A=SIT Technology
E-Government Innovation Center

27 July 2020 Study · Web & Cloud · Mobile Security · 15 min read

Analysis of Electroneum Cloud Mining

Archived. This content is no longer updated. It remains available for reference.

Source https://technology2.a-sit.at/en/publications/analysis-of-electroneum-cloud-mining/

Electroneum is one of the first cryptocurrencies that allows users to mine coins with a mobile device. Electroneum's Cloud Mining process refers to the activity of periodically rewarding users of their application with free ETN tokens that they can store or spend with ETN-accepting retailers. To create an Electroneum account and activate the cloud mining process, Electroneum requires its users to upload selfies with a predefined gesture or a drawing of a symbol. This project analyzed the device verification and authorization-related security measures employed by Electroneum. Based on the analysis, we mounted a device-emulation and app-impersonation attack that exploits Electroneum's cloud mining process. We created non-existing selfies by relying on Generative Adversarial Network (GAN) techniques to bypass the selfie requirement during the account setup. Furthermore, we employed reverse engineering to develop a bot that simulates the genuine Electroneum application and maintains an arbitrary number of illegitimate accounts on one Android device, enabling the malicious user to obtain ETN token rewards illegitimately. Fully adhering to the responsible disclosure guidelines, we submitted a vulnerability disclosure regarding our findings to the Electroneum team. After responsible disclosure, the Cloud Mining feature was closed down. Therefore the described attack is no longer applicable.

1. Introduction

Cryptocurrency mining is the term used to denote the process of verifying transactions and including them into a new created block, which extends the blockchain. The traditional mining process involves solving difficult cryptographic puzzles that require significant computational power, often provided by specialized hardware. Therefore, this mining approach is not suitable for resource-constraint devices such as smartphones, as it may severely impair their overall performance. With both Google and Apple banning applications that mine cryptocurrency in this traditional fashion from their application stores, the accessibility of average smartphone users to cryptocurrency mining and their underlying payment system is significantly restricted.

Nonetheless, there are several projects committed to making mobile cryptocurrency mining feasible. Projects like Electroneum [1], Phoneum [2], and Pi [3] introduced the concept of "cloud-mining" to denote their mobile mining process, which in turn refers to the process of rewarding users of their dedicated application with previously mined tokens. This virtual mining process does not put any strain on the mobile device, but at the same time introduces susceptibility to device or application impersonation attacks. Through such attacks, adversaries can emulate an arbitrary number of mobile devices to illegitimately "mine" cryptocurrency tokens.

This project focused on analyzing the security measures employed by the Electroneum mobile application. We mainly investigated the possibility of circumventing the device verification and authorization-related security measures to mount a device-emulation and app-impersonation attack that exploits Electroneum's cloud mining process. More specifically, we developed an application that mimics Electroneum's application and device authorization protocol to generate an arbitrary number of illegitimate accounts that take advantage of the Electroneum token rewards. Our results showed that, despite newly employed security mechanisms, the Electroneum application is still vulnerable to such attacks, and further measures need to be in place to ensure the execution of the cloud mining application on genuine devices.

Fully adhering to the responsible disclosure guidelines, we submitted a vulnerability disclosure regarding our findings to the Electroneum team. The results of this project were published in a scientific paper [4] accepted at the 17 [th] International Conference on Security and Cryptography (SECRYPT 2020).

2. Electroneum

The Electroneum "mobile-cryptocurrency" project and the underlying mobile payment system were launched in November 2017 with the unique mission of allowing anyone with a smartphone and Internet connection to access digital payments and participate in the global economy. Right from the beginning, the focus of Electroneum was to provide an easily-accessible, cryptocurrency-based mobile payment system that allows its users to store, send, and receive digital coins via their smartphone alone. This strategy proved out to be successful, allowing its user basis to grow quickly up to over 3.4 million registered users [5].

During its short history, Electroneum's proprietary blockchain model has gone through significant changes. At first, Electroneum's underlying blockchain technology was very similar to the one used by the Monero [6] and Bytecode cryptocurrencies. Initially, Electroneum adopted a Proof-of-Work algorithm called Cryptonight [7], which served the purpose of making CPU and GPU mining likely efficient. Aiming to foster their cryptocurrency's accessibility and improve the scalability and durability of their blockchain towards malicious attacks, they introduced numerous changes, which eventually converged in a *permissioned* blockchain model that was labeled as "Moderated Blockchain". As the name implies, the model introduces a second centralized blockchain layer of highly trusted nodes that "moderates" the decentralized layer of miners. They also shifted from employing Proof-of-Work to the new Proof-of-Responsibility mining paradigm, according to which the miners are required to "responsibly" maintain their payment system's integrity. This addition of highly-trusted nodes enabled Electroneum to become one of the few cryptocurrencies that are immune to the notorious 51% attack on their network.

This section goes into the details of Electroneum's cloud mining process and the security features employed by Electroneum when verifying accounts and authorizing devices.

2.1. Cloud Mining Process

Electroneum's Cloud Mining process refers to the activity of rewarding users for running their application with free ETN tokens every month. Every user of the dedicated Electroneum application can earn free ETNs tokens worth up to 3 USD per month, regardless of the mobile devices' specifications or their ability to be always connected to the Internet. To participate in the Cloud Mining process, all that a user needs to do is download and install the Electroneum app, create an Electroneum account, and activate Cloud Mining. When creating an Electroneum account, the user needs to sign up with a unique username and password (or use their Facebook account). Furthermore, the user is required to provide a mobile phone number, and select a PIN code needed when launching the app, or before any transaction is processed. Once the user is logged in using the e-mail address, password, and PIN code, they need to once verify their mobile device by clicking on a link sent to their e-mail address. Finally, the user needs to activate the Cloud Mining to start earning free ETN token rewards. Cloud Mining does not require the mobile device to always be connected to the Internet. Once started, Cloud Mining will stay active for a period of 7 days, and the user has to extend it once a week. In the following section, we analyze the security features involved with each of these required steps in detail.

2.2. Security features

This section provides a detailed description of the server-side and client-side security measures that the Electroneum application employs to prevent unauthorized usage of their API and their Cloud Mining process. The relevant features described below were discovered in an iterative fashion while analyzing the application and developing the app-emulating bot.

Selfies: Selfies are an essential requirement of Cloud Mining activation and also comprise a crucial element of our attack. For every account created, it is necessary to submit two selfies with a predefined gesture or drawing (chosen by the server): one when the mining process is started for the first time, and one before the first payout is received. Figures 1 shows three examples of such predefined gestures: in Figure 1a, the account holder is required to submit a selfie while touching his/her forehead, in Figure 1b the user is requested send a selfie with a thumbs up pose, and lastly 1c demonstrates the case where the user is required to take and submit a selfie with a drawing of symbol predefined by the app. Afterward, the submitted selfies go through a (manual) verification process, and in the case that either one or both of the selfies are not accepted, the user is prompted to try again.

App Encryption : The Electroneum application was developed using the Apache Cordova framework and JavaScript. To protect their source code, Electroneum used the "Cordova crypt file plugin" [8], which encrypts files during the build process and decrypts them at run-time. To enable this, a (randomly generated) encryption key is hardcoded into the Android application.

Account Verification: The creation of an Electroneum account is a multi-step process that involves several security features. The Electroneum application rewards the users for each step of the account verification that they complete by free ETNs. First of all, the user needs to provide and conform control over a valid e-mail address. Once this step is complete, the user is prompted to share his/her phone number. However, the server does not send an SMS-verification challenge to verify the possession of the phone number.

Device Authorization : In the process of analyzing the Electroneum API, we observed that devices are authorized based on a randomly generated device hash. New devices need to be approved by the user, by clicking on an activation link sent to their e-mail address.



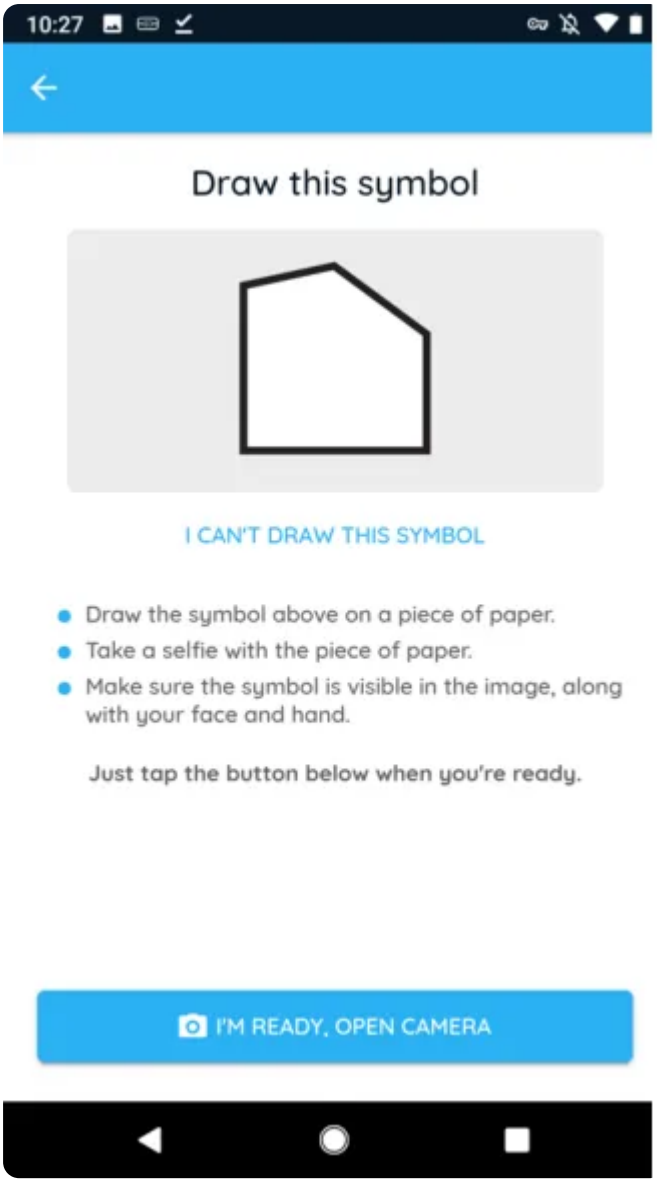
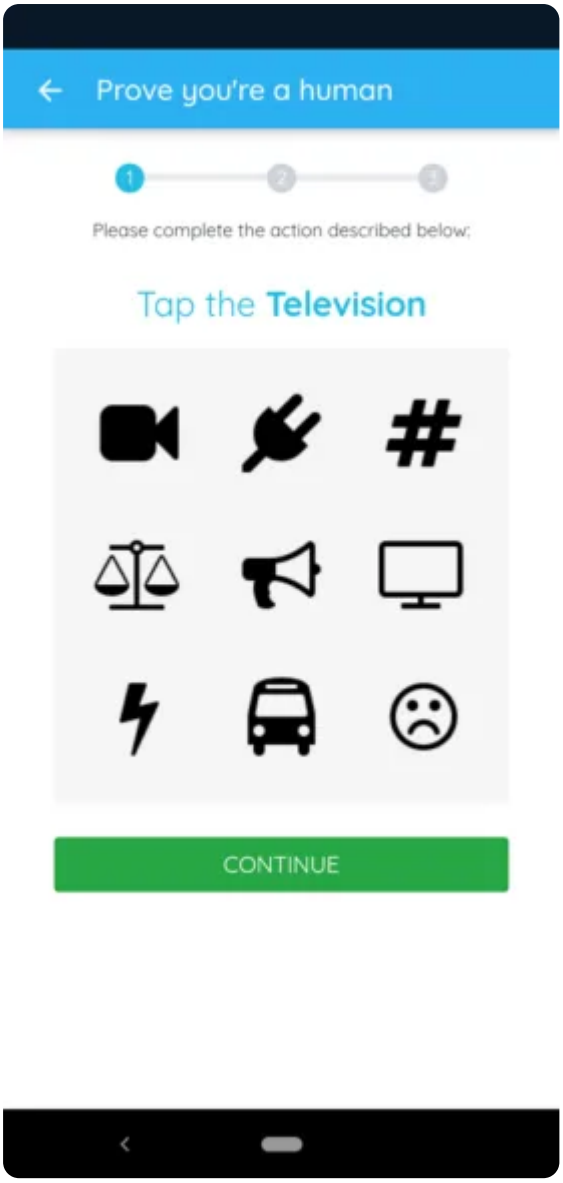

10:27
Draw this symbol
I CAN'T DRAW THIS SYMBOL
Draw the symbol above on a piece of paper.
Take a selfie with the piece of paper.
Make sure the symbol is visible in the image, along with your face and hand.
Just tap the button below when you're ready.
I'M READY, OPEN CAMERA

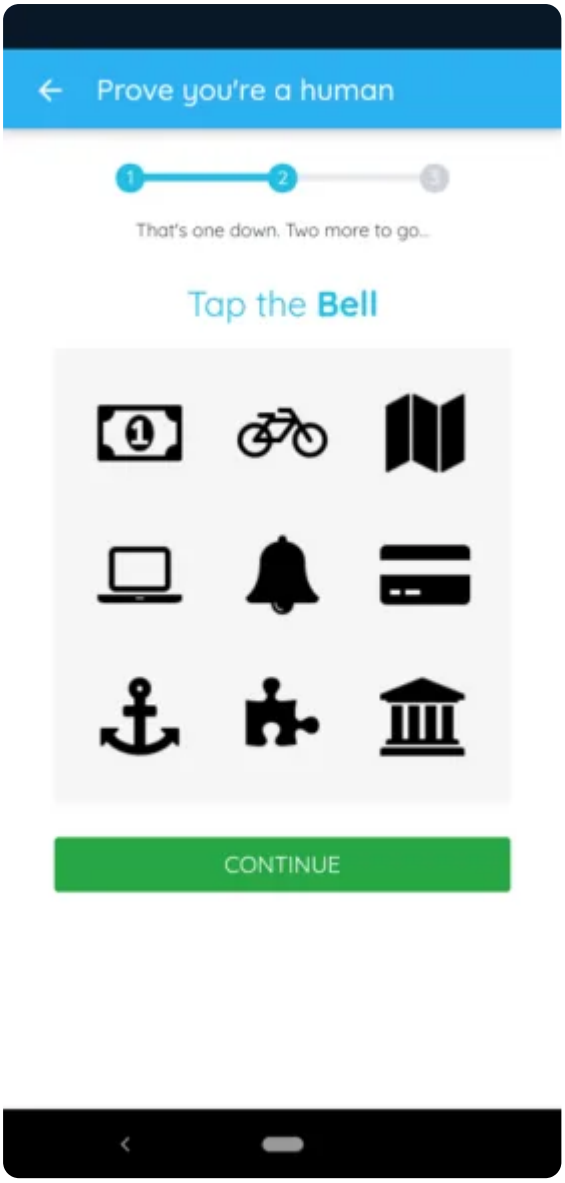


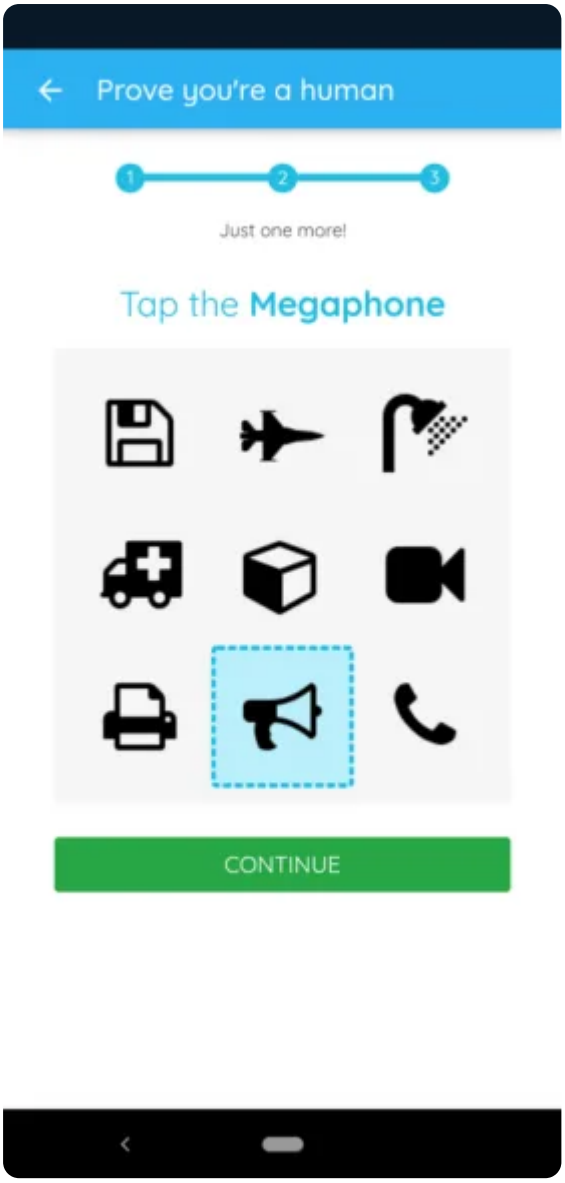
Figure 2: Three example CAPTCHAs to solve in order to activate the wallet.

(a) (b) (c) *Figure 1: Gestures and symbols to replicate and draw during the two selfie creations steps.*

CAPTCHAs : During account verification and activation, Electroneum requires its users to solve three CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). Electroneum employs self-created CAPTCHAs, where the user needs to tap a specific symbol. Figure 2 shows three examples of such CAPTCHAs.


Prove you're a human
Please complete the action described below:
Tap the Television
CONTINUE


Prove you're a human
That's one down. Two more to go...
Tap the Bell
CONTINUE



Maximum Number of Devices: There is no restriction imposed by Electroneum on logging in to the same account with multiple devices. However, the number of devices does not influence the number of tokens received. Also, the performance and other properties of the device do not affect the amount of tokens received.

TOR While the previous version of Mobile Mining of Electroneum used to allow access via Tor IPs, the same is not true for the new Cloud Mining. TOR exit nodes IPs are blocked.

3. Attack

In this section, we describe how the security features presented in Section 2.2 can be circumvented, which would allow a malicious user to earn an increased and unfair share of the reward. We decided to split the attack in a manual initialization phase and an automated phase. The first phase includes all steps that have to be done only once, while the automated phase handles the regular tasks. The next two sections describe these phases in more detail.

3.1. Initial Setup

The initial setup phase includes all steps necessary to create and prepare an account for the use with the developed Android app. These steps are:

1. **Account Creation:** The first step requires only an e-mail address and a user-chosen password. We used Gmail addresses for our experiment.

2. **Wallet Activation:** In order to activate the wallet, the user has to confirm possession of the e-mail address by entering a confirmation code or clicking on a sent link. Furthermore, the user is required to set a PIN for the Electroneum app, enter a name and telephone number, confirm several security hints, and solve three CAPTCHAs. Neither the name nor the telephone number is verified. Electroneum does not use mainstream CAPTCHA solutions like Google reCAPTCHA [9], but instead developed their own approach. Figure 2 shows three sample CAPTCHAs.

3. **Cloud Mining Activation:** The main challenge is the creation of the two selfies. We used StyleGAN [10] and later its improved version, StyleGAN2 [11], to generate photo-realistic photos of non-existing persons. StyleGAN and StyleGAN2 are style-based generative adversarial networks. After generating a face, we took selfies of ourselves in the required gesture, or with the required drawing and swapped the generated face on our selfies. The results are shown in Section 4.

3.2. Automated Attack

After decompiling the Electroneum app, we found the keys necessary to decrypt the JavaScriptbased main part of the app. The relevant code is shown in Listing 1.

Listing 1: Decompiled Java class containing the decryption key and IV. After analyzing the code and capturing the network traffic, we were able to write an Android application, which replicates all relevant requests and thus is able to emulate the original application against the Electroneum server. But the application is not only able to emulate one account but an arbitrary number of accounts. Furthermore, the application does not need any user interaction to

```
public class DecryptResource extends CordovaPlugin {
   private static final String CRYPT_IV = "Qq/y+xTvpdh54r7o";
   private static final String CRYPT_KEY =
         "YgA9pdmOlyZowfy9gOKug9ckNIYrl7Xp";
    private static final String TAG = "DecryptResource";
```

extend the mining process. Instead, on startup, it loads all accounts and checks which account can be extended based on the current IP address and the last time the account has been extended. After extending one account, the application toggles the airplane mode to trigger an IP change [1] and sleeps until the next account can be extended. The next section introduces our evaluation scenario and the results.

4. Evaluation & Results

In order to evaluate our proposed attack, we set up multiple (fake) accounts and used the Android application to extend them automatically. For comparison, we maintained one honest account as a regular user, using the official Electroneum application. Figure 3 shows in the first column three examples of

generated faces. The second and fourth columns show the gesture to replicate or the drawing to hold while taking the selfies. The Third and Fifth columns show the submitted selfies, where we face swapped the generated face from the first column on our selfies.

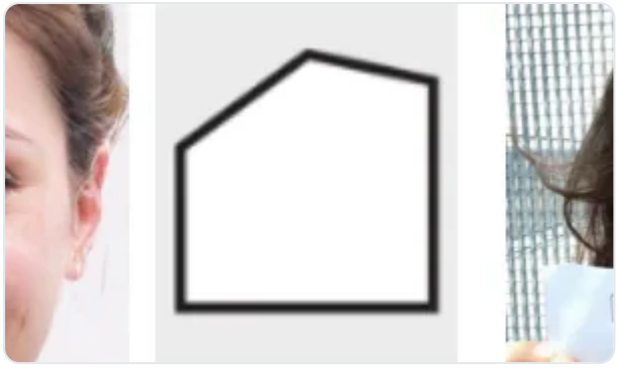
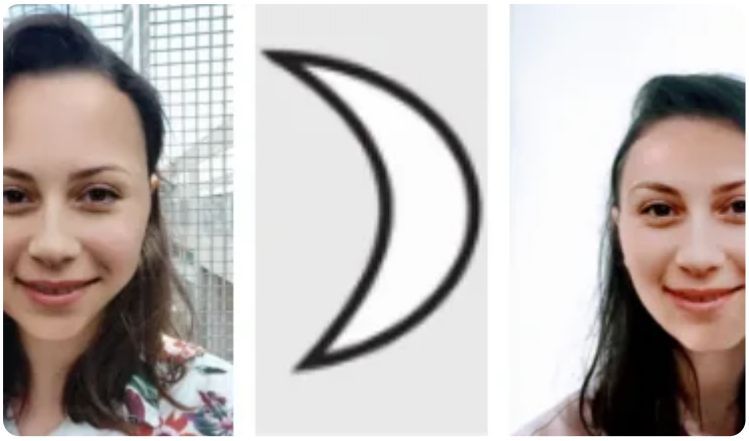


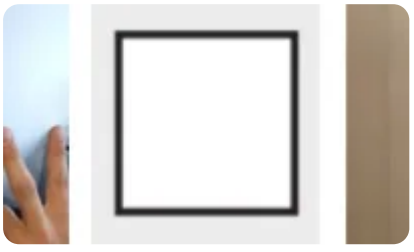
Figure 3: Generated faces, requested gestures and drawings and submitted selfies. 1 Depending on the mobile service operator this may not work. We found it working well with the operator we used in this experiment.



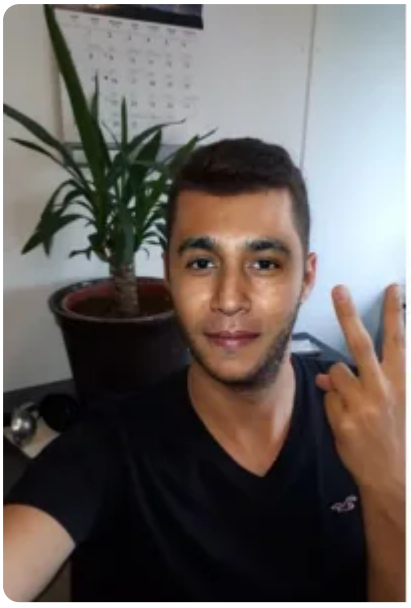




















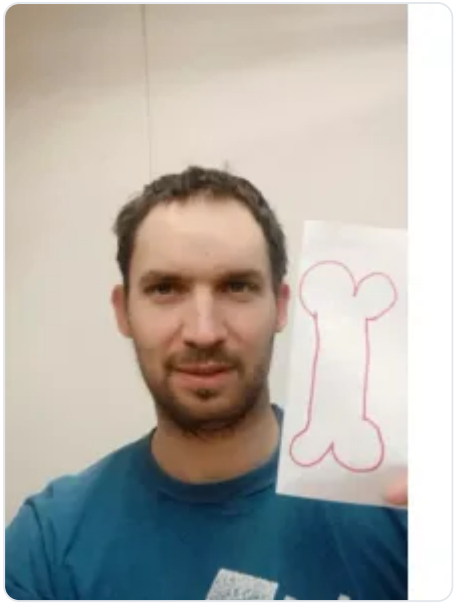
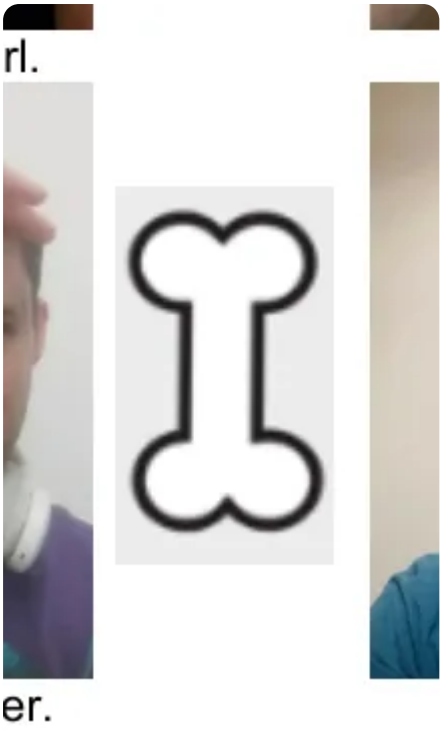


Figure 4: shows the pending balance over time of all our fake accounts and of the reference account maintained using the official app. As shown, all accounts reached the payout multiple times and earned tokens at roughly the same speed.



Figure 4: Pending balance of our accounts over time. Figure 5 shows the mining rate per week of all of our fake accounts. It is noticeable that all accounts earn the tokens at roughly the same speed, but that the mining speed is not constant. It seems that Electroneum adapted it multiple times. This trend can also be observed with our reference account. Figure 6 shows the payout rate of the reference account for a longer time span. It is striking that the timespan between payouts increased over time.



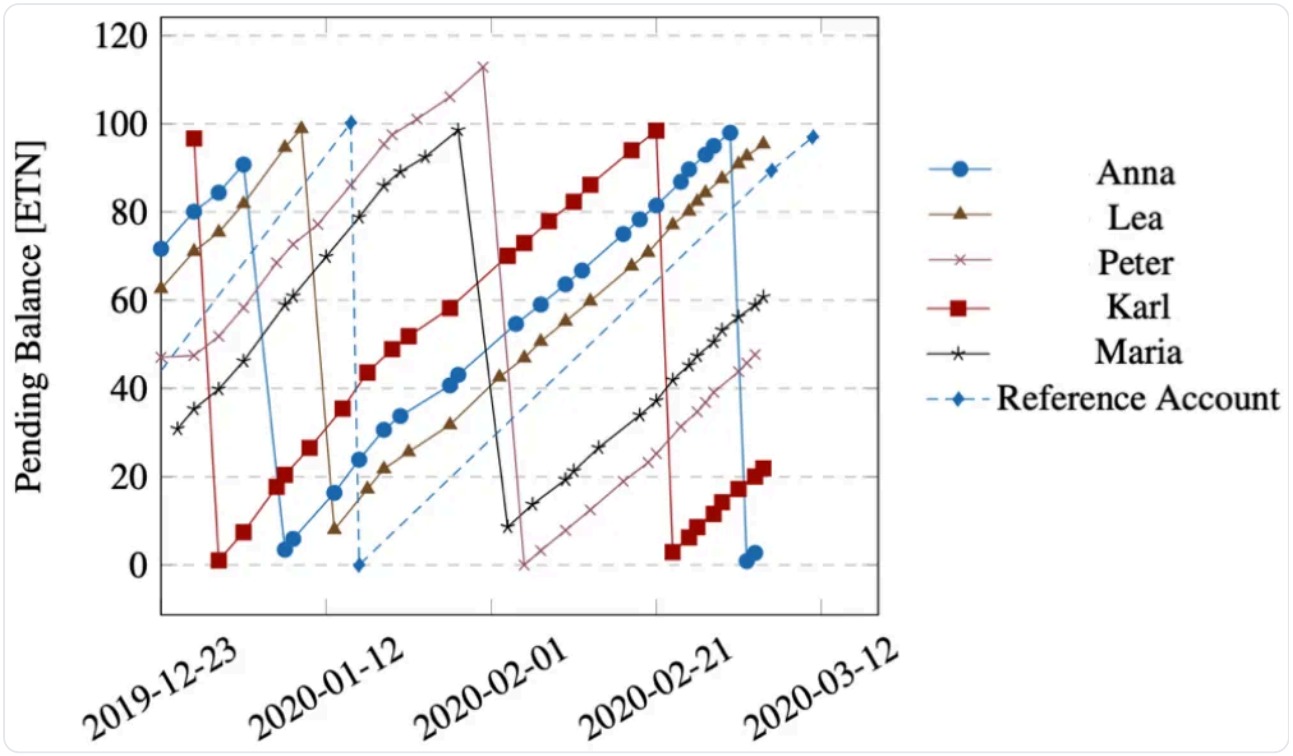
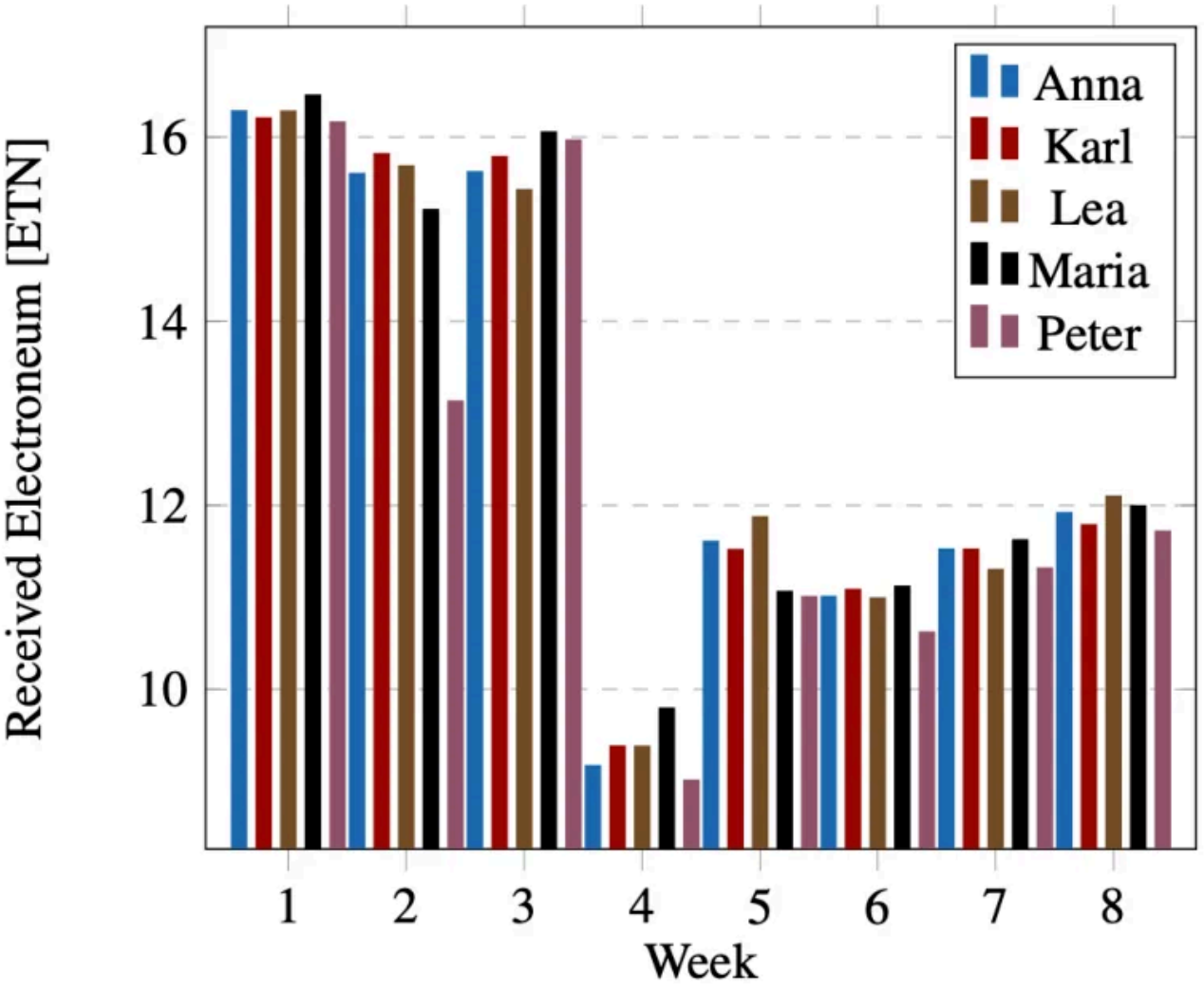
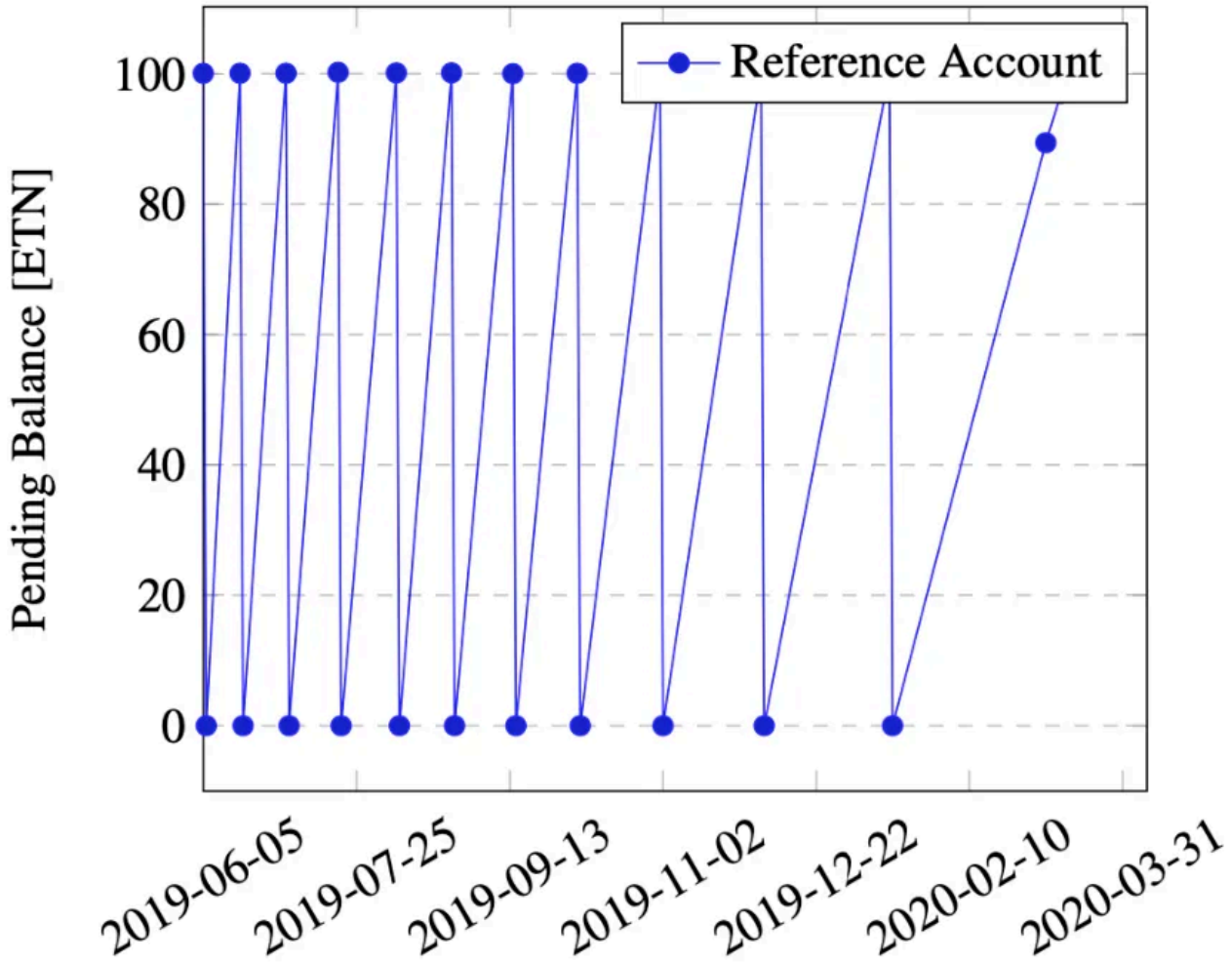


Figure 5: Observed mining rate in Electroneum per Week. Figure 6: Pending balance of the reference account over time.


Received Electroneum [ETN]
16
14
12
10
1
2
3
4
5
6
7
8
Week
Anna
Karl
Lea
Maria
Peter



5. Discussion

Preventing these kinds of attacks is quite hard in an open environment, where open participation is desired. One approach is to ensure unique accounts and devices by binding every account to a limited resource. One example countermeasure would be enforcing stricter identity checks based on government-issued documents, but this would likely prevent some people from participating. On a technical level, mobile client applications can be protected against modifications by remotely verifying their integrity at run time. On Android devices, this functionality is provided via remote attestation. Prünster et al. [12] showed that Android attestation can be used to ensure that an unmodified copy of an application is executed on a secure, not-rooted device. Furthermore, it would be possible to restrict every smartphone to only one copy of the Electroneum application, or at least prevent that both copies earn a reward.

Further mitigation techniques that would increase attackers' efforts include: enforcing phone number verification, applying source code obfuscation, employing certificate pinning to protect their network traffic, and automated checks for modifications on the selfies submitted by the users.

Shortly after the introduction of their proprietary "Moderated Blockchain" model update, the Electroneum team decided to stop "Cloud Mining", and replace it with their marketplace AnyTask [13], a freelancer platform that enables users to sell their digital skills and gain ETN tokens. Therefore, this attack is no longer applicable, as no free ETN tokens are given away. Instead, users have to earn tokens now by offering their skills.

6. Conclusions

In this project we analyzed the security mechanisms employed by the Electroneum Cloud Mining mobile application. We were able to identify multiple vulnerabilities that enabled us to exploit the Electroneum device authorization and account creation protocol to mount an application impersonation and account creation attack. Through this attack, we were able to utilize the Cloud Mining process, and illegitimately earn ETN reward tokens. On a technical level, our attack consisted of an initial account setup phase that required minimal human interaction and a fully automated stage that reconstructs the network protocol to emulate a cloud miner and automatically extends the Cloud Mining process for each of these fake accounts. To evaluate our approach, we created and maintained multiple fake accounts using AI-generated selfies, which enabled us to gain ETN reward tokens at the same rate as our legitimate reference ETN account. Furthermore, our attack can be extended to a fully automated initial account setup stage, (i.e., fully automated combination of generated selfies and the pose/symbol requirement). This would significantly increase the efficiency and the scale of such an attack and enable adversaries to gain considerable amounts of ETN payouts.

References

[1] “Electroneum offers a new way to earn, send and pay.” https://electroneum.com/ (accessed Jul. 20, 2020).

[2] “Phoneum.” https://phoneum.io/ (accessed Jul. 20, 2020).

[3] “Pi Network.” https://minepi.com/ (accessed Jul. 20, 2020).

[4] A. Marsalek, E. Fasllija, and D. Ziegler, “This Selfie Does Not Exist On the Security of Electroneum Cloud Mining,” in *This Selfie Does Not Exist-On the Security of Electroneum Cloud Mining*, SciTePress-Science and Technology Publications, 2020.

[5] “Roadmap - Electroneum - The vision mapped out.” https://electroneum.com/roadmap/ (accessed Jul. 20, 2020).

[6] “Home | Monero - secure, private, untraceable.” https://web.getmonero.org/ (accessed Jul. 21, 2020).

[7] M. Seigen, T. Jameson, N. Nieminen, and A. M. Juarez, “CryptoNight Hash Function,” in *CRYPTONOTE STANDARD 008*, 2013.

[8] “cordova-plugin-crypt-file - npm.” https://www.npmjs.com/package/cordova-plugin-crypt-file (accessed Jul. 21, 2020).

[9] “reCAPTCHA: Easy on Humans, Hard on Bots.” https://www.google.com/recaptcha/intro/v3.html (accessed Jul. 21, 2020).

[10] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks.” 2018.

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN.” 2019.

[12] B. Prünster, G. Palfinger, and C. P. Kollmann, "Fides – Unleashing the Full Potential of Remote Attestation," in *Proceedings of the 16th International Joint Conference on e- Business and Telecommunications*, 2019, vol. 2: SECRYPT, pp. 314–321, doi: 10.5220/0008121003140321.

[13] "AnyTask - Thousands of affordable professional services." https://anytask.com/ (accessed Jul. 27, 2020).