A White Paper Analyzing The MSC Hidden Form Field Web Site Vulnerability

Prepared By

David I. Brussin
Director, MSC Labs

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

ABSTRACT

Many of the Internet's most popular and sensitive sites, including those focused on electronic commerce, banking and medicine, are threatened by a web technique that seriously weakens their security. Miora Systems Consulting (MSC) has found that this technique, called the Hidden Form Field, can seriously compromise security. Moreover, firewalls are not protecting sites from this threat.

MSC Labs has found that serious Hidden Form Field vulnerabilities are extremely common in high profile Internet sites. As web designers and site administrators struggle to achieve electronic commerce and other groundbreaking functionality, they have used hidden form fields extensively. MSC Labs' penetration testing teams have confirmed these vulnerabilities during authorized tests on operational sites.

- *Vulnerability One: Hidden Form Fields Break the 'Teller Window' Authentication Model*, enabling an attacker to gain unauthorized access by perform 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.

MSC has designed a potential solution to the MSC Hidden Form Field Vulnerability, information on which is available free of charge to the Internet community at http://www.miora.com. MSC's solution builds a protective layer on the web server to transparently address the major vulnerabilities of the Hidden Form Field. This type of fix does not require costly redesign of web sites.

Web administrators should identify all of the hidden form fields in their pages to determine how they are being used. In addition, web sites suffering from the MSC Hidden Form Field Vulnerability should be systematically reevaluated to ensure that the sites are built on a strong foundation. Many organizations can deal much more effectively with emerging issues such as the MSC Hidden Form Field Vulnerability by utilizing an IOTA™, or Independent Outside Trusted Authority. Particularly in complex areas such as design integrity and security, it is vital to have independent, expert advice.

TABLE OF CONTENTS

1. INTRODUCTION

The Internet's most popular and sensitive sites, including banking and medicine use a web technique that seriously weakens security.

Many of the Internet's most popular and sensitive sites, including those focused on electronic commerce, banking, medicine, and even Internet infrastructure, use a web technique that seriously weakens their security. Miora Systems Consulting (MSC) has found that hidden form fields, an element of web pages commonly used by web authors and designers, can seriously compromise security. The MSC Hidden Form Field Vulnerability is the result of this web design being misused across the Internet in ways that weaken the fundamental security of Internet sites. The nature of this weakness is such that firewalls do not protect against it, although MSC has proposed a solution that could be implemented on firewalls (see Section 4).

MSC Labs has found Hidden Form Field vulnerabilities in many high profile Internet sites.

MSC Labs has found that serious Hidden Form Field vulnerabilities are common in high profile Internet sites. The accelerated development of the World Wide Web and HTML has resulted in functionality being placed before security and design considerations. As web designers and site administrators struggle to achieve electronic commerce, connectivity to live databases, and dynamic web content, they use available tools to bridge the gaps in a constantly evolving environment. They have used hidden form fields extensively to this end. Hidden form fields have provided expanded functionality to CGI programs, maintained 'state' information before 'cookies' were available, and provided a mechanism for implementing security features.

More complex than a bug that a simple vendor patch could address.

Unfortunately, these very examples clearly show how hidden form fields can severely weaken the 'design security' of web sites. Moreover, these weaknesses are not as simple to fix as the average CGI program bug or a web server-specific hole patched by a vendor; they are more subtle, more pervasive and prevalent, and cannot be addressed without serious attention to underlying security issues, process, and implementation.

MSC is providing its design for a fix to the Internet community -- at no charge.

MSC has designed a solution to the MSC Hidden Form Field Vulnerability. It is called the MSC Hidden Form Field Encryptor (MSCHFFE), and it builds a protective layer on the web server to transparently address the major vulnerabilities of the Hidden Form Field. This type of fix does not require costly redesign. The design is being made available by MSC free of charge to the Internet community.

2. DEFINITION OF PROBLEM

2.1 Technical Definition

Hidden Form Fields contain information not displayed by the browser but passed to CGI scripts.

In defining the MSC Hidden Form Field Vulnerability, we must first define the hidden form field. These special fields, found on user-submitted HTML forms, contain information that is, by default, not displayed by the web browser. This information is passed along to the CGI programs, along with user input, in order to assist in processing in some manner. To find hidden form fields in an HTML document, a user would select the 'view source' option in their web browser and look for fields laid out as follows:

```
<INPUT TYPE="HIDDEN" NAME="abc" VALUE="defghijk">
```

As web content became more interactive with the advent of forms and user response mechanisms, web designers were faced with several problems:

1. The protocol on which the web is based (HTTP) is 'stateless,' meaning it doesn't have the ability to recognize or identify the fact that certain requests are part of one user's 'session.'
2. CGI programs can be expensive to build, so a mechanism permitting their reuse is very attractive.
3. As designers considered moving sensitive information to web servers, they needed a mechanism for implementing some security requirements.

Hidden Form Fields can be a solution to some basic HTTP limitations.

Web designers see hidden form fields as a solution to all of these problems. Hidden form fields store session identifiers to preserve state information, provide a variety of different run-time parameters to CGI programs, and preserve security and user authentication information for use in electronic commerce applications.

Hidden form fields can open systems to compromise of sensitive data.

Unfortunately, many web designers do not realize the serious security implications of their actions. There are a number of serious vulnerabilities associated with the use of hidden form fields, any of which could potentially be exploited by an attacker to compromise sensitive data and systems. Also, the MSC Hidden Form Field Vulnerability is an 'Allowed Path' vulnerability, so it is not addressed by firewalls, which are primarily designed to block non-allowed traffic.

There are five major parts to the MSC Hidden Form Field Vulnerability, impacting critical and diverse segments of web server, electronic commerce, and Internet architecture operation.

2.1.1 The 'Teller Window' Authentication Model

Hidden Form Fields Break the 'Teller Window' Model of Authentication.

Hidden Form Fields Break the 'Teller Window' Authentication Model. Consider the teller window at a bank. When you walk up to the teller window, you are asked for certain pieces of information, based on the type of transaction you want to perform. If, for example, you are making a deposit, you will be asked for a deposit slip with your account number, plus the currency or endorsed checks that you are depositing. If making a withdrawal, however, you might be asked for a withdrawal slip and some form of identification.

The 'Teller Window' Model works because the teller applies bank policy and common sense to all transactions.

This 'Teller Window' Authentication Model works because the bank teller consults specific bank policy regarding the procedure for a given transaction. If the bank teller expected or allowed the customer to dictate procedure for a transaction, the bank would soon find a line of swindlers waiting to make fraudulent transactions.

Consider an example of a broken teller window model: You walk up to the counter and tell the clerk that your are making a deposit. However, instead of handing the teller a deposit slip, you present a withdrawal slip. The teller is thinking "deposit," so does not ask for identification, but instead processes the withdrawal and hands over the cash. This sounds absurd, but that is because we are very familiar with the properly functioning teller model, and because human tellers are better equipped to know when a situation "feels" wrong than are their digital counterparts.

Hidden form fields provide an attacker with an element of control over the authentication process.

Hidden form fields are frequently used to provide run-time options to CGI programs, thereby allowing greater flexibility and code re-use. The many general problems associated with such use of hidden form fields will be discussed later. Here we are concerned with the specific case of CGI programs involved with an authentication subsystem.

There is no accepted standard for security on web sites. Administrators seeking more than simple HTTP access control have had to turn to proprietary products, or build solutions in-house. Unfortunately, many of these solutions have used hidden form fields to control or customize the authentication process. Essentially, this provides the

attacker with some element of control over the authentication process.

A hidden form field that broke the Teller Window model, enabled MSC Labs to log in with greater authority than they should have had.

Consider an example of CGI program re-use through hidden form fields breaking the Teller Window Authentication Model: On one site, different types of users utilized different web pages to log-on to a system. One type was 'customer,' which could perform certain restricted functions; another type was 'vendor,' which could perform numerous sensitive operations across the entire system. When a user of either type logged on from their respective page, username and password were checked against a database and they were logged in with the appropriate authority and options.

Since these login functions were similar, it made sense to the web designer to re-use the same CGI program, and to provide options via hidden form fields. The only distinction between a customer and a vendor was made through a 'user type' hidden form field located on the login pages. The field contained 'customer' on the customer page, and 'vendor' on the vendor page, and the CGI program provided authority and options based on that field. Since this hidden form field provided the user with an element of control over the authentication process, it broke the Teller Window model, and allowed MSC Labs penetration testers to login with greater authority than they should have been given.

On another system, the MSC Labs team used a hidden form field to gain access to sensitive portions of a system.

Consider another example: The login page on a sensitive project management system overtly requested only a username and password. Behind the scenes, however, this login page contained a hidden form field with the destination directory the user should reach upon successful authentication. MSC Labs' team quickly used this field to gain access to sensitive portions of the system.

It is important to realize that any impact the user has on an authentication process, no matter how small the impact appears to be, weakens that security system. That decrease in protection may be sufficient to break the process.

2.1.2 Unexpected Attacks on CGI Programs

Check all login HTML pages for hidden form fields.

While some web designers realize that the contents of hidden form fields are not so hidden, they do not realize that hidden form fields are 'user submitted content,' just as any other form field would be. When used to provide run-time options to CGI programs, as well as to perform other tasks for the web designer, it is assumed that these fields

will be submitted as designed. In fact, an attacker can modify these forms to create unexpected results in CGI programs.

Consider a web page dedicated to collecting feedback information from users. It might do extensive checks on the size and content of user submitted fields, in order to prevent attacks. If this page contains a hidden form field with, for example, an identifier for type of user information being gathered, that form most likely is not considered user-submitted content, and is not checked. An attacker could carefully modify the contents of that hidden field, creating a "buffer overrun" situation within the CGI program on the web server, thus compromising the system.

There is also a more subtle problem. It is possible to modify a hidden form field in such a way that the length is still legal, the characters used are still legal, but the field contents have been modified in such a way that the CGI program performs unexpectedly. For example, if an attacker removes the content of a hidden form field, the corresponding CGI program could possibly “crash” or deliver sensitive data.

It is vital to realize that all form fields, hidden or not, are user-submitted content.

It is vital to realize that all form fields, hidden or not, are user-submitted content. All of these fields may be modified for content or length by a user or attacker, and must be checked by the CGI program or validated in some other way. Typical checks include length, allowable characters, and allowable content. This final check, allowable content, means that extensive error checking must be performed to protect the CGI program from unexpected results.

2.1.3 Permit Unexpected User Control of CGI Programs

When hidden form fields are modified to contain legal information that will pass all safety checks, it is extremely difficult to determine whether or not that legal content will create the intended result, or some nefarious one. This is more difficult to address than the problem described in the previous section, which discussed the fact that hidden form fields could be modified by attackers to contain ‘illegal’ and unintended content, causing system failure or unexpected behavior.

For example, MSC Labs’ researchers were able to attack sensitive systems behind a firewall by exploiting the hidden form fields in a page that called a common CGI ‘e-mail response’ program. The CGI program was designed to

process user feedback and requests on web sites, and accept hidden form field input to specify the destination address for the e-mail.

While several well-known vulnerabilities with such 'e-mail response' programs center on the CGI program failing to check the contents of a form field, the more subtle vulnerability found here was in the presence of the field itself. MSC Labs' researchers were able to specify arbitrary, but legal e-mail recipients, then use 'Sendmail' attacks against systems behind the firewall. In this case the hidden form fields provide the attacker with some unneeded element of control over the behavior of a CGI program, something a web client should never have.

2.1.4 Compromise 'State' to Permit Hijacking and Replay Attacks

Hidden form fields are often deployed to maintain state information for web servers, since those web servers typically have no mechanism for handling that information. Hidden form fields can enable compromise of these 'state' mechanisms, permitting simple hijacking attacks, as well as replay attacks.

Compromised state information enables hijacking and replay attacks.

An electronic commerce server, for example, might use hidden form fields to store a 'session number,' so that a user could log in to the system once, then perform a series of functions, such as adding items to a "shopping cart." Without some type of state information, the web server would be forced to re-authenticate the user for every operation. Since hidden form fields are vulnerable to modification, an attacker could log in to the system, modify his session number, and hijack the sessions of other users. On systems such as on-line banking and electronic commerce, this could mean that an attacker with one account can attack numerous other accounts.

Another common method used to maintain state on web servers involves storing the username and password, or sometimes an encrypted hash of the password, in a hidden form field. When this is done, the hidden form field contents can be obtained from client systems by an attacker, and used later in a 'replay' attack.

Check all HTML pages. If hidden form fields are present and are being used to maintain state, implement the MSC solution discussed below.

2.1.5 Disclose Sensitive Information and Details of CGI Program Operation

In addition to the 'active' attacks discussed above, there are 'passive' vulnerabilities that can disclose sensitive information and facilitate further attacks. Essentially, the contents of hidden form fields are not at all hidden or even obscured, so an attacker can retrieve any username and password data, encrypted authentication data, or other sensitive information. Since users are often not aware of the dangers of leaving their client systems unattended, they can be lulled into a false sense of security by the fact that their sensitive information does not appear on the screen.

Also, the use of hidden form fields to provide options and run-time information to CGI programs gives attackers access to an undesirable level of detail about the operation of those CGI programs, including the options and their format. This information can facilitate more sophisticated attacks.

2.2 Scope of Problem

Check all HTML pages. If hidden form fields are present, and are used to store sensitive information or CGI program options, implement the MSC solution discussed below.

The MSC Hidden Form Field Vulnerability can be found on a majority of the most popular electronic commerce, banking, medical, and other sensitive sites. This problem is extremely pervasive, and has been located in every major sector, across thousands of sites. MSC estimates that 90 percent of active sites use hidden form fields.

The prevalence and nature of this problem suggest larger issues, especially the lack of security consideration during the design phase of Internet projects. Many organizations can deal much more effectively with security issues by utilizing an IOTA™, or Independent Outside Trusted Authority. Particularly in complex areas such as design integrity and security, it is vital to have independent, expert advice. Professional penetration testing should also be considered as a means of validating designs. You can read MSC's article on Professional Penetration Testing at http://www.miora.com/Articles.htm.

3. ADDRESSING THE PROBLEM

MSC has implemented the solution to the MSC Hidden Form Field Vulnerability and has made it available at no charge to the Internet community.

There are three basic ways to address the MSC Hidden Form Field Vulnerability:

1. Eliminate all hidden form fields from web sites.
2. Redesign CGI programs, and eliminate some hidden form fields.
3. Introduce a protective element on the web server that addresses the MSC Hidden Form Field Vulnerability.

The first two options would require significant expenditures of time and energy, while at the same time compromising web site functionality. The third option provides an efficient solution that does not require time-consuming changes in web pages, or even recompiling of existing CGI programs.

An effective solution to the MSC Hidden Form Field Vulnerability itself should:

- Automate any web page updates required by the process.
- Act as a transparent layer between web form submission and CGI program operation.
- Address each of the five major vulnerabilities that comprise the MSC Hidden Form Field Vulnerability, as discussed above.
- Be independent of web browsers.
- Be available for major web server operating systems.

MSC has designed a solution to the MSC Hidden Form Field Vulnerability. This solution meets all of these requirements, and the design is available at no charge from http://www.miora.com. For a detailed discussion of the design and implementation of this solution, please see "The MSC Fix" below.

4. THE MSC FIX - THE MSC HIDDEN FORM FIELD ENCRYPTOR

***MSC Hidden Form Field Encryptor (MSCHFFE) offers a solution -- the design is available from* www.miora.com, at no charge.**

MSC has designed a solution to the MSC Hidden Form Field Vulnerability, building a protective layer on the web server that transparently addresses each of the five major vulnerabilities discussed above. MSC's solution may well be easier to implement than costly redesign or re-coding.

4.1 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 Encryptor 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.

4.2 Implementing the MSC Design

4.2.1 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 CGI 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.

4.2.2 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.

4.2.3 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.

4.2.4 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.

4.3 Features of the MSC Design

The MSC solution is an appropriate long-term solution.

The MSC solution resolves all of the issues discussed in "Addressing the Problem" above, and is an appropriate long-term solution for any web site currently exposed to the MSC Hidden Form Field Vulnerability or that may implement form fields in the future.

4.3.1 Protects Against the Five Major Vulnerabilities

The encryption techniques used in the MSC solution to the MSC Hidden Form Field Vulnerability provide several important guarantees regarding the contents of the fields, mitigating the risks brought about by the individual vulnerabilities associated with hidden form fields.

- **Integrity** - encryption provides assurances that the hidden form contents have not been modified between their transmission to the web client and their return to the web server.
- **Authentication** - the uniqueness of the encryption key corresponds to an assurance that the encrypted fields did in fact originate from the receiving server.
- **Confidentiality** - the most traditional assurance associated with encryption techniques is confidentiality of the encrypted data, in this case encryption protects sensitive information stored in hidden form fields.

4.3.1.1 Resolve Vulnerability One: Hidden Form Fields Break the 'Teller Window' Authentication Model

Encrypt hidden form field contents.

MSC's solution addresses the Teller Window Vulnerability by eliminating the unwanted element of user control. Since the hidden form field contents are encrypted, they cannot be modified or controlled by the user.

4.3.1.2 Resolve Vulnerability Two: Hidden Form Fields Allow Unexpected Attacks on CGI Programs

Encryption minimizes the risk of unexpected attacks.

MSC's solution minimizes the risk of unexpected attacks by making such attacks much more difficult. It is vital, however, that web designers code CGI programs to control the size and content of all user-submitted data. See the WWW Security FAQ and the CGI Security FAQ for more information on coding CGI programs securely:

http://www.w3.org/Security/Faq

http://www.cgi-resources.com/Documentation/Security

4.3.1.3 Resolve Vulnerability Three: Hidden Form Fields Permit Unexpected User Control of CGI Programs

Encryption eliminates attackers' ability to modify values.

By eliminating the ability for users to intelligently control CGI programs by submitting legal, yet inappropriate parameters, MSC's solution addresses the Unexpected User Control Vulnerability. Since the hidden form field contents are encrypted, they cannot be modified or controlled by the user.

4.3.1.4 Resolve Vulnerability Four: Hidden Form Fields Compromise 'State' Mechanisms to Permit Hijacking and Replay Attacks

Encrypt state information to prevent replay and hijacking attacks.

Encryption protects the state information, as well as any encrypted authentication tokens in MSC's solution, so the above detailed hijacking and replay attacks would not be possible.

4.3.1.5 Resolve Vulnerability Five: Hidden Form Fields Disclose Sensitive Information and Details of CGI Program Operation

Encryption protects the confidentiality of information stored in hidden forms fields

The encryption in the MSC solution protects the confidentiality of sensitive information that might be stored in hidden forms fields. Detailed information on the operation of CGI programs is also protected, reducing the capability of an uninformed attacker to test the limits of a web server.

4.3.2 MSC's Solution Can Be Implemented to Automate Web Page Updates

A solution that requires manual updates to web pages would not be practical for a web site of even moderate size. Such a solution would require redesigning the web site, its CGI programs, and would probably require elimination of some site capabilities. It would be beneficial to implement

mechanisms to automates the update process, dramatically reducing the implementation time for the fix.

4.3.3 MSC's Solution Acts as a Transparent Layer between Web Form Submission and CGI Program Operation

No changes are required to existing CGI programs.

No changes are required in existing CGI programs. Since the MSC fix is a transparent layer between form submission and execution of the original CGI program, MSC's solution can be quickly and easily deployed.

Note, however, that CGI programs should be written in compliance with strict security standards. While MSCHFFE mitigates the risks associated with the MSC Hidden Form Field Vulnerability, it does not and could not address the multitude of other known security problems.

4.3.4 MSC's Design is Web Browser Independent

MSC's design works with any web browser.

MSC's solution should work with any web server and web browser capable of handling forms, meaning any modern server and browser. In implementing the solution, it is only necessary to code the solution to work with the server platform in question. Any browser currently unable to handle forms is by definition not vulnerable to the MSC Hidden Form Field Vulnerability.

4.4 Static and Dynamic Hidden Form Fields

Static hidden form fields are the same for each web browser accessing the same URL, while dynamic hidden form fields can differ based on the user and session accessing a URL. Both types of hidden form field are vulnerable to the MSC Hidden Form Field Vulnerability.

The preceding design is principally the same for a dynamic hidden form field solution, with one change: The preprocessing stage must be executed at 'run-time,' between the creation of the HTML document containing the hidden form fields, and the presentation of that document to the user. The manner of implementing this dynamic preprocessing layer will differ for each type of web server.

5. CONCLUSION

The vulnerability is real, common, and impervious to firewalls.

MSC Labs' penetration testing teams have confirmed these examples by compromising numerous sites during authorized tests. They have found that the hidden form fields weaken the HTTP path, which is allowed through most firewalls. This weakness is not something that can be dealt with at a firewall or proxy, but must be handled in the content on the web server.

Weaknesses in the design of a web site are much harder for administrators to deal with than single vulnerabilities. When a bug in a common CGI program is discovered, an aggressive administrator can take steps to check systems and upgrade, patch, or even remove the offending code.

Web sites suffering from the MSC Hidden Form Field Vulnerability should be systematically reevaluated to ensure that the sites are built on a strong foundation.

Web sites should implement a fix and perform a systematic site security review.

Web administrators should identify all of the hidden form fields in their pages and determine how they are being used. Many organizations can deal much more effectively with emerging issues such as the MSC Hidden Form Field Vulnerability by utilizing an IOTA™, or Independent Outside Trusted Authority. Particularly in complex areas such as design integrity and security, it is vital to have independent, expert advice.

6. About Miora Systems Consulting, Inc.

Dedicated to identifying and minimizing security risks.

MCS's world-class experts develop technical solutions, offer innovative training programs, and heighten employee security awareness

Providing information security services to corporations and government agencies, both in the U.S. and around the world, MSC is dedicated to maximizing the benefits of information technology by identifying and minimizing associated security risks.

Using the innovative IOTA™ approach, MSC acts as an independent, trusted, outside authority in the areas of Information System Security, Network and Internet Security, Disaster Recovery Planning and related fields. MSC's pioneering vulnerability analysis finds the holes in system security. MCS's world-class experts develop technical solutions using MSC Labs. Innovative training programs such as WISE[SM], the Web-based Information Security Education service, heighten employee security awareness.

MSC has provided information security services to a broad range of industries, from banking and finance to manufacturing, distribution, retail, and service companies, as well as government agencies, since 1988.

Miora Systems Consulting, Inc.
P. O. Box 6028, Playa del Rey, CA 90296
8055 W. Manchester Ave. Suite 450, Playa del Rey, CA 90293
Phone: 310/306-1365; 310/305-1493 Fax
Toll Free: 1-888-IS GUARD (1-888-474-8273)
info@miora.com
http://www.miora.com