their own session and not an attacker's.

Although some authentication schemes in this category, such as MP-Auth [34] and oPass [57], require users to enter their credentials outside the terminal, they are still susceptible to *CSI* attacks because the initial request is made through a browser. In this case, attackers do not need to provide passwords in the terminal to generate a request, which makes *CSI* attacks more straightforward.

MP-Auth [34], 2FIM [32], and ImageOTP [16] rely on users' visual inspection as part of their terminal verification process. However, these methods are vulnerable to *CSI* attacks, in which the attacker can redirect the user's browser temporarily to an altered site, causing the user to approve the attacker's request. Therefore, relying on visual inspection alone from an infected terminal is insufficient for ensuring security in these cases.

The 2FMA-Netbank system [45] can be exploited by *CSI*, which can show a QR code containing the attacker's signed random value on the user's browser screen. The user, attempting to authenticate from the same terminal and account, can decrypt the signed random value in the next step. In a later step of the scheme, the user is required to manually enter a response to the internet banking site. By doing so on an infected terminal, the user's response can be stolen and applied to a hidden session launched by the attackers.

The SV-2FA [17] can be exploited by launching a concurrent session that generates only the attacker's SMS to the user's phone during the initialization step. In the next step, the user will call the one-time phone number from the SMS generated for the attacker's session as they are expecting the SMS in their procedure, and calling them will verify their voice to complete the attack. To defeat the Device-aware 2FA [22], an attacker can generate an attacker-specific link. The user, who is expected to click on the link to be authenticated, will unknowingly approve the attacker's request.

Table 6 provides a summary of the potentially vulnerable features that can be exploited by the *CSI* attack.

Automated Verification Schemes: As discussed earlier, these 2FA schemes require minimal user involvement beyond the initial authentication credential entry. The focus of these schemes is on capturing ambient audio (e.g., Sound-Proof [24]), recording near-ultrasound (e.g., SoundAuth, Wang et al. [64]), playing encrypted audio through a terminal (e.g., Listening Watch [54]) or smartphone (e.g., Watermelon 2FA [37]). Additionally, Wi-Auth [51] utilizes Channel State Information to confirm whether the terminal and 2FA device are located in the same physical space. In another scheme, 2FA-PP [60], the authentication terminal and 2FA device establish a secure channel for mutual verification.

CSI employs a hidden browser based on chromedriver that functions in the background and shares all the capabilities of Google Chrome, including the ability to record and play audio and send it to a service. Additionally, *CSI* can launch any installed application on the user terminal. Thus, the terminal verification in the presence of *CSI* can not ensure security.

Table 7: Threat Model Analysis–Potential Defense Strategies

Threat	Potential Defense Strategies					
	Bumpy [42]	Jarecki et al. [23]	Replicate [44]	2D-2FA [52]	Varshney et al. [63]	Nyang et al. [41]
Man on the Middle Attack	-	-	-	-	●	-
Co-located Attack	-	-	-	-	-	-
Phishing	-	-	-	-	●	-
Password Leak	●	●	●	●	-	●
Environment Imitating	-	-	-	-	-	-
Remote Attack	-	-	-	-	-	-
Malware on Terminal	●	●	-	●	●	●
Compromising 2FA device	○	-	○	-	-	-
Social Engineering	-	-	-	-	-	-
Concurrent Attack	-	-	●	●	-	-

●- included, ○- excluded, - - Not mentioned

Based on Table 6, all of the authentication schemes in this group share the feature of requiring users to enter their login credentials in the user terminal, except for QuickAuth [69]. Additionally, they all rely on the browser to initiate the primary authentication request. These aspects leave the authentication systems vulnerable to *CSI*. Even passwordless schemes, such as QuickAuth, cannot defeat the attack but rather minimize the complexity for the attacker.

As discussed earlier, the *CSI* has ambient audio recording and playing ability, which makes the attack more straightforward to the attackers, as they don't have to depend on any user-assisted verification. However, as *CSI* is an active attack, it starts an attack when users are intended to authenticate. So, they are expected to keep their 2FA device (smartphone/smartwatch) nearby their terminal, which will be sufficient for the attacker to compromise these authentication systems.

Table 6 provides a summary of features that may be exploited by attackers in this category of authentication systems. Notably, it is evident that all of them are susceptible to the *CSI* attack.

5 POTENTIAL DEFENSE SCHEMES ANALYSIS

This section examines a collection of academic works that have been designed with the intention of mitigating malicious activity on user terminals, such as keylogging and concurrent attacks.

5.1 Threat Model Analysis

From Table 7, we observe that almost all of these secure authentication schemes (Bumpy [42], Jarecki et al. [23], Replicate [44], 2D-2FA [52], Nyang et al. [41]) consider malware in the terminal / total compromise of the user terminal in their threat model. Furthermore, they have asserted that their proposed schemes are resilient to any threats that may arise from such a user terminal compromise.

We observe that two of these schemes, Replicate [44] and 2D-2FA [52], consider concurrent attacks in their threat model. We will evaluate their authentication workflow with *CSI*. Additionally,

Table 8: Schemes with potential defense strategies

Scheme/Author's Name	Interaction With Terminal	Method of Interaction	Secured Input In Terminal	Terminal Verification	Method of Terminal Verification	Device Verification	Method of Device Verification
Bumpy [42]	✗	N/A	Yes, the keylogger will not work for the password field when it is deployed	✗	N/A	✗	N/A
Jarecki et al. [23]	✗	N/A	No	✓	checksum comparison by visual inspection or QR Code	✓	verified by established secure channel
Replicate [44]	✗	N/A	No	✓	Users need to do randomized action shown in the terminal screen	✓	pre-registered to the server
2D-2FA [52]	✗	N/A	No	✓	unique identifier	✓	Pre-registered phone and shared secret with the server
Varshney et al. [63]	✗	N/A	Yes, no username and password	✓	BLE device Mac Address (BT_ADDR)	✓	pre-enrollment with BT_ADDR
Nyang et al. [41]	✓	OTP	Yes, virtual keyboard while typing OTP	✓	QR Code	✗	N/A

✓- Yes, ✗- No

Bumpy [42] and Replicate [44] do not account for compromised 2FA devices or device theft in their model. Varshney et al. [63] include man-in-the-middle and phishing attacks in their model.

5.2 Discussion on Added Protection

These authentication schemes are designed to thwart keylogging attacks primarily. To address this issue, they incorporate protective measures such as Trusted Execution Environments (TEE) and virtual keyboards to safeguard input data from malicious keylogger programs that may be present in the terminal. In addition, some schemes employ innovative techniques for terminal verification to counter remote concurrent attacks.

Bumpy: McCune et al. [35] proposed a secure authentication scheme that protects sensitive inputs (such as passwords) from malicious programs in the terminal, creating a secure pathway for communicating encrypted credentials with the service. Their approach involved using a secure display (such as a phone) to show information or warnings to the user. For instance, the display could warn the user if they inadvertently provide sensitive credentials without indicating it (by failing to press "@@" before the credentials). The display also shows the domain name to which the user is authenticating. Overall, the authentication system is designed to thwart attacks by malicious entities in the user terminal and communication channel.

End-to-end Password Security: Jarecki et al. [23] proposed an end-to-end password security protocol that establishes a secure channel between the terminal and the two-factor authentication (2FA) device. As part of the protocol, a unique checksum is displayed on both the terminal and the 2FA device, which the user can compare and confirm to complete the authentication process.

Replicate: Prakash et al. [44] proposed a method for enhancing the security of push notification authentication through more interactive user responses. Specifically, the user is required to draw a pattern in their smartphone app that is displayed on the user terminal. This approach not only counters remote concurrent attacks but also promotes better user engagement compared to simpler "Just Tap" push notifications.

2D-2FA: Shirvanian and Agrawal [52] proposed the 2D-2FA scheme that displays an identifier (either a pattern or QR code) in the browser window after the user provides authentication credentials. The user then inputs this pattern on their registered device (e.g., smartphone) or scans the QR code shown in the browser window. The application in the registered device computes a high-entropy PIN with the received pattern or QR code and sends it to the authentication service, which then examines the PIN and authenticates the user. According to the authors, 2D-2FA is designed to be resistant to user negligence and compromised user terminals.

Varshney et al.: The authors [63] proposed a secure authentication protocol that leverages Bluetooth Low Energy (BLE) technology to prevent password exposure in potentially compromised terminals. In their protocol, the user registers a BLE device with their account, and the Bluetooth address (BT_ADDR) of the device is associated with the username in the service. During the authentication process, the user pairs the BLE device with the terminal, and the web application displays a list of paired devices in the browser. The user selects the correct BLE device, and the web application fetches the associated BT_ADDR to find the corresponding username. The service then sends a push notification to the user's pre-registered smartphone app, which communicates with the BLE device and sends the response back to the service for user authentication.

Nyang et al.: In their effort to enhance authentication security, Nyang et al. [41] suggested two alternative protocols. The first protocol proposes delivering a one-time password (OTP) via a QR code that only an authorized smartphone application can decode, providing protection against malicious OTP delivery channels. The second protocol presents the user with a blank virtual keyboard and a QR code to type the OTP. Only an authorized smartphone application can retrieve the keyboard layout, thereby hiding passwords from malicious entities in the terminal.

Table 9: Evaluation of protection offered by Potential Defense Strategies

Category	Scheme Name / Author Name	Initialization			Terminal Verification		Vulnerability against CSI
		Keylogging	Concurrent Attack from Terminal	Browser redirection and Request Block	Keylogging	User View Manipulation	
Potential Defense Strategies	Bumpy [42]	✓	✗	✗	✓	✓	✓[1]
	Jarecki et al. [23]	✗	✗	✗	✗	✗	✓
	Replicate [44]	✗	✗	✗	✗	✗	✓
	2D-2FA [52]	✗	✗	✗	✗	✗	✓
	Varshney et al. [63]	✓	✗	✗	✗	✗	✓[2]
	Nyang et al. [41]	✗	✗	✗	✓	✗	✓

✓- Protection Offered. ✗- No Protection.

[1] Passwordless schemes will be vulnerable to *CSI*

[2] Terminal malware can capture the BLE device information (e.g., with a screenshot) and use it in the next concurrent attack.

Table 8 shows that only one authentication scheme in this group (Nyang et al. [41]) interacts with the user terminal during the verification process. However, their use of a virtual keyboard ensures protection against malicious programs. Bumpy [35] and Varshney et al. [63] also provide protection against typing passwords in the user terminal. All other authentication systems in this group use both the user terminal and the 2FA device in their workflows, with the exception of Bumpy [35].

5.3 Evaluation in the Presence of *CSI*

To evaluate authentication schemes with potential defense strategies, we utilize the primary capabilities of *CSI* as discussed in Section 3.2. Our evaluation results are summarized in Table 9.

Although Bumpy [42] and Varshney et al. [63] protect users from keyloggers and similar malware by eliminating the need to type sensitive credentials, such as passwords, their initial request is still sent through browsers, rendering them vulnerable to concurrent attacks similar to *CSI*. *CSI* can especially defeat Bumpy in passwordless authentication schemes. Additionally, if attackers learn the BLE device used for authentication purposes through known attacks, such as shoulder surfing or screenshots taken during authentication, they can also launch concurrent *CSI* attacks on the scheme proposed by Varshney et al. [63].

Bumpy [42] not only protects users from keylogging attacks but also features a secure display, such as a smartphone, to communicate with the user. Similarly, Nyang et al. [41] offer a virtual keyboard to enter passwords. However, both protective measures can be bypassed by *CSI*, as both schemes allow sending an initial authentication request through the browser. Attackers can request an independent session using the user's ID, block the user's legitimate session, and display the attacker's virtual keyboard, QR code, or unique identifier in the display, thus defeating these schemes.

While 2D-2FA [52] and Replicate [44] claim to provide security against concurrent attacks, our evaluation reveals that they only ensure security against remote concurrent attacks initiated from outside of the user terminal. They do not offer protection against an active concurrent attack like *CSI* generated from the user terminal.

Table 9 indicates that none of the potential defense schemes can offer complete protection against active concurrent attack *CSI* that is generated from the user terminal.

6 FURTHER INSIGHTS AND FUTURE WORK

Automated Verification Schemes: From Table 5, it is evident that automated verification schemes generally do not consider malware in the terminal or concurrent attacks. They also rarely consider phishing or other social engineering attacks in their threat models. It should be considered in future works.

Concurrent Attacks: We can see from Table 5 and Table 7 that very few academic authentication schemes consider concurrent attacks in their threat models. Also, the schemes that included concurrent attacks in their threat model only considers remote concurrent attack (i.e., concurrent attacks initiated from a remote computer). No one is considering internal active concurrent attack similar to *CSI*. Researchers have the opportunity to work on it.

User-assisted Verification Scheme: Malware in the user terminal can modify the verification content (e.g., unique identifier). As such, it is not safe to use the user-assisted verification scheme from the malware-infected terminal as it cannot provide a "Secure Display". However, some of our studied authentication schemes include "Malware in Terminal" in their threat model, while others did not mention/consider it. From our studied schemes, no one excludes malware in the terminal from their threat model. Researchers who are working in user-assisted verification schemes should take it into consideration.

Keylogging Prevention is not Sufficient to Defeat *CSI*: From Table 5 and Table 7, we can see that schemes considered malware in the terminal in their threat model, did not include the active concurrent attack in their threat model. Without designing proper protection from active concurrent attacks, keylogging prevention would not be sufficient to defeat malicious programs like *CSI*, which researchers may consider as future work.

7 CONCLUSION

In this work, we systematically evaluated the security of 29 academic authentication schemes over the past 15 years against an active concurrent attack framework called *CSI*. Our analysis identified several potentially vulnerable features in these systems that could be exploited by malware similar to *CSI*. We found that almost all of the evaluated systems were susceptible to this attack. These results highlight the need for increased attention to the security of authentication systems in the face of advanced malware attacks.

ACKNOWLEDGMENTS

This work is funded in part by NSF grants: OAC-2139358, CNS-2201465 and CNS-2152669.

REFERENCES

[1] Claudia Ziegler Acemyan, Philip Kortum, Jeffrey Xiong, and Dan S Wallach. 2018. 2FA Might Be Secure, But It's Not Usable: A Summative Usability Assessment of Google's Two-factor Authentication (2FA) Methods. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 62. SAGE Publications Sage CA: Los Angeles, CA, 1141–1145.

[2] Manal Adham, Amir Azodi, Yvo Desmedt, and Ioannis Karaolis. 2013. How to attack two-factor authentication internet banking. In *International conference on financial cryptography and data security*. Springer, 322–328.

[3] Fadi Aloul, Syed Zahidi, and Wassim El-Hajj. 2009. Two factor authentication using mobile phones. In *2009 IEEE/ACS International Conference on Computer Systems and Applications*. IEEE, 641–644.

[4] Mihir Bellare and Phillip Rogaway. 1993. Entity authentication and key distribution. In *Annual international cryptology conference*. Springer, 232–249.

[5] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 553–567.

[6] Kay Bryant, John Campbell, et al. 2006. User behaviours associated with password security and management. *Australasian Journal of Information Systems* 14, 1 (2006).

[7] Thanh Bui, Siddharth Prakash Rao, Markku Antikainen, Viswanathan Manihatty Bojan, and Tuomas Aura. 2018. Man-in-the-machine: exploiting ill-secured communication inside the computer. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 1511–1525.

[8] Franco Callegati, Walter Cerroni, and Marco Ramilli. 2009. Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security & Privacy* 7, 1 (2009), 78–81.

[9] Danuvasin Charoen, Murali Raman, and Lorne Olfman. 2008. Improving end user behaviour in password utilization: An action research initiative. *Systemic Practice and Action Research* 21, 1 (2008), 55–72.

[10] Fred Cheng. 2011. Security attack safe mobile and cloud-based one-time password tokens using rubbing encryption algorithm. *Mobile Networks and Applications* 16, 3 (2011), 304–336.

[11] Stéphane Ciolino, Simon Parkin, and Paul Dunphy. 2019. Of Two Minds about Two-Factor: Understanding Everyday {FIDO} U2F Usability through Device Comparison and Experience Sampling. In *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*.

[12] Comodo Group Inc. 2020. What is Zeus Trojan ? | How the Zeus Virus infects the computers? https://enterprise.comodo.com/blog/what-is-zeus-trojan/.

[13] Sanchari Das, Andrew Dingman, and L Jean Camp. 2018. Why Johnny doesn't use two factor a two-phase usability study of the FIDO U2F security key. In *International Conference on Financial Cryptography and Data Security*. Springer, 160–179.

[14] Emiliano De Cristofaro, Honglu Du, Julien Freudiger, and Greg Norcie. 2013. A comparative usability study of two-factor authentication. *arXiv preprint arXiv:1309.5344* (2013).

[15] Alexandra Dmitrienko, Christopher Liebchen, Christian Rossow, and Ahmad-Reza Sadeghi. 2014. SECURITY ANALYSIS OF MOBILE TWO-FACTOR AUTHENTICATION SCHEMES. *Intel Technology Journal* 18, 4 (2014).

[16] Chris Drake and Praveen Gauravaram. 2018. Designing a User-Experience-First, Privacy-Respectful, high-security mutual-multifactor authentication solution. In *International Symposium on Security in Computing and Communication*. Springer, 183–210.

[17] Haruhiko Fujii and Yukio Tsuruoka. 2013. SV-2FA: Two-factor user authentication with SMS and voiceprint challenge response. In *8Th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*. IEEE, 283–287.

[18] Zhi Guan, Hu Xiong, Suke Li, and Zhong Chen. 2011. Mobile Browser as a Second Factor for Web Authentication. In *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*. IEEE, 276–281.

[19] Neil M Haller. 1994. The s/key (tm) one-time password system. In *Symposium on Network and Distributed System Security*. 151–157.

[20] Dianqi Han, Yimin Chen, Tao Li, Rui Zhang, Yaochao Zhang, and Terri Hedgpeth. 2018. Proximity-proof: Secure and usable mobile two-factor authentication. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 401–415.

[21] Brendon Harris and Ray Hunt. 1999. TCP/IP security threats and attack methods. *Computer communications* 22, 10 (1999), 885–897.

[22] Markus Jakobsson. 2020. Social Engineering Resistant 2FA. *arXiv preprint arXiv:2001.06075* (2020).

[23] Stanislaw Jarecki, Hugo Krawczyk, Maliheh Shirvanian, and Nitesh Saxena. 2018. Two-factor authentication with end-to-end password security. In *IACR International Workshop on Public Key Cryptography*. Springer, 431–461.

[24] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-proof: usable two-factor authentication based on ambient sound. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*. 483–498.

[25] Burhan Ul Islam Khan, Rashidah F Olanrewaju, Farhat Anwar, and Mashkuri Yaacob. 2018. Offline OTP based solution for secure internet banking access. In *2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*. IEEE, 167–172.

[26] Daniel V Klein. 1990. Foiling the cracker: A survey of, and improvements to, password security. In *Proceedings of the 2nd USENIX Security Workshop*. 5–14.

[27] Radhesh Krishnan Konoth, Victor van der Veen, and Herbert Bos. 2016. How anywhere computing just killed your phone-based two-factor authentication. In *International Conference on Financial Cryptography and Data Security*. Springer, 405–421.

[28] Laurens Koot. 2012. Security of mobile TAN on smartphones. *A risk analysis for the iOS and Android smartphone platforms. Master's thesis, Radboud University Nijmegen* (2012).

[29] Kat Krol, Eleni Philippou, Emiliano De Cristofaro, and M Angela Sasse. 2015. "They brought in the horrible key ring thing!" Analysing the Usability of Two-Factor Authentication in UK Online Banking. *arXiv preprint arXiv:1501.04434* (2015).

[30] Naveen Kumar. 2011. Password in practice: An usability survey. *Journal of Global Research in Computer Science* 2, 5 (2011), 107–112.

[31] Juan Lang, Alexei Czeskis, Dirk Balfanz, Marius Schilder, and Sampath Srinivas. 2016. Security keys: Practical cryptographic second factors for the modern web. In *International Conference on Financial Cryptography and Data Security*. Springer, 422–440.

[32] H Karen Lu, Asad Ali, Benoit Famechon, and Najam Siddiqui. 2019. Out-of-Band Authentication Using 2-Factor Image Matching. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer . . . , 132–140.

[33] Jia-Ning Luo, Meng-Hsuan Tsai, Nai-Wei Lo, Chih-Yang Kao, and Ming-Hour Yang. 2019. Ambient audio authentication. *Mathematical biosciences and engineering: MBE* 16, 6 (2019), 6562–6586.

[34] Mohammad Mannan and Paul C Van Oorschot. 2007. Using a personal device to strengthen password authentication from an untrusted computer. In *International Conference on Financial Cryptography and Data Security*. Springer, 88–103.

[35] Jonathan M McCune. 2009. Safe passage for passwords and other sensitive data. In *Proceedings of the Network and Distributed System Security Symposium, 2009*.

[36] B Dawn Medlin and Joseph A Cazier. 2007. An empirical investigation: Health care employee passwords and their crack times in relationship to hipaa security standards. *International Journal of Healthcare Information Systems and Informatics (IJHISI)* 2, 3 (2007), 39–48.

[37] Joshua Meier, Jesse Zhang, Richard Zou, and James Mickens. 2017. Zero-effort Two-factor Authentication using Audio Signals. In *2017 International Symposium on Cyber Security Cryptography and Machine Learning (CSCML 2017)*. 10.

[38] Robert Morris and Ken Thompson. 1979. Password security: A case history. *Commun. ACM* 22, 11 (1979), 594–597.

[39] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin, and Jean-Pierre Seifert. 2013. SMS-based one-time passwords: attacks and defense. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 150–159.

[40] R Niruban. [n. d.]. User Authentication Scheme for Security and Privacy in Social Networks Using Brightpass Method. *Journal of Current Research in Engineering and Science* ([n. d.]), 34.

[41] DaeHun Nyang, Aziz Mohaisen, and Jeonil Kang. 2014. Keylogging-resistant visual authentication protocols. *IEEE Transactions on Mobile Computing* 13, 11 (2014), 2566–2579.

[42] Jonathan M McCune Adrian Perrig and Michael K Reiter. 2009. Safe passage for passwords and other sensitive data. In *Proceeding of the 16th annual network and distributed system security Symposium*.

[43] Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, and Sotiris Ioannidis. 2015. Two-factor authentication: is the world ready? Quantifying 2FA adoption. In *Proceedings of the eighth european workshop on system security*. 1–7.

[44] Jay Prakash, Clarice Chua Qing Yu, Tanvi Ravindra Thombre, Andrei Bytes, Mohammed Jubur, Nitesh Saxena, Lucienne Blessing, Jianying Zhou, and Tony QS Quek. 2021. Countering Concurrent Login Attacks in "Just Tap" Push-based Authentication: A Redesign and Usability Evaluations. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 21–36.

[45] Andreyanto Pratama and Edit Prima. 2016. 2FMA-NetBank: A proposed two factor and mutual authentication scheme for efficient and secure internet banking. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, 1–4.

[46] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. 2019. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*.

[47] Joshua Reynolds, Trevor Smith, Ken Reese, Luke Dickinson, Scott Ruoti, and Kent Seamons. 2018. A tale of two studies: The best and worst of yubikey usability. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 872–888.

[48] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. 2015. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 57–64.

[49] Aleksey Sanin, Matt Ricketson, Ryan Newlman, Andrew LeBlanc, and Eric Stern. 2014. Systems and methods for push notification based application authentication and authorization. US Patent App. 13/915,475.

[50] Peter Schartner and Stefan Bürger. 2011. Attacking mTAN-applications like e-banking and mobile signatures. *University of Klagenfurt* (2011).
[51] Syed W Shah and Salil S Kanhere. 2017. Wi-Auth: WiFi based second factor user authentication. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 393–402.
[52] Maliheh Shirvanian and Shashank Agrawal. 2021. 2D-2FA: A New Dimension in Two-Factor Authentication. In *Annual Computer Security Applications Conference*. 482–496.
[53] Babins Shrestha, Maliheh Shirvanian, Prakash Shrestha, and Nitesh Saxena. 2016. The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 908–919.
[54] Prakash Shrestha and Nitesh Saxena. 2018. Listening Watch: Wearable Two-Factor Authentication using Speech Signals Resilient to Near-Far Attacks. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 99–110.
[55] Elizabeth Stobert and Robert Biddle. 2014. The password life cycle: user behaviour in managing passwords. In *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*. 243–255.
[56] He Sun, Kun Sun, Yuewu Wang, and Jiwu Jing. 2015. TrustOTP: Transforming smartphones into secure one-time password tokens. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 976–988.
[57] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin. 2011. oPass: A user authentication protocol resistant to password stealing and password reuse attacks. *IEEE Transactions on Information Forensics and Security* 7, 2 (2011), 651–663.
[58] Symentec. 2021. Almost 100 Organizations in Brazil Targeted with Banking Trojan. https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/banking-trojan-latam-brazil.
[59] Viktor Taneski, Marjan Heričko, and Boštjan Brumen. 2014. Password security—No change in 35 years?. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 1360–1365.
[60] Enis Ulqinaku, Daniele Lain, and Srdjan Capkun. 2019. 2FA-PP: 2nd factor phishing prevention. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 60–70.
[61] Blase Ur, Jonathan Bees, Sean M Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Do users' perceptions of password security match reality?. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3748–3760.
[62] Alta Van der Merwe, Marianne Loock, and Marek Dabrowski. 2005. Characteristics and responsibilities involved in a phishing attack. In *Proceedings of the 4th international symposium on Information and communication technologies*. 249–254.
[63] Gaurav Varshney and Manoj Misra. 2017. Push notification based login using BLE devices. In *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 479–484.
[64] Mingyue Wang, Wen-Tao Zhu, Shen Yan, and Qiongxiao Wang. 2018. SoundAuth: Secure zero-effort two-factor authentication based on audio signals. In *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–9.
[65] Catherine S Weir, Gary Douglas, Martin Carruthers, and Mervyn Jack. 2009. User perceptions of security, convenience and usability for ebanking authentication tokens. *Computers & Security* 28, 1-2 (2009), 47–62.
[66] Catherine S Weir, Gary Douglas, Tim Richardson, and Mervyn Jack. 2010. Usable security: User preferences for authentication methods in eBanking and the effects of experience. *Interacting with Computers* 22, 3 (2010), 153–164.
[67] Ira S Winkler and Brian Dealy. 1995. Information Security Technology? Don't Rely on It. A Case Study in Social Engineering.. In *USENIX Security Symposium*, Vol. 5. 1–1.
[68] Jiliang Zhang, Xiao Tan, Xiangqi Wang, Aibin Yan, and Zheng Qin. 2018. T2FA: Transparent two-factor authentication. *IEEE Access* 6 (2018), 32677–32686.
[69] Xiaoyan Zhu, Suiyu Yu, and Qingqi Pei. 2016. QuickAuth: Two-factor quick authentication based on ambient sound. In *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.
[70] Moshe Zviran and William J Haga. 1999. Password security: an empirical study. *Journal of Management Information Systems* 15, 4 (1999), 161–185.