

# **WHMCS Security Advisory Supplemental Data Disclosure**

**WHMCS, Limited**

Thursday, April 11, 2013

<VSR #659294>

## Summary

The following is a Supplementary Data Disclosure for the Security Advisory published March 12th, 2013 <<http://blog.whmcs.com/security.php?t=69402>>. It provided specific details for reported Potential Flaws in WHMCS. All Potential Flaws described by this Supplementary Data Disclosure have been remedied by the Targeted Releases and Patches itemized in the aforementioned Security Advisory.

Six Potential Flaws were report to WHMCS by Vlad C. of NetSec Interactive Solutions <<http://safeornot.net>>. WHMCS would like to thank Vlad C. and NetSec Interactive Solutions for reporting this issues. Their professionalism and dedication to responsible disclosure practices are to be commended. A summary of their report can be found at <http://safeornot.net/advisories/whmcs>.

WHMCS has assigned all six Potential Flaws to VSR-659294 as Exhibits A-F.

WHMCS has rated these Potential Flaws as including critical and important security impacts. Information on security ratings is available at [http://docs.whmcs.com/Security Levels](http://docs.whmcs.com/Security_Levels).

## Reported Potential Flaws

The following itemizes each reported Potential Flaw. Each Potential Flaw has been evaluated as having either Critical or Important security impacts and have been confirmed as valid Vulnerabilities in WHMCS versions published prior to March 12th, 2013.

Each Exhibit has been resolved in the latest Patch files for WHMCS 4 series and Full versions of the WHMCS 5 series. Please reference the Security Advisory published on March 12th, 2013 <<http://blog.whmcs.com/security.php?t=69402>> for details and links to the Patch and Full-versions of WHMCS that are not vulnerability to these Exhibits.

### Exhibit A Summary

An attacker, with access to the “client” area of the software, can perform arbitrary SQL transactions via a SQL Injection Attack.

**Security Level**  
Critical

**Category**  
Input Validation

**Subcategory**  
SQL Injection

**Related Data**  
<http://safeornot.net/advisories/whmcs-2>

### Detailed Summary

Improper input sanitization allows an attacker to provide well-crafted URL request that results in arbitrary SQL execution.

**Potentially Flawed Implementation**  
The exposed areas are limited by

- A) Pages that are served after a successful authentication transaction
- B) Pages which offer one or more of the following:
  1. Optional user-provided ordering of returned results
  2. Optional user-provided sorting direction of returned results
  3. Optional user-provided limit to returned results

**Affected Product Version**

All releases prior to March 12, 2013.

**Intended Functionality**

- A) Provide the user with a UI that allows refinement of possible results
- B) Provide default refinement of possible results when in the absence of user refinement.

**Unintended Functionality**

The improper sanitization of result refinement parameters expose unfiltered access to the SQL engine.

**Potential Consequences**

- Any instance of the software can be compromised by an authenticated user
- Any instance of the software that is configured in a permissive account creation state can be compromised by an automated script.

## **Exhibit B**

### **Summary**

An attacker, with access to the “client” area of the software, can traverse boundaries of trust by crafting an expressive URL.

### **Security Level**

Important

### **Category**

Input Validation

### **Subcategory**

potential XSS, information disclosure, potential SQL Injection, potential arbitrary data manipulation, potential conditional process control

### **Related Data**

<http://safeornot.net/advisories/whmcs-3>

### **Detailed Summary**

Improper input sanitization allows an attacker to provide a well-crafted URL request that can present a significant risk to user data and site integrity, as well as present the attacker with the possibility for process control.

### **Potentially Flawed Implementation**

Implementations of the Flaw must meet the following

- A) Pages that are served after a successful authentication transaction
- B) Pages which populate array elements conditionally
- C) Array has come into page scope without sanitization or filtration

### **Affected Product Version**

All releases prior to March 12, 2013.

### **Intended Functionality**

There is no intended behavior. All observed cases of this implementation have been shown to be a design Flaw.

### **Unintended Functionality**

An attacker can populate specific arrays in specific pages, allowing tainted data to traverse scope.

### **Potential Consequences**

Data integrity is easily compromised since arrays can be populated with false information prior to interface rendering.

## **Exhibit C**

### **Summary**

An attacker can script arbitrary client-side behavior by crafting an expressive URL.

#### **Security Level**

Important

#### **Category**

Input Validation, Output Validation

#### **Subcategory**

XSS

#### **Related Data**

<http://safeornot.net/advisories/whmcs-3>

### **Detailed Summary**

Improper input and output sanitization allows an attacker the ability to provide arbitrary JavaScript code that can be executed by the visitor's computer.

#### **Potentially Flawed Implementation**

Data received is allowed to traverse scope without sanitization. The data is then used, unfiltered, in the presentation layer. This Flaw is present on pages which meet the following criteria:

- A) Pages that are served after a successful authentication transaction
- B) Previously stored data is retrieved and the value is trusted as valid for use in the presentation layer
- C) Specific pages that belong to a subset of those mentioned by the Potential Flaw in Exhibit B

#### **Affected Product Version**

All releases prior to March 12, 2013.

#### **Intended Functionality**

There is no intended behavior. All observed cases of this implementation have been shown to be a design Flaw.

#### **Unintended Functionality**

Client information may be harvested or data presented by the software may be altered for malicious purposes.

### **Potential Consequences**

An XSS Attack: Clients with active sessions to the software can unknowingly pass private data to an attacker. The client can unknowingly trust false information, provide by the attacker, as a legitimate response from the software.



## **Exhibit D**

### **Summary**

An attacker can script arbitrary server-side behavior by crafting a URL which contains values outside the privilege scope of the client or process owner.

#### **Security Level**

Important

#### **Category**

Input Validation, Privilege Confusion

#### **Subcategory**

Potential conditional process control, command control, information disclosure

#### **Related Data**

<http://safeornot.net/advisories/whmcs-5>

## **Detailed Summary**

### **Potentially Flawed Implementation**

Improper validation of client provided data is used by command routines responsible for aggregating service data for the client.

Implementations of the Flaw must meet the following

- A) Pages that are served after a successful authentication transaction
- B) Specific page of the software
- C) The existence of the Potential Flaw described in Exhibit B.

### **Affected Product Version**

All releases prior to March 12, 2013.

### **Intended Functionality**

Clients may check the status of servers under their control to monitor availability of services.

### **Unintended Functionality**

An attacker can abuse the intended functionality and port scan any public or private resource (relative to the network in which the hosted software instance is running).

### **Potential Consequences**

An attacker can gain information about private, internal network resources. It may be possible for improperly validated input to alter runtime conditional control, causing unexpected behaviors.

## **Exhibit E**

### **Summary**

#### **Security Level**

Important

#### **Category**

Privilege Confusion

#### **Subcategory**

CSRF

#### **Related Data**

<http://safeornot.net/advisories/whmcs-4>

### **Detailed Summary**

Improper input and output sanitization allows an attacker the ability to provide arbitrary JavaScript code that can be executed by the visitor's computer.

#### **Potentially Flawed Implementation**

Data received is allowed to traverse scope without sanitization. The data is then used, unfiltered, in the presentation layer. This Flaw is present on pages which meet the following criteria:

- A) Pages that are served after a successful authentication transaction
- B) Previously stored data is retrieved and the value is trusted as valid for use in the presentation layer
- C) Specific pages that belong to a subset of those mentioned by the Potential Flaw in Exhibit B

#### **Affected Product Version**

All releases prior to March 12, 2013.

#### **Intended Functionality**

There is no intended behavior. All observed cases of this implementation have been shown to be a design Flaw.

#### **Unintended Functionality**

Client information can be altered without the clients consent.

#### **Potential Consequences**

A CSRF Attack: Clients with active sessions to the software can unknowingly pass requests to the software on behalf of the attacker.

## **Exhibit F**

### **Summary**

An attacker can script arbitrary client-side behavior by crafting an expressive URL.

#### **Security Level**

Important

#### **Category**

Input Validation, Output Validation

#### **Subcategory**

XSS

#### **Related Data**

<http://safeornot.net/advisories/whmcs-3>

### **Detailed Summary**

Improper input and output sanitization allows an attacker the ability to provide arbitrary JavaScript code that can be executed by the visitor's computer.

#### **Potentially Flawed Implementation**

Data received is allowed to traverse scope without sanitization. The data is then used, unfiltered, in the presentation layer. This Flaw is present on pages which meet the following criteria:

- A) Pages that are served after a successful authentication transaction
- B) Previously stored data is retrieved and the value is trusted as valid for use in the presentation layer
- C) Specific pages that belong to a subset of those mentioned by the Potential Flaw in Exhibit B

#### **Affected Product Version**

All releases prior to March 12, 2013.

#### **Intended Functionality**

There is no intended behavior. All observed cases of this implementation have been shown to be a design Flaw.

#### **Unintended Functionality**

Client information may be harvested or data presented by the software may be altered for malicious purposes.

### **Potential Consequences**

An XSS Attack: Clients with active sessions to the software can unknowingly pass private data to an attacker. The client can unknowingly trust false information, provide by the attacker, as a legitimate response from the software.