Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Summary-Report covering Amnesty International WordPress Themes & Plugins 09.-10.2023

Cure53, Dr.-Ing. M. Heiderich, J. Larsson, MSc. J. Moritz, M. Elrod, M. Wege, MSc. R. Peraglie, MSc. S. Moritz

Index

Introduction

“Amnesty International is a global movement of more than 10 million people who take injustice personally. We are campaigning for a world where human rights are enjoyed by all.”

From https://www.amnesty.org/en/who-we-are/

This report, assigned the unique reference ID AI-02, presents the outcomes of a Cure53 penetration test and source code audit against the Amnesty International main WordPress theme, various plugins, and corresponding deployment.

The assignment was completed during a time frame spanning CW36 to CW41 September-October 2023 following initial procurement discussions with Amnesty International in August 2023. A total of sixteen days were allocated by the client to reach satisfactory examination depth; Cure53 offered its services pro bono.

Three individual and manageable Work Packages (WPs) were created in advance, as defined by the following headers:

- **WP1**: White-box pen.-tests against Amnesty International WordPress templates
- **WP2**: White-box pen.-tests against Amnesty International WordPress plugins
- **WP3**: White-box pen.-tests against Amnesty International WordPress deployment

As observable from the WPs above, the pentest methodology of choice for this exercise was white-box. The audit team’s actions were assisted through the provision of abundant assets, such as sources, URLs, descriptive documentation, and other data points required for access or clarification purposes. Seven senior testers were selected from the Cure53 team to complete the engagement’s multi-stage requirements, from the initial preparatory measures (which were conducted in CW35 2023) to the test itself and final conclusory operations.

All active personnel from Amnesty International and Cure53 were invited to a private communications channel, wherein cross-team communications were facilitated. The collaboration process was altogether effortless, considering that no blockers were encountered throughout the entirety and the auditors were suitably familiarized with the scope.

This format hosted live reporting in addition, which was enacted at the behest of the maintainers in order to raise awareness of a selection of high-profile findings as soon as they were discovered.

The designated time frame for the evaluations was ample and attributed to a highly fruitful pentest. Following widespread coverage, the Cure53 consultants observed and documented a total of five security impacting behaviors blighting the scope. A higher proportion of four constituted security vulnerabilities due to their noteworthy exploitation potential; the sole remaining flaw was deemed miscellaneous in nature.

To break those down, the examinations unearthed four *Medium* severity XSS vulnerabilities and a low-risk SOQL injection limitation. It's significant to note that exploiting these vulnerabilities mandates backend authentication. The majority of templates within the WordPress theme were correctly sanitized for user input, though an exception led to an XSS vulnerability in the resource download block template, as discussed in ticket AI-02-003. Supplemental investigations of the theme revealed that the removal of an internal WordPress filter incurs a persistent XSS shortcoming, as reported in ticket AI-02-002. Whilst the ReactJS framework-based custom JavaScript code generally offers stable protection, the absence of validation for iframe URLs introduced auxiliary XSS concerns, as proposed in ticket AI-02-004. Impressively, WP2 was deemed entirely risk averse due to the enforcement of optimal input sanitization. Lastly, WP3 was affected by a single XSS vulnerability due to insecure HTML attribute embedding, as described in ticket AI-02-001.

In summary, the Amnesty plugins generally garnered a favorable impression from a security point of view, though XSS vulnerabilities are undoubtedly the preeminent threat that must be resolved in due course. Fortunately, these vulnerabilities necessitated authenticated access for exploitation, underscoring the strength of the primary defense layer. With this, Cure53 is pleased to verify that the Amnesty team's implementations provide robust shielding against external breach attempts. However, one can advise proceeding with caution, since any vulnerable third-party code integrated at a later date may contaminate the core WordPress installation if mishandled.

Moving forward, the scope, test setup, and available materials are enumerated in the bullet points below. Subsequently, all *Identified Vulnerabilities* and *Miscellaneous Issues* are provided in chronological order of detection alongside a technical overview. Lastly, the *Conclusions* chapter elaborates on the general impressions gained for the in-scope features and verifies the perceived security posture, based on the evidence collected throughout this project.

Scope

- **Penetration tests & audits against Amnesty International WordPress components**
  - **WP1**: White-box penetration tests against Amnesty International WordPress templates
    - Main WordPress theme source code was provided
    - A test environment URL was provided
  - **WP2**: White-box penetration tests against Amnesty International WordPress plugins
    - **Pertinent WordPress plugins:**
      - *donations*
      - *petitions*
  - **WP3**: White-box penetration tests against Amnesty International WordPress deployment
    - **Alternative plugins, tools & configurations:**
      - *plugin-csp*
      - *plugin-media-copyright*
      - *plugin-image-credit*
      - *plugin-translation-management*
      - *plugin-donations-salesforce-adapter*
      - *plugin-salesforce-connector*
      - *plugin-petitions-salesforce-adapter*
      - *child-theme*
      - *plugin-theme-companion*
      - *plugin-github-updater*
    - **Developer documentation:**
      - *amnesty-wp-theme*
  - **Test-user credentials were available upon request**
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Identified Vulnerabilities

The following section lists all vulnerabilities and implementation issues identified during the testing period. Notably, findings are cited in chronological order rather than by degree of impact, with the severity rank offered in brackets following the title heading for each vulnerability. Furthermore, all tickets are given a unique identifier (e.g., *AI-02-001*) to facilitate any future follow-up correspondence.

AI-02-001 WP3: Attribute injection leads to XSS in image credit plugin (*Medium*)

***Fix Note:** The issue was fixed by the development team and Cure53 verified the fix as working as intended, the problem no longer exists.*

During the source code review of the image credit plugin, Cure53 noted that the *id* and *class* image attributes remain unsanitized, which permits injecting event handlers. Henceforth, malicious *author* role backend users can exploit this XSS vulnerability to hijack the session of higher privileged users.

AI-02-002 WP1: Stored XSS via unsanitized credit field (*Medium*)

***Fix Note:** The issue was fixed by the development team and Cure53 verified the fix as working as intended, the problem no longer exists.*

The test team noted that the Amnesty International WordPress theme inserts metadata into all images present in posts by default, which can contain caption or copyright information. Here, Cure53 verified that the credit or copyright field is embedded in the HTML context without sanitization. This evokes an XSS vulnerability that can be exploited by lower privileged backend users to hijack session administrators.

AI-02-003 WP1: XSS via JavaScript URL in resource download block (*Medium*)

***Fix Note:** The issue was fixed by the development team and Cure53 verified the fix as working as intended, the problem no longer exists.*

Whilst reviewing the custom WordPress theme, the audit team identified that the resource download block sanitizes the download URL in a subpar manner. As a consequence, one can place an XSS payload in an anchor tag by utilizing the JavaScript URL scheme. Once clicked, the XSS will trigger and the attacker is granted the opportunity to hijack the session of the higher privileged victim user.

AI-02-004 WP1: XSS via several iframe blocks (*Medium*)

Fix Note: *The issue was fixed by the development team and Cure53 verified the fix as working as intended, the problem no longer exists.*

Whilst vetting the Amnesty International WordPress theme's client-side JavaScript code, Cure53 detected several XSS issues related to iframes, which are facilitated because the ReactJS code neglects to sanitize or validate the iframes' URL attributes. As such, lower-privileged malicious users are able to hijack admin sessions.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Miscellaneous Issues

This section covers any and all noteworthy findings that did not incur an exploit but may assist an attacker in successfully achieving malicious objectives in the future. Most of these results are vulnerable code snippets that did not provide an easy method by which to be called. Conclusively, whilst a vulnerability is present, an exploit may not always be possible.

AI-02-005 WP3: Potential SOQL injection on Salesforce signature adapter *(Info)*

Fix Note: *The issue was fixed by the development team and Cure53 verified the fix as working as intended, the problem no longer exists.*

Cure53's assessment against the targets confirmed that the Amnesty Petitions plugin embeds potentially user-controlled post metadata from a petition directly into an SQL query of the Salesforce object query language (SOQL) dialect.

This induces a specific risk circumstance, whereby a malicious WordPress backend user with *Author* capabilities can append a malicious SQL query to a signature's post metadata in order to modify the query syntax and extract sensitive data from the Salesforce backend. However, since the Salesforce adapter was disabled within the test environment and the SOQL dialect is restricted, the corresponding ticket was allocated an *Info* severity rating.

One can deduce from the following source code that the plugin receives the post metadata stored under the *salesforce_id* key from a petition post. However, this data is attacker-controllable via configuring the signature post's custom field[1]. This data is subsequently and directly embedded unsanitized into a query sent to the Salesforce backend.

[1] https://wordpress.org/documentation/article/assign-custom-fields/

Conclusions

This pro bono engagement comprised three WPs specifically pertaining to the custom WordPress theme (WP1), the Donations and Petitions plugins (WP2), and other peripheral plugins (WP3). To increase the effectiveness of the test, a local WordPress instance was established in order to debug the theme and plugins in scope. In total, four XSS vulnerabilities (assigned a risk score of *Medium*) and one *Low* rated deficiency were located and raised as tickets for the client to consider. Albeit, the prerequisite for backend authentication in order to successfully exploit these pitfalls inherently nullifies the resulting impact.

In terms of the coverage, Cure53 would first like to highlight the initiatives conducted by the test team for WP1, which focussed on the WordPress templates:

- The initial first step was to examine the WordPress theme for adequate user input sanitization, which was confirmed according to the HTML context in which they are embedded. However, one exception was noted in this area leading to an XSS in the resource download block template, which is further highlighted in ticket AI-02-003. Considering that this represented the sole discrepancy, Cure53 can confirm that the developers are aware of the optimal methods by which to integrate context-sensitive sanitization.

- Besides the templates, the theme was reviewed to ascertain whether a malicious party could modify the HTML code. Here, the removal of an internal WordPress filter evoked a persistent XSS, as presented in ticket AI-02-002.

- Next, the custom JavaScript code based on the ReactJS framework was meticulously inspected to pinpoint any lingering vulnerabilities in the WordPress editor. Whilst the ReactJS framework offers substantial default safeguarding to repel a swathe of bug classes, the neglect to validate iframe URLs exposed alternative XSS issues, as discussed in ticket AI-02-004.

Here, Cure53 would have clarified the results of the WP2 centered undertakings against the WordPress plugins, though this realm completely avoided any exploitable vulnerabilities. This commendable outcome primarily owes to stringent sanitization of user input. Lastly, Cure53 would like to take a closer look at the observable negative connotations affecting WP3, i.e. the WordPress deployment:

- Despite extensive endeavors, the testers were only able to alleviate one XSS vulnerability, which was facilitated due to insecure embedding of HTML attributes (see ticket AI-02-001).

- Several different supply chain aspects were subjected to testing scrutiny in addition. In this regard, the repositories were explored to glean any instances of reliance

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

upon deprecated software dependencies, which may persist known vulnerabilities or omit beneficial patches. However, this area yielded a lack of associated results.

- The handling of secrets and sensitive material regarding the version control system and CI/CD was extensively probed. Similarly to other correlating areas, Cure53 was unable to unearth any glaring errors or misconfigurations in this respect. Furthermore, any references to old and outdated container runtimes were excluded. Lastly, the team could not locate any indication of sensitive material leakage during runtime with respect to either the log files (or third parties via remote logging), metric collection, or via any reporting mechanisms.

To summarize, Cure53 achieved satisfactory coverage against the audited WordPress plugins, which exhibited first-rate audibility given the relative simplicity of the entire Amnesty framework. The default security mechanisms provided within WordPress are effective in neutralizing a vast range of offensive strategies. The security HTML sanitization was disabled in one particular instance and thus evoked XSS vulnerabilities. However, Cure53 must commend the Amnesty team for their proactive efforts toward identifying and deleting the culpable entity.

Aside from this caveat, Cure53 can unequivocally confirm that the Amnesty plugins are suitably equipped to withstand many attack schemes instigated in the modern era. As mentioned previously, XSS was clearly defined as the primary antipattern affecting the compound at present. Naturally, one can strongly recommend allocating enough time and resources for the purpose of reviewing and remediating the associated flaws.

Nonetheless, one must once again reiterate that the identified vulnerabilities cannot be triggered from an unauthenticated perspective, which reflects highly favorably on the developer team’s proficient defensive layers. Cure53 discourages polluting the default WordPress installation with potentially risk susceptible third-party code in order to negate all connected harm.

Cure53 would like to thank the various internal and external teams involved for their excellent project coordination, support, and assistance, both before and during this assignment.