SMS Privacy Security Audit (September 2017)

This report contains the results of a web application security audit of SMS Privacy, a privacy tool which allows users to purchase and use mobile phone numbers using Bitcoin. Users who purchase such mobile numbers pay a fee for the duration of use, and are able to send and receive SMS messages for the period of their ownership of a number. In certain circumstances, users may also be able to receive voicemails on the numbers they have purchase.

A test of SMS Privacy was requested after the discovery of an accidental information leak in August 2017.

Test scope

The test of SMS Privacy covered the following:

- Checking for the presence of common web application vulnerabilities, such as the presence of SQL injection, cross-site scripting and cross-site request forgery.
- Attempting to force the application to reveal more information regarding user accounts on the system than was intended by the application's author.
- Checking for correct validation of user-supplied input.
- Examining the interaction of SMS with the web application to ensure that SMS could not be used to force the application to behave unexpectedly.
- Checking for the correct implementation of access controls across the application.
- Checking that the server configuration for the SMS Privacy website implements current best-practices for application security.
- Ensuring correct session management logic has been implemented in the application

Test limitations

The application test did not include:

- Testing of other applications on the server running SMS Privacy.
- Analysis of any other software running on the server and any vulnerabilities which may apply to it.
- Analysis of the protocols used by the server (such as TLS or SSH) and the implications of their configuration on the application's security.
- Analysis of the SMS Privacy codebase.
- Testing of the application behaviour over Tor and any security implications which might arise from using the application over Tor.

Disclaimer

Security testing in this manner can only confirm the presence of vulnerabilities, and not their absence. It is the responsibility of the application's owner to maintain server software and the application logic such that known vulnerabilities are defended against. The author of this audit report bears no liability for the consequences of any vulnerabilities discovered or exploited in the application, whether these consequences affect the application's owner or its users.

Summary

The general standard of application security on the SMS Privacy website was reasonably high. Common vulnerabilities such as SQL injection and cross-site scripting were correctly defended against. User supplied input was in most cases correctly validated, and access controls prevented the access of information across user accounts.

Server configuration for the application could be improved, however. A number of web security mechanisms which would increase the application's defence-in-depth are not currently in use. Session management could also be improved. The application can also be made to behave in unexpected ways in certain circumstances. None of the issues discovered in the application are critical, but should a critical vulnerability exist, the absence of some of the security protections available will greatly increase the consequences of the application's compromise. Given the application's focus on privacy, it is strongly recommended that the application maintainer implement as many of the suggested improvements below as is feasible.

Retest

In October 2017, a retest of SMS Privacy was undertaken after feedback on fixed vulnerabilities; one incorrectly reported issue was also updated to rectify errors. Where issues have been resolved, a 'Retest' section has been added, noting the fix made and the results of retesting. If a reported issue has no 'Retest' section, it has not been fixed.

Vulnerabilities

Session cookies do not use the `Secure` flag

The `Secure` cookie flag indicates to browsers that the cookie should only be transmitted over an HTTPS connection. SMS Privacy uses HTTPS by default, and it is not possible (under intended circumstances) to access the application over HTTP. Despite this, should the user be tricked into making an unencrypted HTTP request to the SMS Privacy server, this initial request would transmit the session cookie in plaintext. A man-in-the-middle attacker would be able to read this cookie value.

Although the requirement of a man-in-the-middle position on the network limits the practical exploitability of this vulnerability, the use of the `Secure` flag is recommended given the privacy-focused nature of the SMS Privacy application.

Retest: The `Secure` flag is now used on session cookies.

`X-Frame-Options` header not set

The `X-Frame-Options` header controls the embedding of page resources via iframes. The use of the header prevents against 'clickjacking' style attacks. SMS Privacy does not make use of the header. It is recommended that either this header, or the equivalent `Content-Security-Policy` header (a newer mechanism not supported by legacy browsers) is implemented in the application.

Retest: The `X-Frame-Options` header has been set to `DENY`, preventing attempts to load SMS Privacy pages from within frames.

HTTP Strict Transport Security (HSTS) not in use

The HSTS header instructs a browser to serve content exclusively over HTTPS, even if a request was made for a page resource over HTTP. Browsers also prevent users from bypassing certificate errors when HSTS is enabled. It is strongly recommended that the HSTS header is used on the SMS Privacy website, as it provides strong protection against man-in-the-middle downgrade attacks. Also recommended is the use of the `preload` directive, which will ensure that even the first request to the website is conducted over HSTS in browsers which support the use of the header.

Retest: The HSTS header has been correctly implemented, making use of both the `preload` and `includeSubDomains` directives. The site has been submitted to the HSTS preload list, and should shortly have HTTPS enabled from the first request.

HTTP Public Key Pinning (HPKP)

SMS Privacy does not use HTTP Public Key Pinning. HPKP offers protection against the impersonation of websites, and is recommended by Mozilla for use on 'maximum risk websites'. Given SMS Privacy's focus on the privacy of its users, the use of HPKP might be another security mechanism worth considering.

Long session lengths

Once logged in, a session remains valid for a period greater than several hours. While this is not a critical vulnerability, OWASP recommends that session durations be kept to a minimum, to reduce the attack surface on a user's session.

Sessions are not terminated correctly

On clicking the 'Logout' button, the user's cookies are cleared and the user is directed back to the login page. From the user's perspective, this ends their session. The session has not been terminated on the server, however, which means that it is possible to resume a session by restoring the value of the session cookie. The best practice would be to invalidate a user's session after a logout request is received. This reduces the attack surface available to attackers looking to compromise users on the application.

Retest: After the user logs out, all existing sessions for that user are invalidated; after the user changes password, all other sessions for that user are invalidated.

Password policy weak

The password policy does not appear to be particularly strong - the application accepted a string consisting of three whitespace characters as a valid user password.

No protection against brute-force login attempts

The application does not appear to make any effort to prevent against brute-force login attempts. The application's design makes

login lockouts difficult to implement (users are not expected to associate their accounts with email addresses). Temporary IP bans or similar access control restrictions could still be used to reduce the feasibility of such attacks against the application's users.

Retest: If there has been an invalid password attempt in the last 10 seconds, login attempts are delayed by 250ms as a rate limit.

Administrative access controls could be improved

Access control to the administrative section of the application is implemented using the same mechanisms as regular users. Administrators are able to access a large amount of information regarding users via the administrative web interface. It is recommended that other defence-in-depth measures are implemented to bolster the security of the administrative area. For example, these could include IP-based restrictions or the use of client certificates.

Retest: Administrative access is now restricted by IP address.

Voicemails are accessible across accounts

Voicemails stored by the application are available at URLs similar to `/voicemail/XXX-b633eee3-f75c-4x91-e5a9-292ccc7fc224.mp3`. The filenames are sufficiently long and the manner in which they are generated is difficult to reverse-engineer, which reduces the likelihood of voicemails being enumerated. Voicemails are available across user accounts with no access control, however. Should an attacker have knowledge of the system used to generate the filenames used in the application, or be able to acquire a list of voicemail recordings on the application, accessing the recordings would then be trivial. It is recommended that access control is implemented such that users are only served voicemails intended for them.

Retest: Voicemails are now only accessible to their owners.

Implementation notes

Username whitespace bug

Whitespace appears to be accepted in usernames, but it appears that the logic for such usernames isn't correctly implemented. During testing, an attempt was made to create an account with the username ' '. This was accepted, however, the application now claims that all usernames which consist of whitespace have now been taken. It is also possible to log in as ' ' and ' ', for example, using the password for the ' ' account.

Application errors can be forced

The page which serves virtual SMS numbers is found at `https://smsprivacy.org/buy-number/virtual/XX`, where XX is the country code of the country in which the number is required. If this is a country supported by SMS Privacy, the user is returned a page containing a list of numbers which they can purchase. If the country isn't supported, however, the application errors after a lengthy waiting period with a `502 Bad Gateway` message.

Retest: This has now been fixed.