Practical Exploits against the Integrity of Metra Mobile Tickets

Nash Kaminski

nashkaminski@kaminski.io

Revision 1.1

***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 unless noted otherwise and do not require any modifications to the application or the system software of the target device.**

I. Introduction

As a daily Metra commuter and former computer engineering student at the Illinois Institute of Technology, I was significantly interested in the internal workings of the Ventra application pretty much from the day it was released. Specifically, I was curious as to the inner workings of a few features of the app in particular, such as those involving 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 be truly secure. After some investigation into such, I have discovered a series of serious vulnerabilities with varying difficulty of exploitation within such application that allow for the violation of the integrity of Metra mobile tickets and the compromise of user-entered data. None of the vulnerabilities presented require system level modifications, jailbreak, or root access to the target device or any modifications to the Ventra application itself and may be successfully executed as long as one controls the network traffic flows to and from the target device.

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. *Mitigated 6/24/16 in release 1.2.1.39*

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 remote server. Therefore, the following procedure can be followed to activate a limited use ticket a theoretically 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) Observe that when the application is reopened (or in the case of iOS, reinstalled) that the use counter has *not* decremented.

If this procedure is followed properly, the device will render the ticket that it has stored locally, but will be unable to notify the remote server of the ticket's use. While the app does appear to maintain a queue of requests that are to be made to the remote server, this queue can be cleared by clearing the app's database (Android) or reinstalling it (iOS). This prevents the application from communicating the ticket usage event to the remote endpoint. Therefore, the original number of ticket activations is restored upon resyncing with the Ventra account after the app has been reinstalled or its database cleared allowing for a theoretically infinite number of uses of a limited-use ticket. Later versions of the Ventra application appear to maintain a copy of this queue and the ticket state within an encrypted file on the device filesystem. However the key to this encrypted store appears to be present as a constant within the application code and therefore this measure is likely able to be circumvented without great difficulty.

III. Ventra Application API Information Disclosure

A. Impact

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

B. Description

The remote 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 and claiming to be sent by an application running on a supported Android/iOS version 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 is by far the most challenging to exploit of the vulnerabilities presented and is partially dependent on the previous vulnerability for a source of valid mobile ticket security codes. However, such 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 remote API endpoint accessed by the Ventra mobile application as well as compromise the integrity of some or all user entered data. While SSL is used to secure the communication between the Ventra application and its remote endpoint, no certificate validation is performed. As a result, the confidentiality and integrity of all data transacted to and from the remote API endpoint is compromised. One may in turn leverage either control over the WiFi network that the target is connected or Android and iOS's built in VPN functionality in order to gain control over the traffic flow between the application and backend server via either DNS modification and/or IP routing changes. Given control over such network flow, this allows the user or a malicious actor to intercept and/or impersonate the API endpoint accessed by the application at https://ventra.transitsherpa.com/v2/rider. No additional client-side verification of the data is performed by the Ventra application. A proof of concept server implementing a subset of the backend endpoints is located in

```
app_server.py
```

and may be used to send responses to the app that define arbitrary Metra mobile tickets. Valid security codes can also be sourced using the previous vulnerability, such that the generated tickets appear completely genuine.

V. Conclusion and Recommendations

All of these vulnerabilities can enable the counterfeiting of mobile tickets as well as compromise of data transacted via the application if leveraged by a malicious actor. Therefore, these flaws should be mitigated or patched as quickly as possible. With regards to addressing these flaws, I would advise properly implementing request signature verification in the 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 should be periodically audited. Tickets should only be rendered after the device has successfully communicated the ticket's use to the backend server. If rendering tickets without a data connection is absolutely necessary, such tickets should display a prominent notice and have their QR codes audited to ensure that their use is recorded. HTTPS certificates should always be validated and the implementation of certificate pinning should be considered, such that the user may not install his/her own certificate onto the device and regain the ability of spoofing the backend API server. Together, the implementation of either these specific measures or functionally similar measures will make manipulation of the Ventra application as well as the potential compromise of ticket authenticity or user data far more difficult.

NOTE

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