Description of the interaction between the GAT.EXE program and the GAT.MDB database.

April 10, 2001

Introduction

The purpose of the combined GAT.EXE program and the matching GAT.MDB Microsoft Access database is to create an environment which allows field agents to quickly and reliably verify that the authentication information received from a "game under test" in the field matches the benchmark authentication data previously received from a known good game in the laboratory.

This document describes how the GAT.EXE program, in conjunction with the GAT.MDB database, work together to perform those functions.

Description of the GAT.MDB database

The GAT.MDB database is a simple heirarchical database. It is supplied in a minimal form. Each owner of the database can add tables and fields to it, if they wish. But it is important that the tables and fields as supplied not be deleted or renamed. Any deletions or renaming will render the GAT.EXE/GAT.MDB interaction inoperative.

The GAT.MDB database contains the following tables:

Version

This table should only have one record in it. The "Version" entry of the table is pure text, and should be updated manually from time to time so that field agents and headquarters can be sure that the data exported from the GAT.MDB database and copied to the field agents' laptop computers are synchronized. The Version code is reported by the GAT program.

The MostRecentUpdate column is a Date/Time entry that will be automatically updated by the program whenever the "Export GatData" function is executed.

Manufacturers

This table contains information about each manufacturer of GAT-capable games. Each manufacturer can make many games.

Games

This table contains information about all of the games in the database. Each game is linked back to its manufacturer through the ManufacturerID value. Each game can have many hash values.

Hashes

This table contains “Seed Text” combined with the resulting hash value. Each seed/hash pair is linked back to a game through the GameID value.

Note: At this writing, no game has the ability to combine a seed string with the rest of the game data to produce a “forensic” response. The capability to track seed/hash pairs has nonetheless been built in to the GAT program, however, for purposes of testing and evaluation, and to allow for future development.

Principles of Operation

The functionality of the system is based on the SHA-1 “Secure Hash Algorithm”. This algorithm, developed by the National Security Agency, is recommended by the National Institute of Science and Technology for use whenever a secure hash is needed.

The SHA-1 algorithm produces a 160-bit “digest” of a stream of bits. The digest is known as a “hash.” The nature of the SHA-1 is such that any single bit change in the input stream causes each bit of the hash to change with a probability very close to 50/50. The probability that two different messages will accidently produce the same hash is effectively zero. (The odds against that happening are about 1,500,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 to one.) The NSA has also determined that it is computationally infeasible for somebody to intentionally create a second bit stream that generates the same hash as another bit stream. (It is this second feature which qualifies it as a “secure” algorithm.)

The GAT program operates by issuing an authentification request to a game machine. It performs an SHA-1 hash on the entire response (which itself may include secure signatures generated by the game machine.)

The GAT program then looks for that 160-bit hash in a table of previously-generated hash values. If it finds a match, the program displays the manufacturer and name of the game in the table. If there is no match, it says so.

The Field Version of GAT

For a field agent, that Go/Nogo response is all that GAT needs to provide. Thus, there is no need for a field agent’s computer to contain the entire database. Instead, it simply has a GATDATA.TXT file in the same folder as the GAT.EXE executable. This GATDATA.TXT file is generated in the laboratory, and contains a list of all the manufacturers, games, and the seed/hash pairs.

From time to time, updated versions of the GATDATA.TXT will, of course, be created. The system provides field agents with information about the version of the DATDATA.TXT file and when it was created.

The Laboratory Version of GAT

The GAT.EXE file that is used in the laboratory is actually identical to the one used in the field. The difference is the presence, on the lab computer, of the GAT.MDB database.

That database has to be installed as an ODBC Data Source with the name “gatdata”; instructions for doing that are found at the end of this document.

The only real difference in functionality between the laboratory and field version is that the GAT program, upon calculating an SHA-1 hash that it can’t find in the table, will notice that the gatdata ODBC is available and will offer the operator the option of entering the new hash into the database.

If that option is chosen, the hash data is entered in the table, a new GATDATA.TXT file is generated and distributed to the field agents, and that machine’s identity can subsequently be checked in the field.

Data Entry Considerations

In an effort to avoid data entry problems, the GAT program can only add seed/hash pairs entries to games and manufacturers that are already in the database. This results in the following process flow.

Process Flow Summary

1. A new company called “Paradise Gaming” wheels a new game called, “Tropical Sunset” into the lab.
2. The operator opens the GAT.MDB database and opens the “Manufacturers” table. The company, “Paradise Gaming” is added to the end of the list. The table is closed.
3. The operator opens the “Games” table. The game, “Tropical Sunset” is added to the end of the list. The table, and the database, are both closed.
4. The game is attached to the computer and the GAT program is started.
5. The “Authenticate” button is pressed.
6. GAT announces that this hash is new, and asks if it should be entered into the database.
7. The operator clicks “Yes”, and enters the new hash value into the database.
8. The operator then opens the database, opens the “Export GatData” form, and presses the “Create GatData.Txt File” button. (The new file will appear in the same folder that contains the GAT.MDB database file.)
9. The new GATDATA.TXT file is distributed to all field agents. It goes into the same folder on their laptops as the GAT.EXE executable.
10. Subsequent authentication requests of that game will result in GAT finding the entry in the GATDATA.TXT file.

Simulation, Evaluation, Test and Training

Rather than require that an actual game be available for test purposes, the GAT program has the

ability to simulate the existence of a game. Start GAT and click the "Setup" button. Along with the option of the four COM ports, there is a "SIM" option. Using the SIM option will cause GAT to ask you for the name of a simulated game for testing purposes.

It is recommended that the interested reader install the GAT.MDB database as described below. Use the Setup button to set the COM port to "SIM". Then go through the procedure detailed above to create the "Tropical Sunset" game from "Paradise Gaming."

When you do so, remember two things:

First, when asked for the name of the simulated game, be very precise in your typing. The name (and any seed string you might have specified) become part of the simulated response from the game – changing a letter from uppercase to lowercase, or adding an extra space between words, will result in a completely different hash.

Second, don't forget to generate the updated GATDATA.TXT file after entering a new hash entry into the database. If you don't generate the updated file, and put it into the same folder as the GAT.EXE program, then GAT won't be able to find the hash.

Note: This behavior is intentional! It is designed to make sure that the laboratory version of GAT is using the same GATDATA.TXT file that will be distributed to the field agents

If you forget to generate the new GATDATA.TXT file on the laboratory computer, you will get the "Should we add the new hash to the database" message again. If you proceed with the addition, you will get an error message about a duplicate entry, because the database is designed not to allow duplicate hash values.

Instructions for installing the GAT.MDB Access Database on a computer.

1. Create a folder called C:\GatDatabase
2. Copy GAT.MDB into C:\GatDatabase
3. Click on Start → Settings → Control Panel → ODBC Data Sources
4. Select the System DSN tab
5. Click on "Add"
6. Highlight "Microsoft Access Driver"
7. Click "Finish"
8. In the Data Source Name field, enter "gatdata"
9. Click "Select"
10. Browse to C:\GatDatabase\gat.mdb and click "OK"
11. You are back in the "ODBC Microsoft Access Setup" dialog box,so click "OK"
12. You are back in the ODBC Data Source Administrator dialog box, so click "OK"
13. That's it. The GAT program will now be able to access that database using ODBC calls.

Note that the database can actually be anywhere on the computer, it can even have a different name. It is necessary, however, that the Data Source Name be "gatdata"; it is through that name that the GAT program finds the database.