Demo: Combating Caller ID Spoofing on 4G Phones Via CEIVE

Haotian Deng
Purdue University
deng164@purdue.edu

Chunyi Peng
Purdue University
chunyi@purdue.edu

ABSTRACT

We present the demonstration of Ceive (**C**all**e**e-only **i**nference and **ve**rification), an effective and practical defense against caller ID spoofing. CEIVE is a victim callee only solution without requiring additional infrastructure support or changes on telephony systems; It is ready to deploy and easy to use. Given an incoming call, Ceive leverages a callback session and its associated call signaling observed at the phone to infer the call state of the other party. It further compares with the anticipated call state of the incoming call, thus quickly verifying whether the incoming call comes from the originating number or not. In this demo, we demonstrate Ceive installed on Android phones combating both basic and advanced caller ID spoofing attacks.

CCS CONCEPTS

• **Security and privacy → Spoofing attacks**; • **Networks → Mobile networks**; **Signaling protocols**;

ACM Reference Format:
Haotian Deng and Chunyi Peng. 2018. Demo: Combating Caller ID Spoofing on 4G Phones Via CEIVE. In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), October 29-November 2, 2018, New Delhi, India.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3241539.3267725

1 INTRODUCTION

In this work, we present the demonstration of our Ceive solution that combats caller ID spoofing using the callee-only power [1]. Ceive is motivated by two drive forces: real user demands and lack of effective and practical solutions.

In recent years, caller ID spoofing has emerged as a simple yet menacing attack technique to telephony scams which have resulted in substantial monetary loss, sensitive data leakage and victim complaints [2, 3]. Superior to simply claiming to be the trustworthy party, a sly attacker forges the phone number of an trusted caller so that the call appears to come from the "*correct*" number of the authentic party. Upon receiving the call, the victim is deceived into believing that the call comes from the "*trustworthy*" caller indicated by the phone number (e.g., government agencies, public and utility services, banks, insurances, etc). As people are used to trusting the telephony channel, caller ID spoofing makes their scams more likely to succeed, compared to emails, messages, and other channels. As a matter of fact, imposter scam has become the No.2 source of consumer complaints in the USA according to Federal Federal Trade Commission in 2017 [2]. An estimated one in every 10 American adults lost money in a phone scam in the past 12 months with $430 loss on average (56% increase than $274 in 2016), totaling about $9.5 billion overall in 2017 [3]; Similar losses and complaints are reported in Europe, Asia, Australia and globally.

Technique-wise, caller ID spoofing is unfortunately easy to launch, but hard to defend. On the attack side, spoofing is even offered as public service such as FakeCall[4], Spoofcard[5], and many apps alike. On the defense side, existing proposals are deemed effective but not practical due to heavy deployment costs and updates on the telephony systems. They either require building a global certificate authority for end-to-end caller authentication enabling additional network assistance for caller verification or launching challenge-and-response to verify the true caller (changes required on all possible callers), to name a few. Detailed limitations of the state-of-the-art can be found in [1].

We propose **Ceive** (**C**all**e**e-only **i**nference and **ve**rification), a practical and effective solution that leverages callee-only capability to defend against caller ID spoofing [1]. Ceive explores a simple solution concept of caller verification by initiating a callback session to the originating phone number and comparing the call states of the outgoing call session with the incoming call. In the absence of caller ID spoofing, both call states belong to the same party and they should match at all the time; Otherwise, the call state mismatch will be observed at least once which indicates the verification call reaches another party different from the caller of the incoming call. Eventually, Ceive formulates the core design as an inference problem which aims to learn the callee's call state in the verification callback session using the observations at

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
MobiCom '18, October 29-November 2, 2018, New Delhi, India
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5903-0/18/10.
https://doi.org/10.1145/3241539.3267725

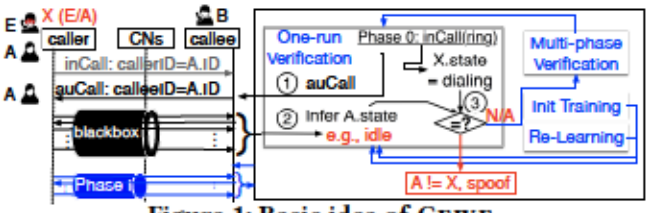

Figure 1: Basic idea of CEIVE.

the caller side. CEIVE uses an unexplored side channel of call setup signaling to infer the call state out of the sequence of observed signaling messages and enables a coarse-grained call state inference which suffices to differentiate spoof from no-spoof in most use scenarios. CEIVE further incorporates techniques such as multi-phase verification and re-learning to resolve ambiguity tailored to caller ID spoofing in certain scenarios. We implement CEIVE on rooted Android phones and validate its effectiveness in four US carriers against the caller ID spoofing from all top-tier US carriers, one landline and two small carriers. It achieves 100% accuracy in almost all spoofing cases except one specially stretched and targeted attack against CEIVE.

In this demo, we aim to show how CEIVE combats real caller ID spoofing attacks on Android phones in 4G networks. The spoofing attack will be launched in a controlled and responsible way. In particular, we and participators will use public spoofing services to launch caller ID spoofing against CEIVE-enabled victim phones provided by us. Participators are encouraged to test with advanced spoofing attacks as they can. Each victim phone uses one of two 4G voice solutions: voice-over-LTE (VoLTE) and circuit-switched fall back (CSFB), or both. It is rooted so that it can retrieve call setup signaling messages. It will prompt the verification result (spoofing or not) upon receiving a call.

2 CEIVE'S DESIGN

Figure 1 presents the core idea and design components in CEIVE. We consider an incoming call *inCall* from Alice (A) in case of no spoofing or Eve (E) who spoofs Alice's number to the callee Bob (B). X denotes the unknown caller and it can be A (no spoof) or E (spoof). The detailed design is elaborated in [1]. We briefly describe the core operations used by main components: one-run verification, multi-phase verification, initial training and re-learning.

CEIVE's core module is runtime spoofing detection when a call comes. We devise a multi-phase (mostly two-phase) verification strategy, which consists one-run verification at each phase. The initial phase starts with $\Omega_1(X) = dialing$, where $\Omega_i(X)$ represents the X's state at phase i. Namely, CEIVE asks B to dial the first *auCall* back to A's number (①) while *inCall* is ringing, infers the A's call state through the observed call setup signaling during the auCall session (②), and compares it with the X's call state based on the inCall's context (③). If mismatch, we can infer that X is not A and the inCall is spoofed. Otherwise, we may need more information to verify that (multi-phase verification adopted). Due to the attacker's manipulation and the existing uncertainties, a match is possible when $X \neq A$ (spoof). Consequently, it is to be determined (TBD) when a match is observed and not all the phases complete. Otherwise, we believe it as no-spoof* when both states match at all the phases. The modules of *initial training* and *re-learning* are to train and update decision tree rules (classifiers) used by the above spoofing verification. The former is mandatory and requires one-time effort before use. The latter is optional and can update rules with user feedbacks (labelled samples) after use.

One-run verification. Core to CEIVE is to exploit an available, yet unexplored side channel to infer call states. Our feasibility study shows that common call information provided by mobile OSes including PRECISE_CALL_STATE, PHONE_STATE in TelephonyManager and system logs fails to infer the state on the *remote* callee side, because it only provides call states on its *own* side. While the caller knows what happens at the terminating party, these high-level APIs hide internal, fine-grained call context and fail to run inference required by CEIVE. We thus exploit the sequence of call setup signaling messages (SIP for VoLTE and CC for CSFB/CS) and find that they convey enough information for call state inference. We also find that the call state inference may vary with carrier networks (e.g., the signaling sequences are different in AT&T and Verizon when A is dialing) and call technologies (VoLTE and CSFB), which can be learned through the training procedure (elaborated later). When a call comes, CEIVE runs an online algorithm to infer the state of auCall.callee as early as possible using the classifier trained in advance, and compares it with the current inCall.caller state known as a prior.

Multi-phase verification. CEIVE runs two-phase verification before and after the inCall is answered, and combines inference results of two auCalls to make multi-phase spoofing inference when single phase is not enough to handle ambiguity. This strategy not only relieves ambiguity caused by indistinguishable call states in one carrier, but also handles diversity across unknown carriers (the same pattern means different states in different carriers). This also helps to combat advanced spoofing strategy which may manipulate A's call state to cheat CEIVE (e.g., forcing A is being dialed when calling the victim B). Unless the attacker can manipulate A's states at all the phases, spoofing will be detected out as long as one mismatch is observed.

Initial Training. It takes three steps to train decision tree rules to infer the callee's state from the caller's observations. First, we design and conduct experiments to collect samples which are labeled with four callee's states: dialing, connected, idle and unavailable. The samples are call setup

signaling sequences received on the caller's side and each is associated with known experiment settings including the caller's and callee's carriers and call technologies. We then extract low-dimensional features out of the raw signaling sequences so that the feature space are greatly reduced while still retaining key information. We find that some signaling messages can differential call states while others not. Finally, we learn the signaling sequences for each distinct callee state by merging the samples within the same output label under other settings which are unknown in use. Specifically, when B running CEIVE, B has no idea about A's carrier and call technology. Consider some sequences may be observed at different call states (e.g., idle or connected under different carriers), they can be inferred to one merged label for multiple call states (here, idle-or-connected). This induces ambiguity in inference. Training will divide all the observed sequences exclusively into each unique label, which can be a single call state or a combo. Note that if a combo call state is inferred, CEIVE will determine a match (with ambiguity) by checking if X's state is included in the possible call states.

Re-Learning. CEIVE supports learning after the use. After one call, it allows the user to manually label this call so that classifier can update their rules based on user feedbacks. This makes CEIVE extensible to new settings (e.g., a new carrier which has not been studied during the initial training).

3 DEMONSTRATIONS

We offer live demonstration in multiple scenarios. Audience are encouraged to participate and play as attackers and/or victims upon their consent.

Default demo setup. A complete demonstration uses 5 cell phones: one as Alice, two Android phones as Bob and Bob's buddy, and another two as Eve and Eve's buddy. Note that only the phones used as Bob and Bob's buddy must have CEIVE installed on them and they should be rooted Android phones. We use Google Firebase service to make connections between Bob and Bob's buddy and thus they also require Internet access during the demo. Alice's phone is optional. It is used to demonstrate what will happen on the true caller when CEIVE works. We encourage to use the volunteer's phone or any number suggested by the audience. Eve and Eve's buddy are used to launch caller ID spoofing attacks. Their use depends on the attack strategy (basic or advanced). The phone used as Bob's buddy will perform CEIVE's core function of call state inference and verification and thus should be capable of making VoLTE or CSFB/CS. We demonstrate how CEIVE works under normal scenario, basic and advanced attacks, illustrated in seven scenarios (Table 1).

No caller ID spoofing. We first let A call B (C1).

	No.	Call Scenario	$\Omega_1(A)$	$\Omega_2(A)$
basic	C1	A→B	dialing	conn
	C2	E→B, A is idle	idle	idle
	C3	E→B, A is connected (on-a-call)	conn	conn
	C4	E→B, A is unavailable (i.e, A6)	off	off
advanced	C5	E→B, E (E') made A on a call	conn	conn
	C6	E→B, E (E') is dialing A too	dialed*	dialed*
	C7	E→B, E (E') first dials A and hangs up once B answers the call	dialed*	idle

Table 1: Call Scenarios. 'being dialed' and 'dialing' is indistinguishable in our state inference.

Basic caller ID spoofing. E simply makes a phone call to B, spoofing A's number, while A is in the idle state (C2), on a connected call (C3) or unavailable (C4) during the attack. Caller ID spoofing will be launched through the tool public available [4, 5]. Note that some caller ID spoofing can be provided via Internet service and E's phone is not needed.

Advanced caller ID spoofing. We next test with advanced caller ID spoofing when E exploits its attack power to manipulate A's call state. In particular, E will ask E's buddy to call Alice during the attack so that A is on a call (C5) or being dialed (C6) or varies its state based on the attack result (C7). In C7, E's buddy dials A at the same time when E makes an spoofing call to B and hangs up immediately once Bob answers the call. C7 is a designated attacks against CEIVE.

In all the scenarios, we will vary the phone numbers, carriers (if can), call technologies (if can). Note that the test option depends on carriers and call technologies provided in India. For example, India carriers may not support VoLTE, available in the US carrier. Consider we will use local carriers in India which have not been tested before, live demos may not work exactly as we have done in the US. We will provide video demos recorded for the US carriers and landlines for quick and brief exhibitions. In this demo, we aim to show CEIVE can handle the spoofing attack properly except some cases in C7 gives alert in a responsive way and remains friendly to normal users.

Acknowledgement. This work was partially supported by NSF Grants: CNS-1750953, CNS-1753500 and CNS-1749045.

REFERENCES

[1] Haotian Deng, Weicheng Wang, and Chunyi Peng. CEIVE: Combating Caller ID Spoofing on 4G Mobile Phones Via Callee-Only Inference and Verification. In *ACM International Conference on Mobile Computing and Networking*, MobiCom. ACM, 2018.
[2] Federal Trade Commission. Ftc releases annual summary of consumer complaints. https://www.ftc.gov/news-events/press-releases/2017/03/ftc-releases-annual-summary-consumer-complaints, 2017.
[3] MarketWatch. Here's how much phone scams cost americans last year... https://www.marketwatch.com/story/heres-how-much-phone-scams-cost-americans-last-year-2017-04-19, April 2017.
[4] FakeCall, 2018. Mobile app at Google Play and App Store.
[5] spoofcard. https://www.spoofcard.com/free-spoof-caller-id, Feb 2018.