A hackers' guide to online voting systems

BY INTIGRITI · MARCH 5, 2024 · LAST UPDATED ON MARCH 6, 2025

In today's digital world, online voting systems are pivotal in various domains. Businesses rely on them for award shows where the public's vote determines winners. Music charts use online voting to influence album sales, shaping the music industry. Even self-driven communities depend on voting to maintain democratic processes. However, ensuring the fairness and integrity of these systems is a complex task.

The methods used to ensure fairness vary and often involve balancing usability and security. In this blog post, we'll delve into techniques for manipulating online voting systems, focusing on attacks that go beyond classic web vulnerabilities, and distinguishing between two common security contexts: anonymous polls and authenticated polls.

Anonymous polls

Anonymous polls utilize identifiers like cookies, IP addresses, or email/phone verification. However, these methods aren't foolproof. For instance, email or phone numbers can be manipulated using tricks like catch-all wildcards, allowing individuals to vote multiple times unfairly.

Cookie-based

Most anonymous polls place a cookie on the users' machine after they have voted. The presence of the cookie prevents them from casting a vote again, even after they have refreshed the webpage. Clearing cookies or swapping browsers is the easiest way to get across this defense mechanism.

IP-based

In IP-based anonymous polls, the system tracks users' IP addresses to prevent multiple votes from the same device or network. While this method may be more effective in stopping individual attempts to manipulate votes, it also has limitations. Users within the same household or business, sharing the same IP address, may find themselves unable to cast separate votes. However, determined individuals can still bypass this restriction using methods like relying on virtual private networks (VPNs) or IP spoofing to change their true IP addresses and cast multiple votes.

E-mail / phone number verification

While not providing complete anonymity, phone number or email verification serves as a simple barrier to entry in online voting systems. In the event that a voter can only rely on a single e-mail address or phone number for verification purposes, multiple notations may be used to fool the system into thinking the phone number or e-mail address hasn't been used before.

E-mail addresses could use wildcards or comments to achieve this, e.g. e-mails sent to john.smith@gmail.com and john.smith**+1**@gmail.com will both arrive in the same e-mail inbox, despite them looking different to an input validation rule. The same can be done by adding padding zeroes to phone numbers, e.g. +44 13371337 becomes +**0**4413371337, +**00**4413371337, etc.

Authenticated polls

Authenticated polls add an extra layer of security but can still be vulnerable to manipulation. Attackers might attempt to change the weight of a vote or manipulate the quantity of votes cast. For example, they could submit the same vote in different formats, exploiting loopholes in the system. These attacks undermine the integrity of the voting process and threaten the democratic principles online voting seeks to uphold.

Vote weight manipulation

Vote weight manipulation involves altering the weight assigned to voting items within an online voting system. For instance, if users are allowed to select their top three choices with varying weights (e.g., 3 points for the first choice, 2 points for the second, and 1 point for the third), attackers may intercept the request and increase the weight of their preferred choice significantly.

For example:

```
POST /vote

{item : 14, votes : 3}
```

In this request, the user assigns 3 votes to item 14.

```
POST /vote

{item : 14, votes : 3000}
```

In contrast, the attacker manipulates the request to assign 3000 votes to item 14, skewing the results in favor of their preferred choice.

Furthermore, attackers may employ negative votes on other contenders, subtracting votes rather than adding them, to further manipulate the outcome of the voting process. For instance:

```
POST /vote

{item : 15, votes : -3000}
```

This request subtracts 3000 votes from item 15, diminishing its chances of success. These manipulative tactics highlight the necessity for robust security measures to uphold the integrity of online voting systems.

Vote quantity manipulation

Vote quantity manipulation can occur in scenarios where users are limited to voting once for each item, but some form of normalization occurs after this validation. For example, consider a situation where a user wants to vote for item 42. The initial request might look like this:

```
POST /vote

{item : 42}
```

If the user attempts to repeat this request, they will encounter an error message indicating that they have already voted for the item:

```
{status : 409, message: “You’ve already voted for this item”}
```

However, crafty attackers may exploit type casting vulnerabilities to bypass the duplication check while still registering a valid vote. For instance:

- As an exponent:

```
POST /vote

{item : 42e0}
```

- As a string:

```
POST /vote

{item : "42"}
```

- As an explicitly declared positive number:

```
POST /vote

{item : "+42"}
```

- As an array:

```
POST /vote

{item : [42]}
```

In each of these cases, the system accepts the vote despite variations in data type, thereby allowing the attacker to manipulate the voting process surreptitiously. This underscores the importance of robust input validation and stringent duplication checks in safeguarding the integrity of online voting systems.

If the data types are currently normalised prior to validating the votes, or only one vote per account is permitted regardless of what item is voted on, a **race condition attack** could be leveraged to gain extra votes. This occurs when multiple requests are sent simultaneously, overwhelming the system and exploiting the window between the initial validation and the final vote submission, potentially allowing for multiple votes to be cast from a single account.

Wrapping up

In conclusion, the effectiveness and credibility of online voting systems hinge on the diligent application of robust security measures. Prioritizing thorough authentication protocols, rigorous validation procedures, and manual oversight is essential for reinforcing the integrity of the voting process and fostering trust among participants.

REQUEST A DEMO

intigriti.com/demo

VISIT THE WEBSITE

intigriti.com

GET IN TOUCH

hello@intigriti.com