FIXING THE HOLE:
DEALING WITH THE MSC
HIDDEN FORM FIELD VULNERABILITY

Prepared By

Mike Cobb, Sr. Consultant

February 11, 1998

MIORA SYSTEMS CONSULTING, INC.

INFORMATION SECURITY ▪ DISASTER RECOVERY ▪ EDUCATION & AWARENESS

P. O. Box 6028, Playa del Rey, CA 90296 WWW.MIORA.COM

310-306-1365 ▪ 310-305-1493 Fax ▪ 888-IS GUARD INFO@MIORA.COM

TABLE OF CONTENTS

1. INTRODUCTION

PURPOSE: ***Provide guidance for solving the MSC Hidden Form Field Vulnerability.***

HTML forms on web pages provide an easy way for visitors to a web site to send comments, requests, orders, and so on, from their web browser (client) to the web site (server). A CGI script on the server processes such forms, which can then record the submitted information, as well as generate an on-the-fly response. This paper provides explanation and guidance for solving the MSC Hidden Form Field Vulnerability.

Hidden Form Fields are Not Hidden

The need to pass sensitive values, such as an email address or session code, between the person using the form and a CGI script, is common among a large number of web sites. Sites engaged in online banking and online commerce, as well as sites that do not handle electronic commerce, routinely use CGI scripts that act on information contained in hidden form fields. In many sites, this need has been poorly implemented through an insecure use of hidden form fields.

MSC HFF Vulnerability leaves systems open to induced crashes, unauthorized entry, and information compromise.

In these sites that use hidden form fields improperly, the user only has to view the source for the HTML page to be able to read the hidden values, thus obtaining information useful to an attacker. Since web browsers are running on computers, rather than dumb terminals, the user has the ability to save the form, edit the HTML, and thus submit bogus and unexpected values in the "hidden" form fields. These unexpected values can cause systems to crash or can provide an attacker with unauthorized entry into sensitive systems. A detailed explanation of this problem and its implications is contained in a white paper available free of charge at http://www.miora.com.

Failure to consider the implications of using hidden form fields in this way has led to the problem becoming widespread. Many web masters, working under tremendous pressures, use HTML code from existing sites.

An InfoSec dilemma.

As information security specialists, we faced a dilemma: How to alert web site operators and network security managers around the world, as quickly as possible. Timing was critical because, at any time, an automated attack program exploiting this vulnerability could appear. Clearly a site-by-site notification process was not feasible, so we felt we had an obligation to make the problem public as soon as we were able to, at the same time, explain how the vulnerability could be eliminated. Hence this paper, which explains the vulnerability elimination process.

2. SUMMARY OF THE PROBLEM

The White Paper *Analyzing the MSC Hidden Form Field Vulnerability*, available to http://www.miora.com, explains the vulnerability in detail. This paper focuses on the solution to the problem. However, it is useful to summarize here the major vulnerabilities that constitute the MSC Hidden Form Field Vulnerability.

The Five Major HFF Vulnerabilities

- *Vulnerability One: Hidden Form Fields Break the 'Teller Window' Authentication Model*, enabling an attacker to gain unauthorized access by performing an "unexpected" activity.
- *Vulnerability Two: Hidden Form Fields Allow Unexpected Attacks on CGI Programs,* enabling an attacker to gain direct access or cause system failures by changing hidden form field values to cause CGI program errors and crashes.
- *Vulnerability Three: Hidden Form Fields Permit Unexpected User Control of CGI Programs,* enabling an attacker to gain entry into internal systems by forcing web server programs to act according to modified instructions passed via hidden form fields.
- *Vulnerability Four: Hidden Form Fields Compromise 'State' Mechanisms,* thereby permitting an attacker to hijack or replay sessions. In this way, for example, an attacker can masquerade as many different users on a banking system, instructing the system to pay out funds from other accounts.
- *Vulnerability Five: Hidden Form Fields Disclose Sensitive Information,* enabling an attacker to collect information by viewing field contents.

In order for a solution to be viable and effective, these five vulnerabilities must be addressed. The remainder of this paper will focus on solutions rather than on explanations of the problem.

3. ANATOMY OF A FIX

Two possible solution paths: Reengineering or Hiding the Hidden Form Field.

There are two techniques for solving the MSC HFF Vulnerability. One technique requires a site reengineering effort to make sure that CGI scripts handle HFF inputs securely and only are only used in a secure fashion. The second technique requires that the content of any hidden form field is truly hidden from the user.

Reengineering is large, time consuming effort.

The reengineering effort is likely to be quite large and time consuming for most web sites. Even simple sites frequently use CGI scripts for requesting information or as a mechanism for users providing feedback to the site. Moreover, even if the data being handled by the script is not sensitive, a visible hidden form field, improperly handled, may open the door to a variety of penetration techniques.

MSC has devised a solution that avoids a complete reengineering effort by hiding the contents of the HFF from average users. The advantage of this solution is that it can be implemented quickly and relatively inexpensively, while leaving the web site operational throughout the process.

3.1 The Reengineering Path

- ***Modify scripts***
- ***Remove sensitive fields***
- ***Find alternative methods for state control***

The reengineering path requires three separate actions: (1) review all CGI scripts for proper controls, (2) check all fields to ensure sensitive information is not displayed, and (3) find alternatives to HFFs where necessary, especially for control mechanisms.

The reengineering effort will need to address the five vulnerabilities described above and in the white paper *Analyzing the MSC Hidden Form Field Vulnerability*, available at http://www.miora.com. This requires changing hidden form fields so they do not contain information that breaks the "Teller Window Authentication Model." This would require that details of authentication and access would be stored in data bases or coded into CGI scripts. Any HTML-coded aids to authentication, such as *user type* or *destination* or *access level* must be removed.

CGI scripts must be recoded to treat hidden form fields the same as user inputs, with error checking, context checking, and length and value checks. These checks must be specific enough to determine that the values submitted are within a range of authorized values so that the CGI

script does not perform in unexpected ways due to re-engineered input values.

CGI scripts must also be recoded to eliminate any use of "state control" parameters as hidden form fields. Since these are not truly hidden, any attacker can modify a state control value to masquerade as another session. Since some state control mechanisms are required, this redesign will require a major change in the way web sites are built and managed.

Lastly, CGI scripts will need detailed review and modification to ensure that scripts check all input values (user-supplied and via hidden form fields) for errors. Such checks must include length, data type, range, and other standard error checks. Further, the scripts will need modification to provide error handling routines for all types of errors so that users can be informed of errors without divulging the nature of the CGI processing logic.

3.2 Hide the Hidden Form Field

Carefully designed encryption can hide the values of Hidden Form Fields

A less costly and time-consuming alternative to the site redesign is to use encryption to hide the nature and values of hidden form fields. Encryption can be used to render hidden form field values meaningless to anyone viewing the source code, and a program can be devised to do this transparently.

This program would add a security layer by interposing a process between the submission of a form field by a user and reception of the field's value by the CGI script. Similarly, the process would intercept values destined for Hidden Form Fields from the server prior to transmission to the user.

The purpose of this new process is to encrypt values bound for the user so that no user can make sense of the Hidden Form Field value. Hence, the value is hidden, even if the field is not. In this way, it would be impossible for an attacker to modify the value sensibly. On the return trip, the process would intercept the encrypted value and decrypt it for transfer to the server. In this way, if an invalid field were received by the process, the decryption would fail and the server would be protected from the transmission.

This mechanism can be developed in such a way that existing CGI scripts can continue to operate as written. Existing web pages can also continue to be used, but the

new process would need to modify existing web pages so that the secure pages replace the existing pages operationally in an automated way.

MSC has performed some testing and has developed a sample design for such an application, called here the MSC Hidden Form Field Encryptor (MSCHFFE). The program, consists of a server component and a web page conversion component, running on a workstation. The application operates using the following steps:

1. On the workstation, read an unsecured HTML form document and encrypt any values that are of the type: INPUT TYPE="HIDDEN" using a key value stored in a seed file.
2. Change the FORM ACTION value to point to the server side program that will decrypt the hidden values.
3. Add two new encrypted hidden form fields, “cgi” containing the original value of FORM ACTION, and “seed,” containing the name of an encryption seed file, which is encrypted with a built-in key.
4. Store the secure version of the form document.
5. When the form is submitted by a user, decrypt the seed value with the server component, using the same built-in key as the workstation component.
6. Decrypt the CGI hidden value using the key contained in the seed file.
7. Process the form values and POST all the name/value pairs on to the original CGI script.

In this way, the original script can run, using the decrypted values, yet be protected against false values sent by an attacker.

3.3 MSC’s Design

In order to address the MSC Hidden Form Field Vulnerability without requiring expensive redesign, MSC has utilized a transparent layer of encryption between the HTML form and the back-end CGI program. The MSC Hidden Form Field Encryptor (MSCHFFE) operates as follows:

1. Preprocess all HTML pages
   - ♦ Search for hidden form fields

- Encrypt hidden form fields
- Encrypt and store CGI program information
- Redirect form results to the Decryptor process, running on the web server

2. Process all appropriate HTML form input
   - Decrypt all hidden form fields
   - Decrypt and process CGI program information
   - POST processed HTML form results to original CGI program

All of the steps in part one can be initiated by a site administrator or web designer, once the MSC solution has been implemented. This 'preprocessing' step should be performed:

- When installing the MSC fix and initially protecting web pages.
- When the hidden form fields are updated on a page. Other web page updates and changes do not affect the MSC fix.

The remaining steps, part of the 'real-time' operation of the MSC fix, should occur automatically whenever a web user submits a form containing encrypted hidden form fields.

3.4 Implementing the MSC Design

3.4.1 Implementation for Individual Sites

Many web sites with programming staff will find that implementation of the MSC solution is the best short-term solution available. These sites should implement the MSC solution for their affected server platforms, utilizing a program as described above.

While such a solution will typically require preprocessing of web pages, it will not require any modifications to existing CGI programs. Additionally, some high-load web sites may choose to implement shared libraries based on the MSC design, for use in securing new CGI programs during the development stage.

3.4.2 Implementation for Web Server Developers

Web server developers have an opportunity to address the MSC Hidden Form Field Vulnerability using the MSC

design. By implementing the MSC solution, and thoroughly integrating it with the server's dynamic HTML capabilities, web server developers can protect all sites using their product without requiring any modification to existing sights or manual preprocessing of web pages.

Web server developers have the unique option of integrating Hidden Form Field Encryption into the web service process, making it completely transparent to the user and site administrator.

3.4.3 Implementation for Firewall Developers

Firewalls are useful partially as a result of their ability to abstract security controls to a central location. They provide a 'choke point' capable of compensating for mis-configurations and shortcomings in individuals systems. Firewall developers will want to add the MSC Hidden Form Field Vulnerability to their body of protection by implementing a variation on the MSC design.

Essentially, firewalls can implement the MSC solution by applying the discussed design at a network, rather than system, level. As a modification to existing HTML proxy processes, firewalls could transparently identify, encrypt and decrypt Hidden Form Fields as web pages pass through the firewall en route to or from a web client.

3.4.4 Implementation for Middleware Developers

Several middleware products use hidden form fields heavily in implementing their features. The developers of such products will want to implement the MSC solution in order to protect the integrity of back-end systems and data. By employing the techniques described above, middleware developers can protect their users from the MSC Hidden Form Field Vulnerability without depending on other products.

4. CONCLUSIONS

We trust that the design specifications presented in this document will lead to rapid deployment of fixes for the MSC HFF Vulnerability. We know that the design works because we have written working software based on the design (you can see a demonstration of this software at http://www.miora.com). What MSC has not done is develop software that can either be sold or given away. There are several reasons for this.

4.1 MSC is a security consulting company

MSC is not a software company. We do not devote resources to developing and supporting our own line of software. We prefer to devote our energies to our core business of disaster recovery and information system security consulting. However, MSC's technical experts are ready to assist, on a consulting basis, any organization that would like their input on the development of solutions to MSC HFF Vulnerability.

4.2 Time is not on anyone's side

The goal of creating a generic version of the fix that we have designed, capable of running on any site, on any platform, is a major undertaking. It will require considerable resources, notably person hours. We could have continued our in-house development effort until we reached such a goal, but that would have meant a considerable delay in bringing the HFF Vulnerability to the attention of those people whose sites are insecure right now.

We had to strike a balance between the time required to develop at least a working design for a solution, and the chance that someone could, at any time, publish automated attack tools to systematically exploit this vulnerability.

4.3 MSC is vendor independent

In order to attain maximum effectiveness as a security consultancy, MSC acts as an Independent Outside Trusted Authority, (IOTA™). Thus, MSC eschews all ties with vendors. We do not sell, promote, or endorse any hardware or software.

MSC prefers to let software companies develop software solutions. For our part, we stand ready to provide our security expertise to such companies, as well as to the people who are building web sites, and managing extranets, and securing intranets.