**Problem**

utils.generateUUID, a helper function available in essentially all versions of NodeBB (as far back as v1.0.1 and potentially earlier) used a cryptographically insecure Pseudo-random number generator (Math.random()), which meant that a specially crafted script combined with multiple invocations of the password reset functionality could enable an attacker to correctly calculate the reset code for an account they do not have access to.

The impact of this vulnerability is slightly lessened by the requirement that the **target's email address must be known**, and user emails are protected values in NodeBB. However, since NodeBB can be configured to display email addresses if the admin so wishes, and as email addresses can often by derived from other sources and/or guessed, the impact of this vulnerability is still fairly high.

Math.random()

The Math.random() function returns a floating-point, pseudo-random number that's greater than or equal to 0 and less than 1, with approximately uniform distribution over that range — which you can then scale to your desired range. The implementation selects the initial seed to the random number generation algorithm; it cannot be chosen or reset by the user.

**Note: Math.random() does not provide cryptographically secure random numbers. Do not use them for anything related to security. Use the Web Crypto API instead, and more precisely the window.crypto.getRandomValues() method.**

**src\_utils.js (both)**

**public\_src\_utils.js (v.1.19.x)**

const utils = {

generateUUID: function () {

/\* eslint-disable no-bitwise \*/

return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function (c) {

const r = Math.random() \* 16 | 0;

const v = c === 'x' ? r : ((r & 0x3) | 0x8);

return v.toString(16);

});

/\* eslint-enable no-bitwise \*/

},

// https://github.com/substack/node-ent/blob/master/index.js

decodeHTMLEntities: function (html) {

**public\_src\_utils.common.js (v.2.x)**

generateUUID: function () {

/\* eslint-disable no-bitwise \*/

return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function (c) {

const r = Math.random() \* 16 | 0;

const v = c === 'x' ? r : ((r & 0x3) | 0x8);

return v.toString(16);

});

/\* eslint-enable no-bitwise \*/

}

**Fix**

v2.x patch : <https://github.com/NodeBB/NodeBB/commit/e802fab87f94a13f397f04cfe6068f2f7ddf7888>

v1.19.x patch : <https://github.com/NodeBB/NodeBB/commit/81e3c1ba488d03371a5ce8d0ebb5c5803026e0f9>

**src\_utils.js (both)**

**const crypto =** **require('crypto');**

**module.exports.generateUUID = function () {**

**// from https://github.com/tracker1/node-uuid4/blob/master/index.js**

**let rnd = crypto.randomBytes(16);**

**/\* eslint-disable no-bitwise \*/**

**rnd[6] = (rnd[6] & 0x0f) | 0x40;**

**rnd[8] = (rnd[8] & 0x3f) | 0x80;**

**/\* eslint-enable no-bitwise \*/**

**rnd = rnd.toString('hex').match(/(.{8})(.{4})(.{4})(.{4})(.{12})/);**

**rnd.shift();**

**return rnd.join('-');**

**};**

**public\_src\_utils.js (v.1.19.x)**

**const utils = {**

**generateUUID: function () {**

**// from https://github.com/tracker1/node-uuid4/blob/master/browser.js**

**const temp\_url = URL.createObjectURL(new Blob());**

**const uuid = temp\_url.toString();**

**URL.revokeObjectURL(temp\_url);**

**return uuid.split(/[:\/]/g).pop().toLowerCase(); // remove prefixes**

**},**

**// https://github.com/substack/node-ent/blob/master/index.js**

**decodeHTMLEntities: function (html) {**

**public\_src\_utils.common.js (v.2.x)**

Moved fix to **public\_src\_utils.js**