KERBY
KERBEROS CREDENTIALS MANAGER SYSTEM FOR ANDROID

Deborah Chen, Catherine Zuo, Shiyang Liu, Isra Shabir

I. Overview

Kerberos is an authentication system which requires a user to only input their password once in order to gain access to several individual services via tickets. This access lasts for as long as the tickets are valid.

In the mobile world, a user will typically access each of these services through a separate app. To make use of Kerberos, the user provides a password, which the app then uses to fetch tickets for the service. This is a security concern, because the user has to trust each individual app with their password. Additionally, the developer would have to implement Kerberos for each service app.

We identified the need for a centralized Kerberos manager, and chose to implement Kerby, a system that extends this ticket granting functionality to service applications built in Android. A service app broadcasts Android Intents in order to securely communicate with a Ticket Granting Application. In return, the service app retrieves a service ticket that it can use, for instance, to authenticate itself on a third party server/application. This report is divided into three sections; Section 2 describes the system components and implementation, and Section 3 sheds light on security.

II. Description of System

Kerby employs four main components: the IS&T Ticket Granting App, an example XVM Service App, a Proxy Server, and the XVM service run by SIPB. The “XVM Service App” communicates with IS&T’s Ticket Granting application to retrieve a ticket. It then uses this ticket to authenticate itself on a third party server. In our case, we created a proxy server that would accept these tickets to use in communication with SIPB’s XVM server, but in general, the developer could collapse these two functionalities into one server.

A. System Components

IS&T Ticket Granting Application: This app provides to authentication to a Kerberos user and grants a TGT upon login. In our model, this app also needs to securely communicate with a service application. Hence, we customized the application in order to receive and send intents besides retrieving and granting tickets. In our system, this app acts like a black box which retrieves the appropriate service tickets for sending to the service apps.

XVM Service App: Technically, this is the initiating point for a user. This app sends a broadcast intent to the Ticket Granting Application in order to ask for a ticket service. In order to receive a ticket, this app also listens for broadcasting intents from the Ticket App. So when the ticket is sent back, our service app receives an intent with the ticket as its extra information.

Our service app has two main features: it allows users to view a list of their virtual machines, as well as reboot a specified VM in XVM, a virtualization service run by SIPB. We chose XVM because it supports Kerberos tickets as a form of authentication, and supports a command line interface through the remctl protocol.

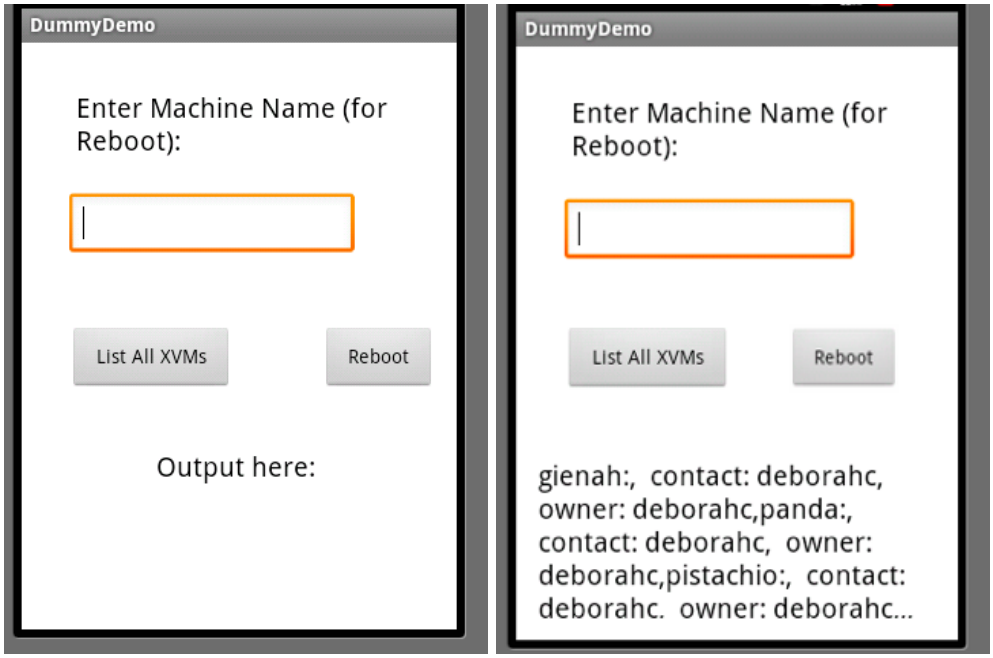

Figure 1. The XVM service app allows a user to list all their VMs and reboot a specified machine. The output of List all XVMs is shown above.

Proxy Server: After the service ticket for XVM is received, the app sends an HTTP POST request to an XVM proxy server (hosted on XVM itself! panda.xvm.mit.edu). The request contains the ticket and the corresponding parameters with the desired command (list or reboot) and the name of the VM if necessary.

The proxy server saves the ticket to a temporary file and calls a python script that runs the XVM **remctl** command, setting the ccache location to the new temporary file. The output of the python remctl script command is then sent back to the XVM service app in the HTTP response, and is displayed to the user.

The server is running on Ubuntu 10.04.4, and we chose to use Node.js and Express for ease of development. remctl and python-remctl was installed as well.

XVM Server: XVM provides virtualization service to the MIT community. The XVM server is run and maintained by the SIPB and hosts VMs for free. Our Proxy Server server issues the XVM remctl command, the output of which is then sent back to the XVM service app in the HTTP response, and is displayed to the user.

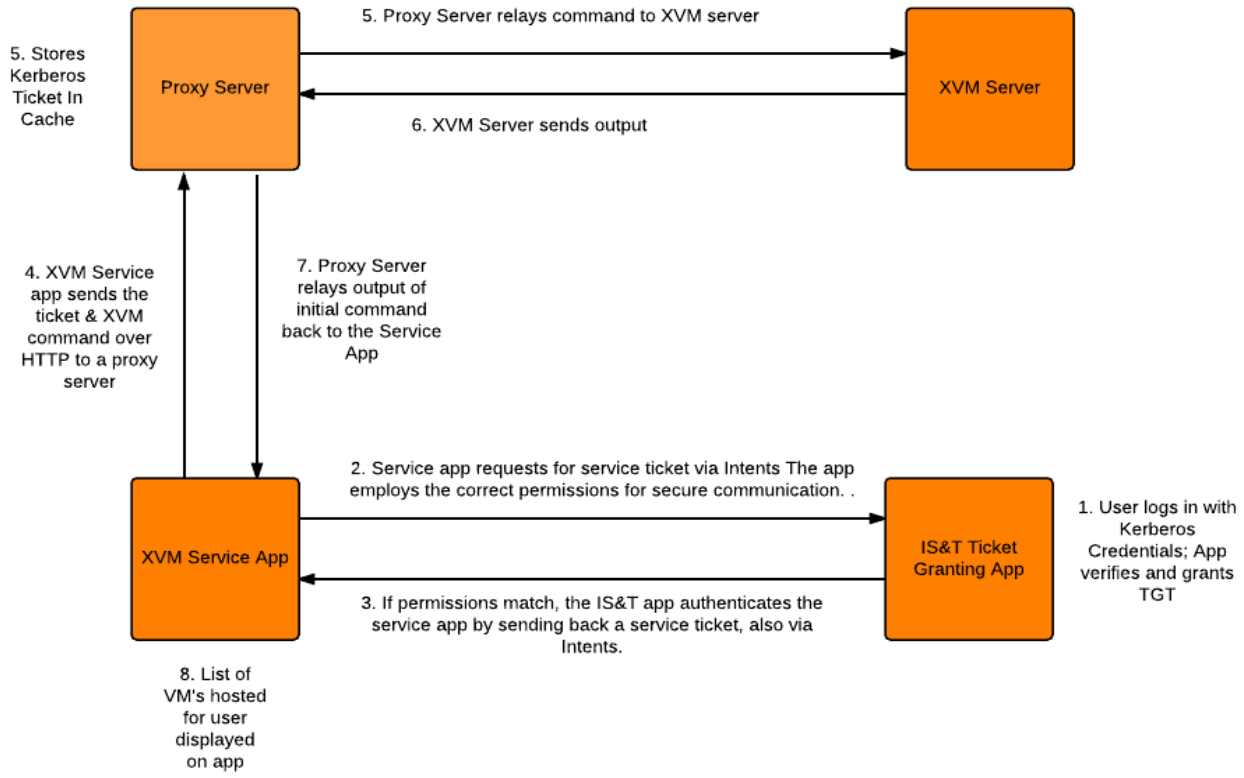

Figure 2. System diagram of Kerby.

III. Security

A. Threat Model

The adversary is a malicious third party Android app that tries to obtain the user's Kerberos tickets without the user's knowledge or permission. The adversary may try to send intents to trick the Kerberos app into replying with tickets.

B. Intent Permissions

The main goal of our Android-side security is for the Kerberos app to only exchange Intents with user-trusted apps. In the Kerberos app we create three permissions, all of which are marked dangerous and must be user-approved. The *SEND_REQUEST_PERM* grants apps the ability to send Intents to the Kerberos app, while the *GET_REPLY_PERM* grants apps the ability to receive replies from the Kerberos app. By using these two permissions, we only let the Kerberos app receive Intents from apps with *SEND_REQUEST_PERM*, and send replies to apps which have been granted *GET_REPLY_PERM*. The *KERB_LISTENER_PERM* is held by the Kerberos app and should not be granted to service apps.

To give developers the option of not sending the user's service ticket to all apps with *GET_REPLY_PERM,* the Kerberos app looks for a receiver package name in the request Intent; if it is there, it will send the reply only to that package. If the service app wants to make its outgoing Intents private from non-user-trusted apps and/or securely receive Intents from the Kerberos app, we facilitate this by having a unique permissions label for the Kerberos app (*KERB_LISTENER_PERM*). This way, service apps may specifically send Intents to the Kerberos app, and listen to its replies, securely.

C. Kerberos Tickets

Originally IS&T's app appended the service tickets to the same file for consecutive service ticket requests. This is problematic, because this could cause the Kerberos app to also grant service tickets of services previously requested by other apps, which the currently requesting app did not specifically ask for. We modified the IS&T app so that for each service ticket request, it saves the service ticket in a separate file, which is deleted after the ticket gets sent to the service app.

D. Proxy XVM server security

The Proxy XVM server stores all tickets in a ccache in the location /tmp/krb5cc_{RANDOM_HASH}. Upon receipt of a POST request containing a ticket, the server generates a random hash, 'randomBytes(32).toString('hex')', and stores the ticket in the corresponding location. After the ticket is used to communicate with the SIPB XVM server, it is deleted immediately. In this way, if the proxy server is compromised, there will not be a large trove of tickets available (though any tickets acquired would expire eventually)

The proxy server only supports two commands, list and reboot, parameters that are predefined within the XVM service app. All other commands will be rejected. In the case of the VM machine name the user inputs in the service app, the proxy server relies on the service app to validate that the name is alphanumeric and optionally has dashes (in accordance to SIPB's guidelines). This prevents users from crafting malicious inputs, though the scope of an attack in this realm is limited in that all commands are eventually formatted to a remctl defined interface.

IV. Future Work

We wrote a simple demo app; future work may considerably expand functionality and usability - for example, accessing different services. The IS&T Kerberos app could also be improved to allow multiple users.

We did not incorporate HTTPS into our server, but using HTTPS would greatly reduce the risk of a man in the middle attack.

V. Acknowledgements

Steven Valdez, Professor Zeldovich, General 6.858 TA Staff, Benjamin Kaduk (IS&T), David Benjamin (MIT CSAIL, SIPB), SIPB Community

VI. Appendix

Our code is online at the following URLs:

Demo Service App - https://github.com/cthrnez/DummyDemo/tree/lsyang
Proxy Server - https://github.com/deborahc/kerb-server
Modified IS&T Kerberos Ticket-Granting App - https://github.com/deborahc/kerberos-android-ndk/tree/lsyang