

## Sequoia Voting System database laid bare but no secrets found

by David M Williams  
Thursday, 22 October 2009

Liberal US shock-blog Daily Kos gained legal access to the database used by electronic voting machines produced by Sequoia Voting Systems. The Daily Kos sensationally claimed the database violated Federal voting law. A closer examination gives a different story.

Today the site posted "breaking news" by a contributor going under the moniker Mokurai. The headline in question stated "Sequoia Voting Systems hacks self in foot."

In essence, an organisation known as the Election Defense Alliance (EDS) was granted access to the database used for elections held in Riverside County on Sequoia's equipment.

EDS paid \$USD 105 for the rights to the database, and was not constrained by a non-disclosure agreement (NDA). EDS made good use of this omission by sharing the database to members of the Open Voting Consortium (OVC) mailing list, of which Mokurai is a founding member.

Before continuing, let me go on record as having pointed out failures in closed-source voting systems in the past.

In December I spoke about how a Python programmer discovered optical Premier Election Systems, formerly known as Diebold, had a software bug which actually miscounted ballots. It turned out Premier knew of this for four years but failed to disclose it.

In August I spoke about how Premier Election Systems, again, patched a security weakness in its vote tabulation software.

This time around the specific flaw failed to record significant events that occurred — like matters as important as the deletion of votes, both during and after an election. There is no way that Premier can prove vote tampering has not taken place in elections using its system because its software simply did not record this activity.

Premier Election Systems is practically a billboard for the necessity of open source voting systems. Only open source can offer confidence through the accountability and verifiability which transparency offers. A closed source system can never be trusted wholeheartedly because its inner operations are a mystery.

In fact, the whole matter of electronic voting is widely used as an argument for why open source software exists in the first place and its philosophical raison d'être. You'll hear this presented at any Software Freedom Day event but it is neither mere rhetoric nor a clichéd example because it is a real concern, as proven by even just the two instances above of faults in closed source systems.

Consequently, I am an advocate of open source in electronic voting systems. Yet, at the same time, I have to be logical and realistic.

Mokurai's posting on DailyKos caught my attention due to my interest in the topic. Yet, this time the fail is not on the vendor's side.

Let me explain, with just a touch more technical proficiency than the DailyKos.

Well-intentioned Election Defense Alliance and Open Voting Consortium member, Jim March, made the database available to all through a Wiki site dubbed "study Sequoia";

March recognised the data as a Microsoft SQL Server database backup but, it seems, had no access to SQL Server, or any proficiency in the product.

He first claims Sequoia "vandalized the data as valid databases by stripping the MS-SQL header data off, assuming that would stop us cold";

Elsewhere, March explicitly says the database "has so far proven unreadable in any database";

Yet, this is not the case. The file is a valid SQL Server backup and I was able to restore it with ease and without problem on my laptop.

Those with access to SQL Server 2005 or 2008 can easily repeat my steps.

First, download and unzip the database.

Next, create a new database in SQL Server through the Management Studio console. It does not matter what the database is named. When creating the database, add a second .mdf file for rows as part of the primary filegroup.

Right-click on the database and choose Tasks / Restore / Database.

In the restore dialog set the source as a device (instead of a database) and browse to the unzipped .bak file. Tick the restore box in the backup set list.

Don't click OK yet, or the restore will fail. Click the Options page in the left-hand pane. Check the option to overwrite the existing database (that is, the blank one you created.)

Note that there are three files being restored, with two being row data. Click the &ldquo;...&rdquo; icon on the second set of rows data and browse to the second .mdf file you added above when creating the database.

Now you can click OK and the database will restore successfully to your live SQL Server system where you can inspect its contents.

This restore proceeds without failure, and it is clear Sequoia have not &lsquo;vandalized&rsquo; the binary backup file in any way.

Failing to achieve the same result himself, March proceeded to use Linux tools to attempt to make the database yield its secrets.

His primary tool was the command-line strings command which picks out sequences of readable text from a binary file, omitting anything which is not printable ASCII.

What March saw was sequences of SQL code, the industry standard structured query language used by database companies worldwide for manipulating and querying databases.

March makes several claims about this.

First, he says, the SQL code &ldquo;appears to control the logical flow of the election.&rdquo;

Second, he claims the existence of code within a database violates the United States Federal Election Commission&rsquo;s regulations which ban the use of interpreted code in voting machines.

Specifically, this regulation is part of stringent security that demands the program cannot be modified while executing, which an interpreted language is able to do more readily than a compiled one.

Nevertheless, this regulation is not a regulation and is not a law. It comes from section 5.2.2 of the Election Assistance Commission (EAC)&rsquo;s voluntary voting systems guidelines which states can choose to adopt.

Being a plain old SQL Server database it&rsquo;s clear there was not going to be (say) shell scripts or Python or Perl programs being run, but I did expect to find stored procedures, a database mechanism which permits a measure of program code to be stored within the database and executed by the database server. Could these be the SQL code to which March referred, the code that controls the flow of the election while it is ongoing?

Well, in a nutshell, no.

At first glance, it appears there is no code within the database at all. Browsing the appropriate section in SQL Server's Management Studio shows no stored procedures, views or triggers at all.

Yet, Jim March's output from the Linux strings command yielded actual SQL statements. The examples he posted were nothing unusual despite the odd inference March drew that they somehow controlled the flow of the election.

Rather, they appeared to be nothing more than regular old maintenance scripts which create the tables in the first instance and define the fields.

With some work the same code can be found. March stated that Sequoia redacted the database before sending it to EDS and it seems the nature of this redaction was not to 'vandalize' the binary backup file but instead to drop all the programmability.

Unfortunately for Sequoia their database administrator did not think to compact the database before backing it up. SQL Server - like most databases - does not automatically shrink its files.

Like your computer's hard disk a database does not truly remove data when deleted but instead marks the space it held as free for re-use. Thus, the code remained in the database although invisible to Management Studio and can be inspected via other means.

Nevertheless, the code does not perform any nefarious task but simply serves to create the 88 database tables found within, with names like AUDIO, BALLOT\_CONTEST, BALLOT\_CONTEST\_POSITION, BALLOT\_PRECINCT, BALLOT\_STYLE, CANDIDATE, CONTEST, EVENT\_LOG, LAYOUT, PROVISIONAL\_VOTE, REGISTRATION, TALLY\_BLANK\_BALLOT, VOTER and so forth.

So, the structure of the database file is uninteresting - being untampered with. The programmability content of the database is uninteresting. What's more, the data content of the database is similarly uninteresting.

Within table VOTER we find records of voters but with such non-identifying fields as VOTER\_ID, SERIAL\_NUMBER and PRECINCT\_ID. I can tell you that VOTER\_ID 885 has SERIAL\_NUMBER 41970 and is in PRECINCT\_ID 594 but that doesn't tell me who the person is or who they voted for, or even if they voted at all.

Mokurai's DailyKos story has hit the Internet with prominent sites like SlashDot featuring it but yet the claims raised are sensational and baseless.

While I can respect the very good intentions of the EDS and Jim March there actually is no grounds for the criticism being levied against Sequoia.

The database file has not been vandalised, and the fact March couldn't restore the database should have tipped

him off from the start he didn't actually have the technical literacy to analyse what had been supplied.

I'm bound to receive criticism for this article by open source voting advocates. Yet, we can't advance a cause by putting forth false accusations.

There are many good reasons why voting system ought not to be closed source without embarrassing ourselves by making others up.