

**CleverVoice**

Web Application

February 2018

**Version:** 1.0

**Report By:** Andrew Mabbitt

**Email** [Andrew@fidusinfosec.com](mailto:Andrew@fidusinfosec.com)

**Telephone:** +447534586788

Table of Content

[1 Document Information 3](#_Toc505682403)

[2 Management Summary 4](#_Toc505682404)

[3 Risk Ratings 5](#_Toc505682405)

[4 Summary of Findings 6](#_Toc505682406)

[4.1 Key Findings 6](#_Toc505682407)

[4.2 Recommended Next Steps 7](#_Toc505682408)

[5 Vulnerability Findings and Technical Details 8](#_Toc505682409)

[5.1 External Infrastructure 9](#_Toc505682410)

[5.1.1 Broken Access Controls 9](#_Toc505682411)

[5.1.2 Cross-Site Request Forgery (CSRF) 10](#_Toc505682412)

[5.1.3 Security Related HTTP Headers 12](#_Toc505682413)

[5.1.4 TLS/SSL Issues 14](#_Toc505682414)

[5.1.5 Concurrent Logins Permitted 16](#_Toc505682415)

[5.1.6 Verbose Server Headers 18](#_Toc505682416)

[5.1.7 Verbose Error Messages 19](#_Toc505682417)

[5.1.8 Password Autocomplete Enabled 20](#_Toc505682418)

[5.1.9 Cookies Not Set with HttpOnly or Secure Flag 22](#_Toc505682419)

[5.1.10 Username Enumeration 24](#_Toc505682420)

[6 Tool List 26](#_Toc505682421)

[7 Methodologies 27](#_Toc505682422)

# Document Information

**Confidentiality and Copyright**

All information contained in this document is provided in confidence for the sole purpose of adjudication of the document, and shall not be published or disclosed wholly or in part to any other party without Fidus Information Security’s (Fidus) prior permission in writing, and shall be held in safe custody. These obligations shall not apply to information that is published or becomes known legitimately from some source other than Fidus. All transactions are subject to the appropriate Fidus Standard Terms and Conditions.

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Issue Date | Issued By | Change Description |
| 0.1 | 05/02/2018 | Andrew Mabbitt | Draft for internal review only |
| 0.2 | 06/02/2018 | Kurtis Baron | Revised QA |
| 1.0 | 06/02/2018 | Andrew Mabbitt | Released to Client |

**Testing Team**

|  |  |
| --- | --- |
| Name | Qualifications |
| Andrew Mabbitt | CRT, CCT, OSCP |
| Kurtis Baron | CSTM, SST, CISSP |

**Accreditations and Certifications**



# Management Summary

This report details the findings of the CleverVoice web application assessment carried out on behalf of CleverVoice on the 3rd February 2018. The objective of the assessment was to identity any vulnerabilities or potential issues which could impact the Confidentiality, Integrity or Availability of the CleverVoice application and any data contained within.

Overall, the application was found to suffer from a handful of serious issues which could impact both the confidentiality and integrity of the environment. Whilst some protections were identified, such as; validation of user-input and sanitisation of data, other issues were identified, such as; broken access controls, information leakage and a lack of security best practices.

The application was found to be overly descriptive in areas such as; error messages and forgotten password mechanisms. Whilst the issues may seem like they do not pose much risk, they provide an attacker a lot of information which they could use in further attacks against the application and/or the environment.

Another issue identified related to the configuration of encryption ciphers used to secure communication between the service and its clients. It was identified that support was offered for outdated methods that may prove vulnerable to attack. A suitably-placed attacker could potentially leverage such weaknesses to intercept, monitor and possibly alter legitimate traffic.

The findings contained in this report should be reviewed by the relevant service stakeholders, and remediated in accordance with CleverVoice’s risk appetite.

# Risk Ratings

The table below has been generated to provide an insight into the colours, risk rating and scoring system used throughout this report to help provide a concise and transparent overview.

It should be noted that issues have been rated based on the evidence discovered by the tester and whilst there may be controls in place in the backend of the systems to prevent specific attacks occurring, it may have been known to the tester throughout the assessments.

|  |  |  |  |
| --- | --- | --- | --- |
| Colour | Risk Rating | CVSSv2 Score | Explanation |
| Purple | **Critical** | **9.0-10.0** | **This requires resolution as quickly as possible.** |
| Red | **High** | **7.0-8.9** | **This requires resolution in the near future.** |
| Orange | **Medium** | **4.0-6.9** | **This requires resolution in the medium term.** |
| Blue | **Low** | **1.0-3.9** | **This requires resolution as part as routine maintenance.** |
| Green | **Informational** | **0-0.9** | **This requires resolution to be in line with best practices.** |

# Summary of Findings

The following table summarises the issues identified throughout testing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Critical | High | Medium | Low | Total |
| Web Application | 1 | 1 | 2 | 6 | 10 |
| Total | 1 | 1 | 2 | 6 | 10 |

All security issues are presented alongside recommendations for mitigating the risks posed. These can be found alongside the related issues in section [[5]](#_Vulnerability_Findings_and) of this report.

## Key Findings

* It was possible to access other user’s clients sensitive information.
* Forms were vulnerable to Cross-Site Request Forgery (CSRF) attacks.
* Security headers which would improve the application’s security posture were found to be missing.

## Recommended Next Steps

The goals provided are set out to be suggestions and should be reviewed and fixed according to the risk posed to your business model.

**Short Term Goals**

* Ensure users can only export their own call/message history.
* Add one-time nonces to forms to mitigate CSRF attacks.
* Enable missing security headers on the application server.

**Medium Term Goals**

* Expire user sessions upon a new login; therefore mitigating concurrent logins.
* Remove all verbose error messages and headers.
* Change response messages on the password reset form to avoid username enumeration.
* Remediate the remaining findings in accordance with CleverVoice’s risk appetite.

# Vulnerability Findings and Technical Details

The following section details vulnerabilities found throughout the testing phase alongside the technical details associated. The following information is included, where applicable:

* **Risk Rating**
* **Description**
* **Recommendation**
* **Affected Hosts**
* **References**

Where possible, enough information to replicate the finding will be included within the vulnerability write up. Additional information, where required, will be added into the Appendices of this report.

## External Infrastructure

|  |  |
| --- | --- |
| Broken Access Controls |  |
| Risk Rating: | **Critical** |

Issue Description:

Due to broken access controls, it was possible to download an exported version of ALL users call/message history including; first names, surnames, company and mobile numbers. The only requirement to exploit this is a registered user account which can be freely obtained.

It would be possible to automate the process and iterate through all calls/messages made on the application.

For example, it is possible to create a new user account and access the following URL:

<https://my.clevervoice.com/message/excel/4020>

This would provide the user an Excel download containing sensitive data belonging to another user of the application.

Due to the sensitive nature of data contained within the Excel file, a screenshot has not been provided for reporting purposes.

Recommendation:

Ensure users of the application are only allowed to generate exports of their own data.

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

OWASP Guidance

<https://www.owasp.org/index.php/Broken_Access_Control>

|  |  |
| --- | --- |
| Cross-Site Request Forgery (CSRF) |  |
| Risk Rating: | **High** |

Issue Description:

The affected applications were found to be vulnerable to cross-site request forgery (CSRF) attacks. This could allow an attacker to manipulate a user into unwittingly performing actions within the application such as changing account information.

For the CSRF attack to be successful, the victim must click the malicious link or visit a malicious webpage whilst logged into the vulnerable application. This would cause the victim’s browser to issue a GET or POST request to the application. The result of this is a genuine request to the application executed on the user’s behalf.

Proof of concept code for the CSRF attack can be seen below:

<html>

<body>

<form action="https://my.clevervoice.com/user/create" method="POST">

<input type="hidden" name="user&#91;firstname&#93;" value="CSRF" />

<input type="hidden" name="user&#91;surname&#93;" value="Example" />

<input type="hidden" name="user&#91;email&#93;" value="CSRF&#64;fidusinfosec&#46;com" />

<input type="hidden" name="ok" value="OK" />

<input type="submit" value="Submit request" />

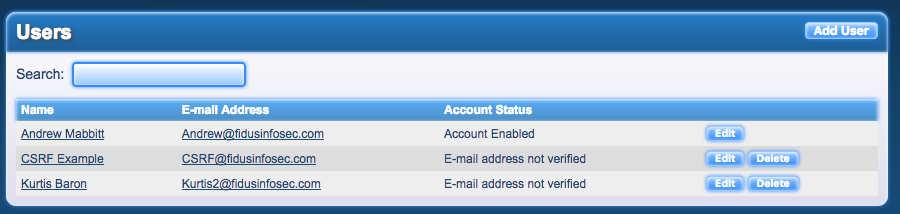
</form>

</body>

</html>

Using the above proof of concept code, it was possible to have a new user added to the user’s account who clicked the submit button on the form which was hosted on a separate domain.

This issue has been rated high due to the fact it allows complete account takeover with one click from an authenticated user.



**Figure 1: Cross-Site Request Forgery User Added**

Recommendation:

The primary defence against CSRF attacks is to create a one-time token that are replaced on the page before being validated server-side to ensure that the requests can only be executed one.

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

OWASP Guidance

<https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)>

|  |  |
| --- | --- |
| Security Related HTTP Headers |  |
| Risk Rating: | **Medium** |

Issue Description:

The affected hosts did not make use of some HTTP headers that could be used to enhance the security posture of the application. It should be noted that whilst some pages did make use of security headers, some pages, such as /account/profile, did not.

The **Strict-Transport-Security** HTTP header is used to instruct the browser to only access a web application over a secure connection and for how long to remember this restriction. Thereby forcing continued use of a secure connection.

This header cannot completely defend against man-in-the-middle attacks, but providing that the user has previously visited the site without outside interference, it can be useful in defending against an attack in which an attacker establishes an encrypted connection to the application and presents an unencrypted fraudulent service to the user, as the user’s browser will know not to use the unencrypted service.

The **X-Content-Type-Options** HTTP header can be used to prevent web browsers from using content sniffing to discover a file’s MIME type. This header, when set, can help protect against cross-site scripting attacks.

The **Content-Security-Policy** header is a mechanism for controlling which external applications/sites can host resources used by an application and control how these resources may behave. The Content-Security-Policy (CSP) can provide defence in depth against attacks from content injection attacks.

The **cache-control** and **pragma** HTTP headers have not been set by the web servers. This enables the user’s browser and any proxies to cache the content of secure pages, which could enable sensitive information disclosure. Content that should only be accessible to an authenticated user can therefore be recovered after the session has terminated through the local cache or by pressing the ’back’ button in the browser.

An example of a basic GET request to the /account/profile page can be seen below.

HTTP/1.1 200 OK

Date: Sun, 04 Feb 2018 13:13:38 GMT

Server: Apache/2.2.22 (Ubuntu)

X-Powered-By: PHP/5.3.10-1ubuntu3.26

Vary: Accept-Encoding

Content-Length: 6367

Connection: close

Content-Type: text/html; charset=utf-8

Recommendation:

Consideration should be given to implementing the below recommended HTTP headers.

Strict-Transport-Security: max-age=31536000; includeSubDomains

X-Content-Type-Options: nosniff

Cache-Control: no-cache, no-store, must-revalidate, private

Pragma: no-cache

X-Frame-Options: DENY

Content-Security-Policy: policy

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

OWASP Secure Headers Project

<https://www.owasp.org/index.php/OWASP_Secure_Headers_Project>

Guidelines for Setting Security Headers

<https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers>

|  |  |
| --- | --- |
| TLS/SSL Issues |  |
| Risk Rating: | **Medium** |

Issue Description:

Numerous TLS/SSL issues were identified, these are identified below:

**Export Grade Ciphers Supported (FREAK / Logjam)**

The remote service utilises a Diffie-Hellman Modulus 1024 bits long or less. Due to advances in hardware able to factorise Diffie-Hellman moduli, this is no longer considered secure for the exchange of cryptographic keys.

**SSL Version 3 Protocol Detection**

The affected hosts were found to accept connections encrypted using SSL 3.0. Both of these versions of SSL are affected by several cryptographic flaws. An attacker could leverage these flaws to perform Man-in-the-Middle (MITM) attacks and steal sensitive data.

**SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)**

The affected hosts are vulnerable to a Man-in-the-Middle (MITM) information disclosure vulnerability known as POODLE. The vulnerability lies within the way SSL 3.0 handles padding bytes when decrypting encrypted messages. The vulnerability exists within SSLv3 and not within any particular SSL implementation.

**SSL RC4 Cipher Suites Supported**

The affected hosts were found to support RC4 cipher suites. The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness. If plaintext is repeatedly encrypted (such as a static cookie) and an attacker is able to steal a large enough sample, they may be able to derive the plaintext and break the encryption.

**RC4** (CVE-2013-2566, CVE-2015-2808) VULNERABLE (NOT ok): **ECDHE-RSA-RC4-SHA RC4-SHA**

**Weak / Medium SSL Ciphers Supported**

The remote host supports the use of SSL ciphers that offer weak or medium strength encryption, which is currently regarded as ciphers using keys shorter than or equal to 112 bits.

Although Triple DES (3DES) uses a 168 bit key, it has been shown to provide an effective key strength of 112 bits, at best.

xc012 ECDHE-RSA-DES-CBC3-SHA ECDH **3DES** 168 TLS\_ECDHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

x16 EDH-RSA-DES-CBC3-SHA DH **3DES** 168 TLS\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

x0a DES-CBC3-SHA RSA **3DES** 168 TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

Recommendation:

**Export Grade Ciphers Supported (FREAK / Logjam)**

It is recommended a Diffie-Hellman modulus be generated that is no longer than 1024 bits. This can be achieved with the OpenSSL dhparam tool.

**SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)**

Disable SSLv3 where possible. If SSLv3 cannot be disabled, ensure the TLS Fallback SCSV mechanism has been enabled to mitigate the risk.

**SSL Version 3 Detection**

Disable SSL v3 in favour for more secure alternatives, such as TLS 1.1 and TLS 1.2.

**SSL RC4 Cipher Suites Supported**

Where possible, disable the RC4 cipher suite in favour of more cryptographically secure algorithms.

**Weak / Medium SSL Ciphers Supported**

Disable all weak ciphers in favour of cryptographically secure alternatives.

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

**Export Grade Ciphers Supported (FREAK / Logjam)**

<https://www.ssl.com/article/logjam-ssl-tls-vulnerability-time-to-freak-out-again/>

**SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)**

<https://blog.qualys.com/ssllabs/2014/10/15/ssl-3-is-dead-killed-by-the-poodle-attack>

**SSL Version 3 Detection**

<http://disablessl3.com/>

**SSL RC4 Cipher Suites Supported**

<http://www.isg.rhul.ac.uk/tls/>

<https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/>

**Weak / Medium SSL Ciphers Supported**

<https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet#Server_Protocol_and_Cipher_Configuration>

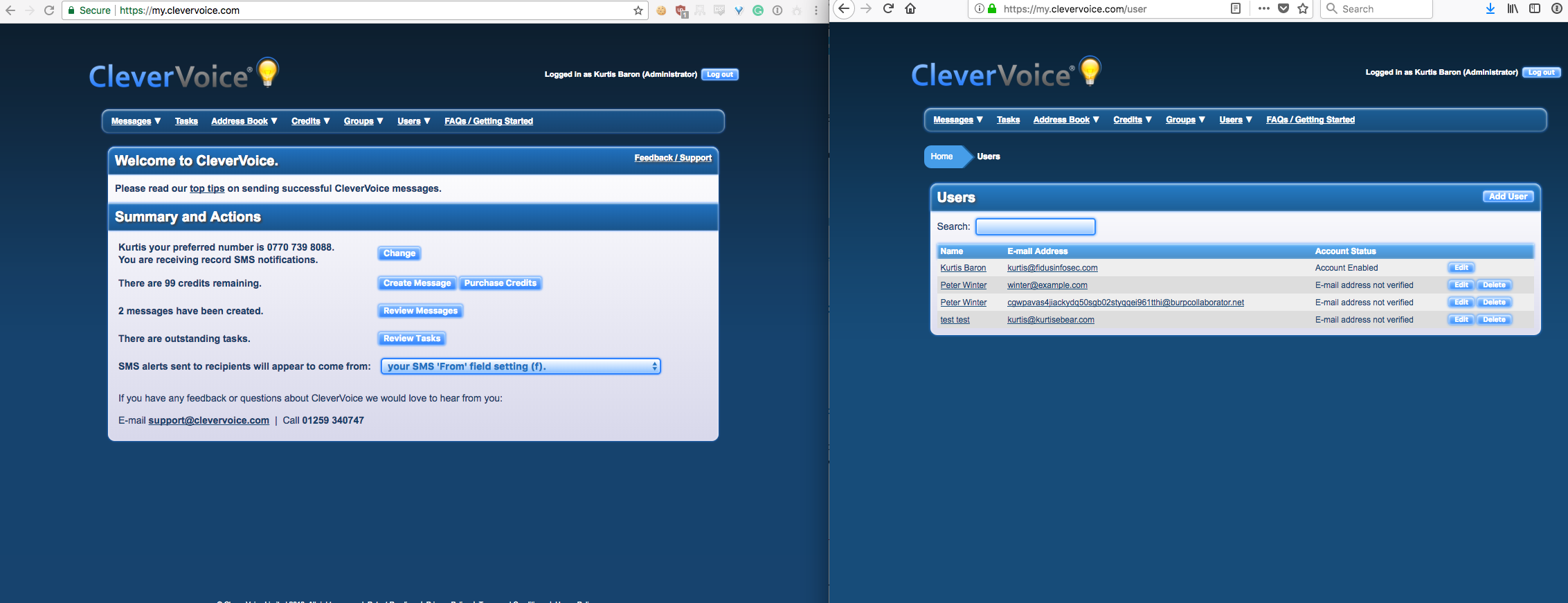
|  |  |
| --- | --- |
| Concurrent Logins Permitted |  |
| Risk Rating: | **Low** |

Issue Description:

The CleverVoice application permitted multiple concurrent logins using the same user account and allowed for simultaneous sessions to be generated. This may also be possible from different IP addresses.

Failure to prevent concurrent logins hinders accountability in the event of any malicious acts, or accidental actions and would make it harder for a user to notice should their account be compromised.

Furthermore, concurrent logins may cause errors if similar data is entered and/or uploaded almost simultaneously.

**Figure 2: Concurrent Logins in 2 Separate Browsers**

Recommendation:

User accounts should only be permitted to have one active session at a time. If the user authenticates in a new request, the previous session(s) should be invalidated and terminated immediately with an appropriate message displayed to inform the user on both ends.

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

OWASP Guidance

<https://www.owasp.org/index.php/Session_Management_Cheat_Sheet#Simultaneous_Session_Logons>

|  |  |
| --- | --- |
| Verbose Server Headers |  |
| Risk Rating: | **Low** |

Issue Description:

The external web services disclosed software versions in HTTP response headers. Although not a direct risk to the servers or related infrastructure, this allows an attacker to identify potential vulnerabilities, leading to targeted attacks against those types of servers or the particular version installed.

It is common for web servers to disclose their type and version information in their headers. This information is a form of banner, which can be retrieved via HTTP requests to the web server such as HEAD and GET.

A basic request to the homepage gives a HTTP response that discloses the server is running the following:

HTTP/1.1 200 OK

Date: Sun, 04 Feb 2018 13:13:38 GMT

**Server: Apache/2.2.22 (Ubuntu)**

**X-Powered-By: PHP/5.3.10-1ubuntu3.26**

Vary: Accept-Encoding

Content-Length: 6367

Connection: close

Content-Type: text/html; charset=utf-8

Recommendation:

If the verbose server headers do not serve a business purpose then they should be removed, obfuscated or replaced with a less specific text, such as: ”Web Server”. Version information should also be removed.

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

OWASP: Web Server Fingerprinting

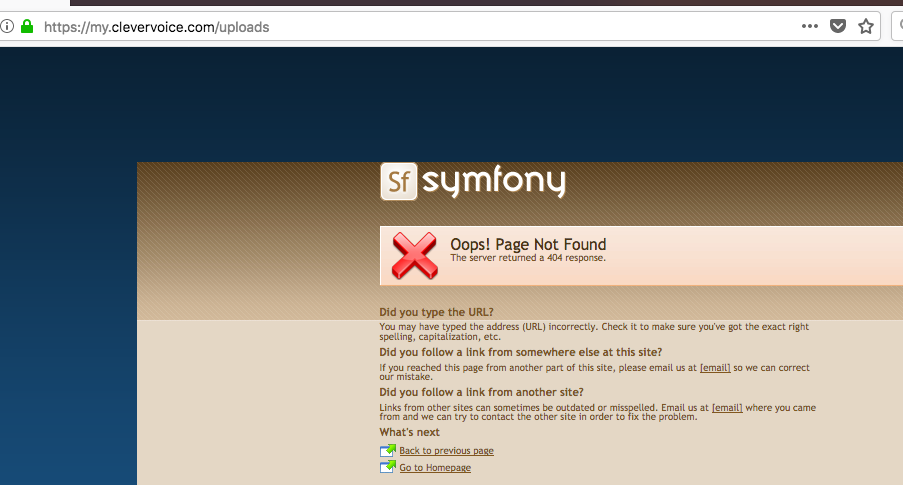
<https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002)>

|  |  |
| --- | --- |
| Verbose Error Messages |  |
| Risk Rating: | **Low** |

Issue Description:

The CleverVoice application was configured to display detailed error messages when an application exception was generated.

This information can be extremely beneficial for an attacker and could allow for the launch of further, specific, attacks. In this instance, it is possible for an attacker to ascertain that Symfony is in-use within the application.



**Figure 3: Verbose Error Messages**

Recommendation:

Application errors should be handled gracefully and provide as little information as possible to end-users.

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

CWE-209: Information exposure through an error message

<https://cwe.mitre.org/data/definitions/209.html>

|  |  |
| --- | --- |
| Password Autocomplete Enabled |  |
| Risk Rating: | **Low** |

Issue Description:

The password field on the CleverVoice application was identified as accepting the autocomplete function. The autocomplete function, implemented by popular browsers, allows end-users to store field values so they can be automatically populated by the browser next time.

Although this can be seen as a convenience feature, it creates a security risk and leaves a user’s username and password at risk as they are stored locally.



**Figure 4: Autocomplete Enabled**

Recommendation:

Form fields containing sensitive data, such as passwords, should have the autocomplete option disabled on the form. An example of this can be found below:

<form autocomplete=”off”>

…

<input type=”text” name=”username” **autocomplete=”off”**>

<input type=”text” name=”password” **autocomplete=”off”**>

…

</form>

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

How to Turn Off Form Autocompletion

<https://developer.mozilla.org/en-US/docs/Web/Security/Securing_your_site/Turning_off_form_autocompletion>

|  |  |
| --- | --- |
| Cookies Not Set with HttpOnly or Secure Flag |  |
| Risk Rating: | **Low** |

Issue Description:

Session cookies are used to track a user’s application session and are typically used to check the permissions that a specific user session has within the application with regard to its available resources and functionality. By acquiring another user’s session cookie value, an attacker may be able to connect to the application as that user.

If the acquired session cookie value relates to an authenticated session, then the attacker would have access to the application resources typically available to the targeted user.

The HttpOnly flag is available to be set on cookie values to prevent the cookie from being accessed by non-HTTP resources, such as JavaScript. By not setting the HttpOnly flag on a number of Session cookies, sites are potentially vulnerable to session hijacking attacks through the exploitation of other application issues, such as Cross-Site Scripting.

If a browser that supports HttpOnly detects a cookie containing the HttpOnly flag, and client side script code attempts to read the cookie, the browser returns an empty string as the result. This assists in protecting the logged-in user’s session identifier and helps to prevent session hijacking attacks.

The absence of the Secure flag on sensitive cookies means they are at risk of being transmitted over unencrypted HTTP connections.

Setting the Secure flag on HTTPS issued cookies, such as the SessionID cookie, ensures the cookie will not be returned over HTTP connections.

An example can be seen below:

Set-Cookie: clevervoice=l6069f2qo9ik6jmjeoth6slbc1; path=/

Recommendation:

All sensitive cookies should be set with both the HttpOnly and Secure flag.

Set-Cookie: clevervoice=l6069f2qo9ik6jmjeoth6slbc1; path=/; **HttpOnly; Secure**

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

OWASP Guidance

<https://www.owasp.org/index.php/HttpOnly>

CWE-614: Sensitive Cookie in HTTPS Session Without ‘Secure’ Attribute

<https://cwe.mitre.org/data/definitions/614.html>

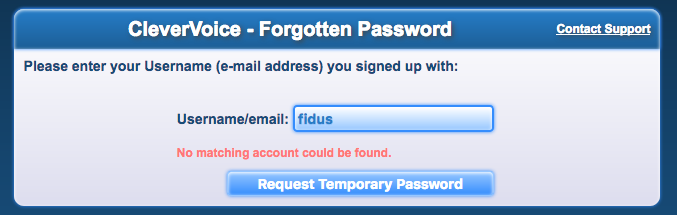
|  |  |
| --- | --- |
| Username Enumeration |  |
| Risk Rating: | **Low** |

Issue Description:

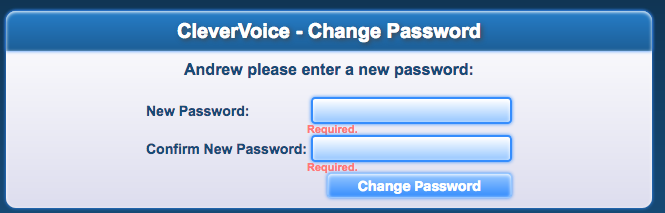
It was identified that the CleverVoice application was vulnerable to username enumeration through the forgotten password mechanism.

When requesting a password reset for a valid username, a different message was displayed to that of a non-existent username. Due to this, it would be possible for an attacker to automate the process of enumerating usernames.

Examples of this can be seen below:



**Figure 5: Invalid User Message**

****

**Figure 6: Valid User Message**

Recommendation:

A generic message should be displayed to all users requesting a password reset, regardless of validity of the username. An example of this would be “If the user is found on the system, a password reset will be emailed to you”

Affected Hosts:

|  |  |
| --- | --- |
| Hostname | Port |
| my.clevervoice.com | 443/TCP |

References

About Username Enumeration

<https://blog.rapid7.com/2017/06/15/about-user-enumeration/>

# Tool List

|  |  |
| --- | --- |
| Tool Name & Version | Description |
| Nessus 7.60  (Professional License) | Vulnerability scanner  https://www.tenable.com/products/nessus/nessus-professional |
| Nmap 7.60 | Open source port scanning tool  https://nmap.org/download.html |
| Testssl 2.9.5 | Open source SSL/TLS enumeration tool  https://testssl.sh/ |
| Burp Suite 1.7.31  (Professional License) | Web Application Proxy and Security Tool  https://portswigger.net/burp |
| Dirbuster 0.12 | Directory Brute-Force Tool  https://www.owasp.org/index.php/Category:OWASP\_DirBuster\_Project |
| Nikto 2.1.5 | Web Server Vulnerability Scanner  https://github.com/sullo/nikto |

# Methodologies

Web Application

Fundamentals

Web application assessments can be performed either remotely or on site, depending on the accessibility of the application. The purpose the assessment is to uncover all vulnerabilities and weaknesses which could be leveraged by a malicious actor in order to gain access to sensitive data held by the application.

Throughout the assessment, consultants utilise proven methodologies to ensure the application gets a thorough test. This is a mixture of non-invasive testing and more hands testing techniques when required.

A web application assessment can be divided into three stages:

* Discovery
* Assessment
* Exploitation

Test Areas

Fidus utilise a wide range of tools to scan and map out the web application(s). A mixture of the latest automated and manual tools is incorporated into testing by our consultants. Some of these include:

* Burp Suite Spider
* Nikto
* Dirbuster

Once the discovery phase has ended, Fidus consultants interpret the results and use them to identify possible attack vectors and perform manual attack simulations. Manual assessments focus on the OWASP top 10:

* A1: Injection
* A2: Broken Authentication and Session Management
* A3: Cross-Site Scripting (XSS)
* A4: Insecure Direct Object References
* A5: Security Misconfiguration
* A6: Sensitive Data Exposed
* A7: Missing Function Level Access Control
* A8: Cross-Site Request Forgery (CSRF)
* A9: Utilising Known Vulnerable Components
* A10: Unvalidated Redirects and Forwards

If a successful venue of attack is identified, Fidus will work with you to conduct safe exploitation (where possible) and verification of the issue whilst ensuring there are no disruptions to the daily running of your organisation. All exploitation is conducted under the agreed rules of the engagement.

Should exploitation successfully permit, Fidus will aim to escalate to the highest of privileges and, with your agreement, continue to leverage this access to penetrate as deep as possible in your network to help portray a realistic attack scenario.