How to secure Stata in a secure environment?

Bjarte Aagnes, Cancer Registry of Norway

The 2022 Northern European Stata Conference
background/motivation

New data extraction and delivery project:

- legislation/regulations/corporate guidelines
- ultimatum: secure Stata use (or ...)

cancer patient data

• data integrity
• privacy of sensitive data (information on health issues)
• monitoring/logging: who did what when with data
CHALLENGE: Stata is basically open:

- read/write data
- optional logging
- Stata jdbc add (clear text password)
CONTEXT: data flow
data flow

A  DWH  JDBC  B  Stata  JDBC  C  resDB
data flow
Stata
securing Stata

- define trusted/private network zone
- Java plugin for DB access and audit logging
- logging of Stata session
define trusted/private network zone

- firewall restrict data communication by rules
- private file server/file system (Stata network installation)
- personal clients (persistent VDI VMware Horizon Clients for Windows)
Java plugin for DB access via JDBC

- provides connection string
- hides JDBC connection password
- provides audit log of data load (who did what when)
Java plugin

- Reads `c(username)`. The user is *authenticated* when logging into the VDI
- Finds AD-group for user
- Maps AD-group to a *database service account*
- Decrypt and provide password for *database service account*

The java plugin provides a connection string including a secret password stored (encrypted) in java resource file
- The user does not know the password
- The *DB service account* provide *authorization* to relevant data in DB
- Next version will use Vault (HashiCorp) to provide short lived password (password rotation)
import com.stata.sfi.Macro;
import com.stata.sfi.SFIToolkit;

//Service stata commands to disable debugging and tracing
SFIToolkit.executeCommand("set trace off", false);
SFIToolkit.executeCommand("set debug off", false);

//Template connection string
String run_jdbc_add_dsn =
    "capture noi jdbc add " + jdbcConnectionValues.get("DSN") + ", " +
    "jar(" + jdbcConnectionValues.get("JAR") + ") " +
    "driverclass(" + jdbcConnectionValues.get("CLASS") + ") " +
    "url(" + jdbcConnectionValues.get("URL") + ") " +
    "user(" + jdbcConnectionValues.get("USERNAME") + ") " +
    "password(" + jdbcConnectionValues.get("PASSWORD") + ")";

//Execute the stata command
int jdbc_add_dsn_result = SFIToolkit.executeCommand(run_jdbc_add_dsn, false);
What do the user do to fetch data from DB?

* typically define SQL select statment in filename.sql

KRG_KNUT_DWH_load, sqlfile("filename.sql")

```plaintext
prog define KRG_KNUT_DWH_load, ///
   nclass ///
   properties(KRG_KNUT_DWH)

findfile java_auditlog_plugin.jar // dependencies (plugin)

version 17

syntax, sqlfile(string) [DISPLAY]

confirm file "\sqlfile";
scalar sql = fileread("\sqlfile")
local sql = ///
   char(34) ///
   + trim(itrim(ustrregexra(sql,"\s|\x09|\x0a|\x0d"," "))) ///
   + char(34)

javadoc krg.JavaAuditlogPlugin executeSql, jars(java_auditlog_plugin.jar)

if ("\display" != ") {
   di _n _n "\sqlfile"
   di scalar(sql)
}
```

end
java code (elements) using Stata-Java API (https://www.stata.com/java/api17/)

```java
//Get the sql from Stata
String sql = Macro.getLocal("sql");
String run_jdbc_load_sql =
    "capture noi jdbc load, exec(" + sql + ")" + " " + "clear";

//run Stata jdbc load
SFIToolkit.executeCommand(run_jdbc_load_sql, false);

//run standard Stata commands
SFIToolkit.executeCommand("noi compress", false);
SFIToolkit.executeCommand("noi describe", false);
```
KRG_KNUT_DWH_load

KRG_KNUT_DWH_load, sqlfile("filename.sql")

- Call the `executeSql` method, the only method exposed to user:
  - provide connection (jdbc add)
  - fetch data (jdbc load)
  - close connection (trick: jdbc add non-existing URL)

`javacall krg.JavaAuditlogPlugin executeSql, jars(java_auditlog_plugin.jar)`
Stata session log

- logging Stata session (started by sysprofile.do)
- monitoring Stata session log for illegal commands every x seconds (GoAnyWhere)
- moving Stata session log files for archiving (GoAnyWhere) for x months
What Stata properties made this possible?

- Stata jdbc command
- Stata-Python API (https://www.stata.com/python/api17/Data.html) for fast prototyping
- Stata-Java API (https://www.stata.com/java/api17/)
What would we add to Stata?

- JDBC close connection command
- run default commands at end of session (like sysprofile.do)
What did StataCorp fix?

- FIX update 04oct2022: jdbc load, when loading CLOB data, produced error when a NULL value was encountered.
What has been solved?

- restricting data reading/writing (plugin)
- logging of data reading/writing (plugin)
- avoiding clear text password (plugin)
- forced logging of Stata session including monitoring of content
Not covered

- formal guidelines for use
- training
- git/gitlab
Thanks