Practical Exploits against the Integrity of Metra Mobile Tickets

Nash Kaminski

nashkaminski@kaminski.io

***Abstract*—In this paper, a series of practical exploits against multiple data integrity vulnerabilities present in the Ventra mobile application are discussed. All exploits effect the current production version of the application and do not require any modifications to the application or the system software of the device that it is running on.**

I. Introduction

As a daily Metra commuter and computer engineering student at the Illinois Institute of Technology, I was significantly interested in the Ventra application pretty much from the day it was released, owing to the major convenience that is provides. However with a strong background in low level programming as well as computer networking, I quickly became curious as to the inner workings of a few features of the app, primarily surrounding Metra mobile tickets. Even though the app was claimed to be secure, I did not see a logical means given how the app operates for such functionality to actually be secure. After a small amount of investigation into such, I have discovered a series of serious vulnerabilities with trivial to moderate difficulty of exploitation within the application that allow for the violation of the integrity of Metra mobile tickets. None of the vulnerabilities presented require system level modifications to the client device or any modifications to the Ventra application.

II. Ventra Application Mobile Ticket Integrity

A. Impact

Allows for limited-use tickets such as the one-way and ten-ride tickets to be used an infinite number of times.

B. Description

The Ventra mobile application renders mobile tickets stored locally on the device prior to being able to successfully update the remaining number of uses for a given ticket on the central server. Therefore, the following procedure can be used to use a limited use ticket an infinite number of times:

1) Install(if necessary) and run Ventra application.
2) If ticket is owned under a Ventra account, sign in to the account.
3) Ensure ticket is displayed under ”My Metra Tickets”. If not, sync manually using the button at the top right.
4) Activate airplane mode on the device.
5) Use the mobile ticket(s).
6) If the device is an Android device, go Settings $\rightarrow$ Apps $\rightarrow$ Ventra and choose ”Clear Data”.
7) If the device is an iOS device, uninstall the Ventra application.
8) Disable airplane mode.
9) Repeat procedure as many times as needed.

If this procedure is followed properly, the device will render the ticket that it has stored locally, but will be unable to notify the central server of the ticket's use. While the app does appear to maintain a queue of requests that are to be made to the central server, this queue can be cleared by clearing the app's database or reinstalling it. This prevents the central server from ever being notified of the ticket's use and therefore the original number of remaining uses is restored upon resyncing with the Ventra account after the app has been reinstalled or the data cleared. This allows for a theoretically infinite number of uses of a limited-use ticket.

III. Ventra Application API Information Disclosure

A. Impact

Allows any unauthenticated user to access 14 days of mobile ticket security codes.

B. Description

The Internet based API accessed by the Ventra mobile application located at https://ventra.transitsherpa.com/v2/rider/sync does not verify that the device ID presented in the request headers actually owns any tickets prior to providing daily security codes. Additionally, the request signature provided in the *x-gs-signature* header of the request is not validated by the remote server. Any request sent with a recent timestamp results in the remote server providing the user with 14 days of valid ticket security codes. A proof of concept is implemented as the main function of v_client.py and can be run by running:

```
python3 v_client.py
```

IV. Ventra API Spoofing

A. Impact

Allows for the rendering of a valid Metra mobile ticket with all parameters arbitrarily defined.
Allows for limited-use tickets such as the one-way and ten-ride ticket to be used an infinite number of times.

B. Description

This vulnerability is by far the most technical of the vulnerabilities and is partially dependent on the previous vulnerability for a source of valid mobile ticket security codes. However, it allows for a user to cause the Ventra mobile application to render *any mobile ticket, regardless of what tickets, if any, the user actually owns* via the impersonation of the Internet based API accessed by the Ventra mobile application. While SSL is used to secure the communication between the Ventra application and its central server, all trusted certificates on the device, including user installed certificates are trusted. Therefore, if the user installs his/her own SSL CA certificate on the device certificates issued by the user defined CA will be trusted by the Ventra application. In order to gain control over the IP address that the Ventra application accesses, the VPN functionality built into Android and iOS can be used as designed to gain control of DNS resolution. This allows the user to control the IP address that ventra.transitsherpa.com resolves to on the device. Combined with a user installed CA certificate, this then allows a user to intercept and/or impersonate the central server accessed by the app at https://ventra.transitsherpa.com/v2/rider. No additional verification of the data is performed by the Ventra application. Therefore, a user controlled server application, such as proof of concept located in

```
app_server.py
```

can be used to send responses to the app that include fully user defined Metra mobile tickets. Valid security codes can also be sourced using the previous vulnerability, such that the spoofed ticked appear completely genuine.

V. Conclusion and Recommendations

All of these vulnerabilities enable the theft of mobile ticket fares and therefore should be mitigated or patched as quickly as possible. With regards to fixes, I would advise properly implementing request signature verification in your web API, as well as only providing security codes to devices which actually own valid mobile tickets. Serial numbers should be made unique to each ticket and the QR code containing such actually validated by conductors. Tickets should only be rendered after the device has successfully communicated the ticket's use to the central server. If rendering tickets without a data connection is absolutely necessary, such tickets should have their QR codes scanned to ensure that their use is recorded. The implementation of HTTP public key pinning should be considered such that the Ventra application only communicates with servers presenting HTTPS certificates on a whitelist distributed with the application. Together, the implementation of either these measures or equivalent measures will make manipulation of the Ventra application and potential potential theft of Metra fares far more difficult for those who attempt to use these exploits for nefarious purposes.

NOTE

This paper was written using the IEEE LaTeX style found here: IEEE - Manuscript Templates for Conference Proceedings