Eat Spray Love Mobile App Multiple Vulnerabilities

High

← View More Research Advisories

Synopsis

Backdoor Account

Hardcoded into the applications is an administrative backdoor that could allow an attacker to manipulate information with administrative controls that they normal would not have access to. For example, with this backdoor an attacker could modify or delete information with malicious intent. An example of code responsible for this backdoor follows and occurs numerous times throughout the codebase:

```
    ClimbInfoPage.prototype.checkIfUser = function () {
        var _this = this;
        this.localUser.getUser().then(function (localUser) {
            if (localUser && _this.climb) {
                _this.user = localUser;
                _this.userIsSetter = (localUser.username === _this.climb.setBy) ? true : false;
                _this.userIsAdmin = (localUser.email.toLowerCase() === _this.wallAdmin.toLowerCase() || localUser.email === 'mark@markdgold.com' || _this.userIsSetter) ? true :
                if (localUser['logbook' + _this.climb.wall] && localUser['logbook' + _this.climb.wall].length > 0) {
                    localUser['logbook' + _this.climb.wall].forEach(function (climb) {
                        if (climb.id === _this.climb.id)
                            _this.userHasLogged = true;
                    });
                }
            }
        });
}
```

As an example attack scenario, an attacker can simply change this address manually and abuse the extra functionality granted within the app.

Insufficient Security Controls

It appears that all administrative functionality for the application is enforced client-side, which could allow a malicious actor to manually forge API requests in order to access information they would not normally have access to. For example, by manually forging requests, our researcher was able to add, modify, and delete walls (private or not), problems, images, users, etc. For example, we were able to obtain a full list of walls and the associated password hashes for private walls by manually sending these requests within a rogue app:

```
]],[31,[{
  "documentChange": {
    "document": {
      "name": "projects/whatsyourspraywall/databases/(default)/documents/walls/<censored>",
      "fields": {
        "setDate": {
          "integerValue": "1597465843256"
        },
        "gym": {
          "stringValue": "<censored>"
        },
        "skin": {
          "mapValue": {
            "fields": {
              "grades": {
                "booleanValue": true
              },
              "aboutText": {
                "stringValue": ""
              },
              "logo": {
                "stringValue": ""
              },
              "aboutImg": {
                "stringValue": "https://i.imgur.com/<censored>.jpg"
              }
            }
          }
        },
        "name": {
          "stringValue": "<censored>"
        },
        "location": {
          "stringValue": "NY"
        },
        "website": {
          "stringValue": ""
        },
        "password": {
          "stringValue": "$2a$08$/<censored>/9kABLq9D5e0IyVCbhK"
        },
        "admin": {
          "stringValue": "<censored>@gmail.com"
        },
        "id": {
  },
  "targetIds": [
    6
  ]
}
```

Disclosure Timeline

September 4, 2020 - Tenable discloses to vendor.
September 14, 2020 - Tenable requests acknowledgement.
September 21, 2020 - Tenable requests acknowledgement.
October 19, 2020 - Tenable requests status update or acknowledgement.

All information within TRA advisories is provided "as is", without warranty of any kind, including the implied warranties of merchantability and fitness for a particular purpose, and with no guarantee of completeness, accuracy, or timeliness. Individuals and organizations are responsible for assessing the impact of any actual or potential security vulnerability.

Tenable takes product security very seriously. If you believe you have found a vulnerability in one of our products, we ask that you please work with us to quickly resolve it in order to protect customers. Tenable believes in responding quickly to such reports, maintaining communication with researchers, and providing a solution in short order.

For more details on submitting vulnerability information, please see our Vulnerability Reporting Guidelines page.

If you have questions or corrections about this advisory, please email advisories@tenable.com

Risk Information

CVE ID: CVE-2020-5799
CVE-2020-5800
Tenable Advisory ID: TRA-2020-65
CVSSv3 Base / Temporal Score: 7.3 / 7.1
CVSSv3 Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
Affected Products: Eat Spray Love for Android 2.0.20
Eat Spray Love for iOS 2.0.20
Risk Factor: High

Advisory Timeline

December 3, 2020 - Initial Release

FEATURED PRODUCTS

Tenable One Exposure Management Platform

Tenable.cs Cloud Security

Tenable.io Vulnerability Management

Tenable.io Web App Scanning

Tenable.asm External Attack Surface

Tenable.ad Active Directory

Tenable.ot Operational Technology

Tenable.sc Security Center

Tenable Lumin

Nessus

→ View all Products

FEATURED SOLUTIONS

Application Security

Building Management Systems

Cloud Security Posture Management

Compliance

Exposure Management

Finance

Healthcare

IT/OT

Ransomware

State / Local / Education

US Federal

Vulnerability Management

Zero Trust