

On Vulnerabilities in the 2024 Nova Scotia Municipal Election

Andrew Cautisanu^[0009-0001-0439-871X] and Aleksander
Essex^[0000-0002-0228-0371]

Western University, Canada
{acautisa, aessex}@uwo.ca

Abstract. This paper presents a cybersecurity study of online voting deployments in the 2024 municipal elections of Nova Scotia, Canada. We identified 44 municipalities offering online voting, of which 43 used a common vendor. Our study identified over 20 CVEs on this vendor’s server around the time of the election, including CVE-2023-38408 (CVSS 9.8), which was publicly disclosed for over a year before the election. Following our vulnerability disclosure, the affected vendor updated software components, mitigating many (but not all) of the identified CVEs. We then compare these findings against the requirements in a recently published national standard for municipal online voting and a provincial report on the agreed-upon procedures for the election. Our analysis identifies gaps in continuous vulnerability monitoring and supply chain oversight, as well as limitations in audit and reporting practices. We conclude with recommendations for immediate mitigation, updates to the standard, and integration of real-time threat intelligence to strengthen the security and resilience of future online elections.

Keywords: E-Voting · Online Elections · Software Vulnerabilities

1 Introduction

Online voting offers the potential to enhance accessibility, convenience, and participation in municipal elections, yet it also introduces complex cybersecurity risks to election integrity, confidentiality, and availability. The 2024 Nova Scotia municipal election serves as a timely case study for evaluating these risks in a real-world context. This paper examines the online voting infrastructure, operational procedures, and security practices of the election, focusing on identifying gaps, vulnerabilities, and opportunities to improve the resilience of future elections.

The primary lens for this evaluation is the CAN/DGSI 111-1:2024 [1], which was published just *after* the election. This national standard outlines security, auditability, accessibility, and management requirements for municipal online voting systems. Halifax Regional municipality (HRM), the largest city in Nova Scotia, commissioned a report by Ernst & Young (EY) [3] outlining the agreed-upon procedures for the election. We also consider HRM’s Municipal Elections

Project Review [4] against the DGSi standard, highlighting areas of compliance, partial implementation, and notable deficiencies.

We observed over 20 Common Vulnerabilities and Exposures (CVEs) present on election servers around the time of the election. Given the scope and relevance, we focus our analysis on the 6 most severe CVEs: CVE-2023-38408, CVE-2023-51767, CVE-2023-44487, CVE-2021-41617, CVE-2021-23017, and CVE-2021-3618. This narrower focus allows us to provide a more concrete illustration of how real-world vulnerabilities could translate into threats, and how they relate to DGSi compliance gaps.

The paper proceeds as follows: Section 2 provides an overview of the Nova Scotia municipal election, the DGSi standard, and a census of which cities used online voting and which vendors. Section 3 summarizes findings from the EY report [3] and the municipal project review [4]. Section 4 explores the most severe CVEs and their implications, mapping them to potential gaps against DGSi requirements. Section 5 discusses remediation strategies, recommendations for future elections, and proposed enhancements to the DGSi standard to improve election security, resilience, and trustworthiness. Finally, Section 6 discusses details of our vendor disclosure.

2 Background

2.1 The 2024 Nova Scotia Municipal Election

The 2024 Nova Scotia municipal election marked a significant milestone in the adoption of online voting across Canadian municipalities. Administered by Intellivote, the election used a centralized digital platform that allowed eligible voters to cast ballots electronically from personal devices or designated municipal voting stations. The election encompassed multiple municipalities, each responsible for coordinating voter registration, authentication, and vote tabulation. The system aimed to improve accessibility, particularly for remote or mobility-impaired voters, while also providing faster vote counting and reporting.

At a high level, the online voting system followed a client-server architecture model typical of modern web-based election platforms. Voters access the system through a web interface using standard browsers on personal or public devices. Authentication is performed using credentials distributed through voter information packages. Once authenticated, voter selections were transmitted over encrypted connections to centralized application servers that handled ballot processing, session management, and transaction validation. These application servers interface with backend database systems that store voter eligibility status, ballot definitions, and encrypted vote records.

2.2 City/Vendor Census

To identify which municipalities participated in online voting and the associated vendor for each, we conducted an intensive manual web-based search, similar to

the methodology used in prior research [7,2]. A table of our findings is given in Appendix A.

Of Nova Scotia’s 49 municipalities, we determined that 44 offered an online voting option. HRM worked with Montreal-based Simply Voting, while the other 43 municipalities partnered with Intelivote, a well-known vendor in the province that has delivered online elections since it was first introduced in 2008. According to Simply Voting’s report, Halifax (Nova Scotia’s largest municipality) saw 123,529 voters cast ballots, representing a 36.8% turnout. Of these, 62% voted online, with the remainder casting paper ballots [17].

2.3 The DGSI Standard

The CAN/DGSI 111-1:2024 (DGSI) standard provides comprehensive guidance for the secure design, deployment, and administration of online voting systems in Canada [1]. It establishes benchmarks for both technical and procedural robustness to ensure election integrity and public trust. The standard emphasizes system security and data protection, requiring measures such as end-to-end encryption, timely patch management, rigorous access control, and safeguards against both insider and external threats. These requirements are intended to reduce the risk of unauthorized access, data exfiltration, and system compromise, which could undermine the legitimacy of the election.

Voter identity and authentication processes are mandated to ensure that each vote is cast legitimately and counted accurately. The standard further specifies mechanisms for auditability and transparency, including maintaining immutable system logs, conducting logic and accuracy testing, and enabling independent verification of results. These measures allow election administrators and observers to confirm the integrity of both the voting process and the final outcomes.

Secrecy and privacy of the vote are also central principles, with ballots required to remain anonymous and all voter information protected against re-identification. DGSI addresses accessibility and usability by mandating inclusive interface designs that comply with legal accessibility standards, ensuring equitable participation for voters with varying abilities. The standard also provides guidance on network capacity, redundancy, and contingency planning to maintain service availability during peak loads and mitigate risks from denial-of-service attacks, power failures, or network outages.

Finally, DGSI outlines operational and administrative requirements, including staffing protocols, subcontractor oversight, and comprehensive risk assessments to identify vulnerabilities before and during the election. By codifying these best practices, DGSI provides both a technical and procedural benchmark, enabling evaluations of elections, such as the 2024 Nova Scotia municipal election, to determine compliance, identify potential gaps, and guide improvements in online voting security and reliability.

3 Review of Audit and Project Documents

Two independent documents were examined as part of this research: the Ernst & Young (EY) Agreed-Upon Procedures (AUP) report [3] and the 2024 Municipal Elections Project Review [4]. The EY report focuses on procedural compliance with election protocols, while the Project Review addresses operational planning, system deployment, and post-election assessment.

While the two documents differ in scope, their findings are complementary, and their limitations are shared.

3.1 Procedural findings

Both documents confirm that core election functions were carried out in accordance with established guidelines. The EY report verified that voter authentication procedures ensured only eligible voters could access the online system, that ballot submissions were accurately recorded and consistently logged, and that exported election results were compared across multiple copies to confirm accuracy and integrity. Record management practices—including encryption, secure storage, and formal handoff of election data to the returning officer—were also assessed and found to be compliant [3]. The Project Review corroborates these findings and further documents operational planning, system deployment logistics, and post-election process evaluation, confirming that appropriate logging and monitoring were in place throughout the voting period [4].

3.2 Shared limitations

Despite their procedural thoroughness, neither document addresses the technical security of the underlying election infrastructure. Their scopes are limited to verifying that prescribed processes were followed. They do not evaluate software configurations, patch levels, exposed network services, or susceptibility to known vulnerabilities. No penetration testing, vulnerability scanning, or source-code review was conducted as part of either assessment. Additionally, no independent test environments or staging systems were available to support external verification of the vendor’s security claims.

As a result, while the documents together provide strong evidence of administrative oversight and procedural due diligence, they are insufficient to ensure that the election infrastructure is protected against cyber threats. This gap between procedural assurance and technical security is particularly significant given the high-severity CVEs identified in Section 4, motivating the need for the technical analysis that follows.

4 Observed CVEs and Potential Impacts

Building on the technical context of the previous section, it is essential to examine specific vulnerabilities relevant to the systems observed during the Nova Scotia

municipal election. The presence of known CVEs in a system represents a serious security risk, as these vulnerabilities have publicly documented exploits that attackers can leverage if unpatched.

4.1 Methodology and Timeline

The voting period took place from October 7-19, 2024. On October 17th, we looked up the two voting services (Simply Voting and Intelivote) on `shodan.io`, a website that performs routine scans of hosts in the IP space, collecting and logging service banners and metadata. No CVEs were observed on the Simply Voting server. The Intelivote server, however, showed 21 CVEs with severity scores ranging from 2.8 (low) to 9.8 (critical). We did not perform a vulnerability disclosure at that time on the basis that (a) our observation came far too late for any mitigations to be completed and tested during the remaining two days of the election period, (b) the presence of these CVEs was already publicly known (via `shodan`, at minimum), (c) the newest CVEs were over a year old at the point (with the oldest dating back as far as 2008). We did note CVE-2023-38408, a critical vulnerability affecting OpenSSH versions prior to 9.3p2. We performed an `nmap` version scan of the server and independently confirmed that it was running an affected version (7.4). We spent the period between the election and the eventual disclosure conducting a (slow but steady) analysis of these CVEs in an election context. At the conclusion of our analysis, we noted that the affected software version (and, by extension, the CVEs) remained unchanged.

4.2 Overview of CVEs

Although numerous vulnerabilities were observed, we only report in detail on six CVEs based on our assessment of their importance to the infrastructure and election context. In this section, we focus on six CVEs, selected for their severity and relevance to software and services identified via `shodan.io` scans of public-facing network interfaces.

1. CVE-2023-38408 affects certain implementations of the SSH protocol and has a CVSS score of 9.8, marking it as critical and thus demanding more attention than other vulnerabilities. The vulnerability allows an attacker to bypass authentication under specific configurations. Its high CVSS score reflects the combination of ease of exploit, potential for complete system compromise, and the lack of prerequisite conditions beyond network access [11]. Exploitation can permit unauthorized administrative access, making the vulnerability extremely impactful. The scoring also accounts for the broad range of affected SSH versions and systems.

Based on CVSS 3.1, it has an impact score of 5.9 and an exploitability score of 3.9, resulting in a severe score of 9.8. One attack vector is through knowledge of the operating system. If the attacker is given the correct libraries for the vulnerable Ubuntu version, the exploit is a straightforward list of commands. However, in the real world, for an attacker to gain this information, it would be difficult

to determine using nmap alone, as it is not always reliable for version detection. Despite this, only two OS versions have been found to contain exploitable libraries: Ubuntu 22.04 and 21.10 [15]. While these are the only OS versions currently found, any Ubuntu version below 22.04 can be exploited, as they use vulnerable OpenSSH versions. This is because determining which libraries cause this vulnerability on the affected operating systems is a lengthy process that requires substantial knowledge of OpenSSH’s internal workings. While it is a difficult attack vector, if enough recon is done, it is a relatively straightforward exploit, which is most likely why it is assigned the Low Attack Complexity as part of the 3.9 exploitability score.

A major aspect of the overall 9.8 score is that no special privileges are required for the attack. This is likely because one of the prerequisites for this attack is “access to a server with ssh-agent forwarding enabled” [5]. While it is an easy requisite to write, achieving it is another process entirely. Most servers are protected at minimum with a password and at most with a fingerprint, which are not easy to brute-force or bypass. This leads to the conclusion that this exploit is tailored for an attacker with inside access to the server they are trying to exploit, meaning that a disgruntled worker or someone paid off within the company could easily launch it.

2. CVE-2023-51767 affecting OpenSSH through version 10.0, carries a CVSS score of 7.0. The vulnerability involves a potential row hammer attack on DRAM that may allow authentication bypass, as the `mm_answer_authpassword` value does not resist single-bit flips. Exploitation requires co-location of attacker and victim with user-level privileges, limiting the threat to specific scenarios [13]. The moderate CVSS rating reflects the attack’s complexity and limited applicability, and while some dispute its relevance, it highlights a potential weakness in authentication resilience.

3. CVE-2023-44487 with a CVSS score of 7.5 affects HTTP/2 implementations. Malformed HTTP/2 requests can trigger server errors or denial-of-service conditions [12]. The score reflects moderate impact and exploitability, as the vulnerability requires a crafted sequence of requests, but can be executed remotely. It demonstrates how protocol handling errors can destabilize servers even without direct code execution.

4. CVE-2021-41617 involves specific SSH server configurations and carries a CVSS score of 7.0. It allows potential privilege escalation or authentication bypass [10]. The scoring reflects the moderate complexity of exploitation, combined with the significant impact if administrative access is obtained. The vulnerability highlights weaknesses in SSH access control and privilege separation mechanisms.

5. CVE-2021-23017 affects the NGINX resolver and has a CVSS score of 7.7. Improper handling of certain DNS queries can lead to server crashes [8]. The CVSS score accounts for the relatively high complexity of triggering the vulner-

ability and the possible systemic effects on service availability. It underscores the criticality of properly configuring and updating server components to prevent remote exploits.

6. CVE-2021-3618 the ALPACA cross-protocol attack affecting libcurl, carries a CVSS score of 7.4. It stems from TLS servers using compatible certificates across different protocols, allowing a network attacker to redirect a client’s connection to another protocol while still establishing a valid TLS session [9]. This breaks TLS authentication and can compromise data integrity and confidentiality. Exploitation requires network access and depends on specific certificate and protocol configurations, which is why the CVSS rating is high but not critical. The vulnerability could facilitate broader attacks, especially if combined with other system flaws.

4.3 Potential Impact on Election Systems

Following the technical overview of the CVEs, it is important to consider how each vulnerability could directly impact the integrity, availability, and confidentiality of an election system.

1. **CVE-2023-38408** could allow an attacker to gain unauthorized administrative access to election servers. To launch an attack on these servers, one would first need to have access to the servers. This could easily be obtained, for example, by a low-level DevOps worker or contractor. From here, the worker, whom we will call Alice, would simply need to check if the server has the required libraries for the exploit. From this point, Alice would need to run some specific commands to make the PKCS#11 stack executable and exploit the server. Alice could run code, manipulate files and more under the mask of the attacked machine [5]. Examples include, but are not limited to: privilege escalation, running malicious code, denial-of-service attacks, tampering with vote submission processes, manipulation of tallies, deletion of stored ballots, or denial of voters’ ability to vote.

We note this attack scenario is hypothetical, and there have been no known attacks attributed to this exploit. Additionally, in this specific example, the attacker would need to find other libraries than the publicly available ones that are referenced in the exploit’s documentation, as those libraries are specific to Ubuntu 21.10 and 22.04 [15]. Based on the OpenSSH versioning provided by shodan.io, which was 7.4 [16], it can be theorized that they were using an OS older than the ones mentioned.

2. **CVE-2023-51767** could be exploited to interfere with server operations or escalate privileges once initial access is gained [13]. If exploited within election infrastructure, this could allow an attacker to modify system processes, disrupt vote-processing services, or deploy additional malicious tooling to maintain persistence on election servers. In a high-availability election environment, even limited command execution could enable manipulation of logging, monitoring, or backend election services that is difficult to detect in real time.

3. **CVE-2023-44487** could result in denial-of-service conditions [12], rendering online voting portals temporarily inaccessible. Such outages could reduce voter participation, create confusion, or delay vote submission, allowing attackers to selectively disrupt service while retaining the ability to manipulate votes or monitoring systems. While manipulation is a major security concern in online elections, accessibility vulnerabilities, like the aforementioned CVE, are equally concerning. An example of this occurred in the 2021 New South Wales election, where 143 ballots were not counted [6] after the election server was disrupted by two incidents on polling day.
4. **CVE-2021-23017** and **CVE-2021-41617** could allow privilege escalation or unauthorized access [8] [10]. These vulnerabilities increase the risk that internal processes, such as vote tallying or ballot storage, could be altered undetected.
5. **CVE-2021-3618** could compromise the confidentiality of voter communications or election data in transit [9]. Although less directly impactful on server-side operations, it could facilitate man-in-the-middle attacks, especially if combined with other vulnerabilities that grant administrative access.

The presence of multiple vulnerabilities amplifies risk, potentially resulting in the (concurrent) compromise of multiple layers of the election system, including voter authentication, ballot submission, tallying, and the integrity of stored results. Even if mitigated individually, their simultaneous presence in an interconnected infrastructure poses compounded threats, underscoring the critical importance of comprehensive patching and system hardening.

4.4 Relation to DGSi Standards

The identified CVEs highlight critical conflicts with multiple requirements in the DGSi standard [1], revealing potential risks to the security and integrity of the election system.

1. **CVE-2023-38408** directly affects Section 5.2.2 on vote authentication, which mandates that only eligible voters may cast votes and that administrative access must be protected. Exploitation could allow attackers to access administrative interfaces, manipulate vote records, or alter ballot tallies, violating strict access control and vote integrity requirements.
2. **CVE-2023-51767** affects Sections 4.1 and 5.2.2, which require secure server-side systems, role-based permissions, and prevention of unauthorized execution. This vulnerability could bypass access restrictions and permit manipulation of vote processing systems.
3. **CVE-2023-44487** could result in denial-of-service conditions. This impacts Section 10.1.3, which requires sufficient network capacity and resilience to ensure voters can cast ballots without interruption. Exploitation could prevent voters from accessing the system and interfere with election continuity.
4. **CVE-2021-23017**, could allow an attacker to forge UDP packets from a DNS server to trigger a 1-byte memory overwrite. This could crash worker

- processes, resulting in service disruption or other system impacts. It directly violates Sections 10.1.1–10.1.3 and 4.1.1.a–c, which require continuity of service and mitigation against external threats and denial-of-service conditions.
5. **CVE-2021-41617** could allow unauthorized access if exploited, violating Sections 4.1.1 and 4.1.11. This could compromise vote storage, submission, or tallying.
 6. **CVE-2021-3618** impacts Sections 4.1.8 (end-to-end encryption) and 7.1.1 (secure handling of voter data). Exploitation could enable the interception or manipulation of voter communications, undermining confidentiality and voter trust.

Collectively, these CVEs demonstrate multiple divergences from DGSI’s guidance on cumulative risk management, patch enforcement, and specific mitigation strategies for high-severity software vulnerabilities, highlighting that compliance with general standards does not guarantee protection against known critical exploits.

4.5 Gaps in EY and Project Review Documentation

A review of the EY report and the 2024 Municipal Elections Project Review and Next Election Document indicates that while procedural diligence is described in areas such as voter authentication, ballot submission, vote tallying, and record management [3], there are notable gaps when viewed against the DGSI standards and the presence of high-severity CVEs. Neither document explicitly addresses systematic vulnerability management or continuous monitoring of publicly exposed services, leaving potential blind spots in identifying unpatched critical vulnerabilities such as CVE-2023-38408 and CVE-2023-44487.

The EY report emphasizes adherence to procedural controls but does not provide detailed guidance on defending against attacks that exploit protocol-level flaws, memory vulnerabilities, or cross-protocol TLS attacks. For instance, vulnerabilities such as CVE-2021-3618, which can compromise encrypted voter communications, are neither anticipated nor mitigated in the documented procedures. Similarly, the Project Review document outlines operational readiness and continuity plans but lacks explicit consideration of cumulative risks from multiple interacting vulnerabilities, which could allow a chain of exploits to compromise authentication, ballot integrity, and system availability.

Neither document references vulnerability discovery techniques, vulnerability scans, or the use of CVE tracking to proactively identify exposures in network-facing components. The documentation also does not include remediation timelines, patch enforcement procedures, or validation steps for high-severity vulnerabilities, which are critical for compliance with DGSI’s guidance on access control, server security, and data confidentiality.

While the reports provide robust procedural frameworks, gaps exist in the proactive identification and mitigation of technical vulnerabilities, monitoring for emerging threats, and formal integration of software security assessments into the election lifecycle. These gaps highlight areas where DGSI standards

could be supplemented to ensure alignment between operational procedures and contemporary cybersecurity risk management practices.

5 Recommendations and Next Steps

5.1 Immediate Fixes

Table 1 summarizes the recommended remediation actions for each CVE discussed in Section 4. In all six cases, the primary fix is to update the affected component to a patched release. Where a patch alone is insufficient, additional configuration hardening is noted. Intelivote should prioritize these updates by CVSS severity, starting with CVE-2023-38408.

Table 1. Recommended remediation actions for identified CVEs.

CVE	Component	CVSS	Remediation
CVE-2023-38408 [11]	OpenSSH	9.8	Update SSH; enforce strict <code>ssh-agent</code> forwarding policies and disable unused authentication methods.
CVE-2023-51767 [13]	OpenSSH	7.0	Update OpenSSH; enable server-level memory protections (e.g., ECC RAM) and monitor co-located environments.
CVE-2023-44487 [12]	HTTP/2	7.5	Patch web server; validate HTTP/2 request parsing and deploy rate-limiting to mitigate rapid-reset DoS.
CVE-2021-41617 [10]	OpenSSH	7.0	Update OpenSSH; harden <code>sshd_config</code> with role-based permissions, MFA, and strict privilege separation.
CVE-2021-23017 [8]	NGINX	7.7	Update NGINX; apply recommended DNS resolver settings to prevent worker-process crashes.
CVE-2021-3618 [9]	libcurl / TLS	7.4	Update libcurl; enforce per-protocol certificate isolation and strict TLS validation across all endpoints.

Beyond individual patches, two practices would prevent similar exposures in future elections: (1) adopting an automated patch-management pipeline with a defined service-level agreement (e.g., critical CVEs patched within a pre-defined period following initial disclosure), and (2) conducting periodic external vulnerability scans to verify that public-facing services remain free of known CVEs.

5.2 DGSi Additions and Improvements

The identified CVEs reveal areas where the current DGSi standard could be enhanced to better address high-severity vulnerabilities. The standard should

explicitly include requirements for timely patch management and software version control, ensuring that all critical components, such as SSH, HTTP/2 servers, NGINX, and libcurl, are regularly updated to mitigate known vulnerabilities.

DGSI could benefit from enhanced guidance on cumulative risk assessment, emphasizing how multiple moderate-severity vulnerabilities can interact to create more severe threats, as demonstrated by the potential combination of SSH and network-level attacks. The standard should include specific configuration hardening standards for server software, covering secure SSH configurations, memory protection measures, and HTTP/2 request validation. This would support compliance with existing DGSI sections on access control, vote integrity, and service availability, while providing more concrete technical guidance to prevent exploitation. Fourth, DGSI should strengthen TLS and encryption requirements, mandating strict certificate validation, protection against cross-protocol attacks such as ALPACA, and guidance for verifying end-to-end encryption for all voter communications.

Finally, the standard could incorporate continuous monitoring and vulnerability scanning protocols, including publicly accessible tools or internal assessments similar to shodan.io, to identify exposed services and proactively detect vulnerabilities before an election. These improvements would provide a more robust, prescriptive approach, ensuring that election providers like Intelivote can effectively address known CVEs while maintaining compliance with DGSI standards.

5.3 Future Work

Looking beyond immediate fixes and DGSI improvements, future work should focus on long-term resilience and adaptability of the election system, including:

1. Integrating automated vulnerability management and real-time threat intelligence, enabling election administrators to continuously monitor for emerging CVEs and prioritize patching based on both severity and likelihood of exploitation.
2. Establishing formal red-teaming protocols that simulate multi-vector attack scenarios, validating both technical defences and procedural safeguards against sophisticated attacks targeting voter authentication, ballot submission, tallying, and data retention.
3. Ongoing research exploring cryptographic enhancements and multi-layered authentication mechanisms for online voting, including multi-factor authentication, hardware security modules, and end-to-end verification (E2E-V), strengthening DGSI compliance while addressing identified gaps.
4. Expanding training and public transparency initiatives. Election administrators, technical staff, and observers need updated guidance on emerging vulnerabilities and mitigation techniques. Greater transparency can improve voter confidence by demonstrating that proactive measures are in place to maintain election integrity.

Collectively, these efforts ensure that lessons from current CVEs inform the long-term evolution of secure online voting practices.

6 Vendor Disclosure

We conducted a vulnerability disclosure to Intelivote in January 2026, following the completion of our analysis. We informed them of the presence of numerous CVEs on their servers, summarized the most significant ones, and provided instructions for viewing the complete list on Shodan. We explained that most of these CVEs could be mitigated by simple software updates. We concluded with an offer to have a follow-up discussion of details and possible mitigations.

At the time of writing, we have not received a response. However, beginning four days after our disclosure, we observed they made several software updates to their servers, reducing their total CVE exposure from over 20 to only four. Of these, three were listed in our disclosure: CVE-2023-44487 [12], CVE-2021-23017 [8], and CVE-2021-3618 [9]. The fourth, CVE-2025-23419 [14] was discovered after the 2024 election. Given that these updates were made within days of contacting them after a prolonged presence on their servers, we take it to be an implicit acknowledgement of our disclosure.

7 Conclusion

This paper examined the security of the Nova Scotia municipal election system against the DGSi online voting standard. By analyzing six high-severity CVEs, the discussion highlighted how known vulnerabilities can directly affect voter authentication, ballot submission, and vote tallying. The review revealed gaps in both the DGSi standards and existing election documentation. While the DGSi standard provides comprehensive guidance on access control and auditability, it lacks detailed provisions for cumulative vulnerability management and patch enforcement. Maintaining secure elections requires a holistic approach combining procedural diligence, technical mitigation, and proactive risk management. Addressing both known and emerging vulnerabilities is essential to protect election integrity in increasingly digital environments.

Acknowledgements. The authors thank the reviewers for their feedback. This work was supported by a National Science and Engineering Council of Canada (NSERC) Discovery Grant.

References

1. Online Voting – Part 1: Implementation of Online Voting in Canadian Municipal Election (CAN/DGSI 111-1). Digital Governance Standards Institute (2024)
2. Cardillo, A., Akinyokun, N., Essex, A.: Online Voting in Ontario Municipal Elections: A Conflict of Legal Principles and Technology? In: Electronic Voting: International Joint Conference (E-Vote-ID). Lecture Notes in Computer Science, vol. 11759, pp. 67–82 (2019)
3. Ernst & Young: Halifax regional municipality agreed-upon procedures report. Tech. rep., Ernst & Young (2025), <https://www.halifax.ca/sites/default/files/documents/city-hall/elections/2024-election-ey-report.pdf>
4. Halifax: Project Review & Advancing to the Next Election (2024), URL: https://www.halifax.ca/sites/default/files/documents/city-hall/elections/2024_municipalelections_projectreviewandnextelectiondocument_8.5x11-web.pdf
5. Jakaba: Exploring OpenSSH’s Agent Forwarding RCE (CVE-2023-38408) (March 28th, 2024), URL: <https://www.vicarius.io/vsociety/posts/exploring-opensshs-agent-forwarding-rce-cve-2023-38408>
6. Kelly Fuller: NSW council elections hang in the balance as court hears iVote failure caused by ‘defect or irregularity’. ABC (February 22nd, 2022), URL: <https://www.cbc.ca/news/canada/nova-scotia/election-nova-scotia-voting-email-scam-fraud-1.7370936>
7. Klassen, E., Brunet, J., Goodman, N., Essex, A.: Credential attacks in ontario’s online elections. In: International Joint Conference on Electronic Voting (E-Vote-ID). Lecture Notes in Computer Science, vol. 16028, pp. 141–157 (2025)
8. NIST: CVE-2021-23017 Detail (2021), URL: <https://nvd.nist.gov/vuln/detail/CVE-2021-23017>
9. NIST: CVE-2021-3618 Detail (2021), URL: <https://nvd.nist.gov/vuln/detail/CVE-2021-3618>
10. NIST: CVE-2021-41617 Detail (2021), URL: <https://nvd.nist.gov/vuln/detail/CVE-2021-41617>
11. NIST: CVE-2023-38408 Detail (2023), URL: <https://nvd.nist.gov/vuln/detail/cve-2023-38408>
12. NIST: CVE-2023-44487 Detail (2023), URL: <https://nvd.nist.gov/vuln/detail/CVE-2023-44487>
13. NIST: CVE-2023-51767 Detail (2023), URL: <https://nvd.nist.gov/vuln/detail/CVE-2023-51767>
14. NIST: CVE-2025-23419 Detail (2025), URL: <https://nvd.nist.gov/vuln/detail/cve-2025-23419>
15. Qualys Security Advisory: CVE-2023-38408: Remote Code Execution in OpenSSH’s forwarded ssh-agent (July 19th, 2023), URL: <https://www.openwall.com/lists/oss-security/2023/07/20/1>
16. Shodan.io: Shodan.io host of 140.238.135.104 (2025), URL: <https://www.shodan.io/host/140.238.135.104>
17. Simply Voting: Successful Online Voting Solutions for the 2024 Halifax Regional Municipal Election (November 26th, 2024), URL: <https://www.simplyvoting.com/halifax-elections-2024/>

A Online Voting Adoption by Municipality

Table A shows the online voting adoption status the vendor affiliation of Nova Scotia’s municipalities in the 2024 election.

Municipality	Municipal Type	Voted Online	Vendor
Amherst	Town	Yes	Intelivote
Annapolis Royal	Town	Yes	Intelivote
Annapolis	County municipality	Yes	Intelivote
Antigonish	Town	Yes	Intelivote
Antigonish	County municipality	No	N/A
Argyle	District municipality	Yes	Intelivote
Barrington	District municipality	Yes	Intelivote
Berwick	Town	Yes	Intelivote
Bridgewater	Town	Yes	Intelivote
Cape Breton	Regional municipality	Yes	Intelivote
Chester	District municipality	Yes	Intelivote
Clare	District municipality	Yes	Intelivote
Clark’s Harbour	Town	No	N/A
Colchester	County municipality	Yes	Intelivote
Cumberland	County municipality	Yes	Intelivote
Digbyes	District municipality	Yes	Intelivote
Digbyes	Town	Yes	Intelivote
East Hants	District municipality	Yes	Intelivote
Guysborough	District municipality	Yes	Intelivote
Halifax	Regional municipality	Yes	Simply Voting
Inverness	County municipality	Yes	Intelivote
Kentville	Town	Yes	Intelivote
Kings	County municipality	Yes	Intelivote
Lockeport	Town	No	N/A
Lunenburg	District municipality	Yes	Intelivote
Lunenburg	Town	Yes	Intelivote

Table A. Municipality and Online Voting Adoption Status

(Table continued from previous page)

Municipality	Municipal Type	Voted Online	Vendor
Mahone Bayes	Town	Yes	Intelivote
Middleton	Town	Yes	Intelivote
Mulgrave	Town	No	N/A
New Glasgow	Town	Yes	Intelivote
Oxford	Town	Yes	Intelivote
Pictou	Town	Yes	Intelivote
Pictou	County municipality	Yes	Intelivote
Port Hawkesbury	Town	Yes	Intelivote
Queens	Regional municipality	Yes	Intelivote
Richmond	County municipality	No	N/A
Shelburne	District municipality	Yes	Intelivote
Shelburne	Town	Yes	Intelivote
St. Mary's	District municipality	Yes	Intelivote
Stellarton	Town	Yes	Intelivote
Stewiacke	Town	Yes	Intelivote
Trenton	Town	Yes	Intelivote
Truro	Town	Yes	Intelivote
Victoria	County municipality	Yes	Intelivote
West Hants	Regional municipality	Yes	Intelivote
Westville	Town	Yes	Intelivote
Wolfville	Town	Yes	Intelivote
Yarmouth	District municipality	Yes	Intelivote
Yarmouth	Town	Yes	Intelivote

Table A. Municipality and Online Voting Adoption Status