Achieving Open Source Software Assurance with Steganography

Robert James
Department of Mathematics and Computer Science
University of Detroit Mercy, Detroit, MI 48219, USA
jamesrr@udmercy.edu

Submitted to
The 2005 International Conference on Software Engineering Research and Practice

Presenter: Robert James

Keywords

Steganography, Open Source Software, Information Security, Web Services, Security Policies

ABSTRACT

Putting a system on a local area network is, in itself, a security problem. Putting a web server on the network causes an increased security concern. Allowing anyone to put a web server on the network, without proper security policies for configuration and administration, can create a security nightmare which can lead to compromising of network services and data secrets. Overt methodologies and policies to find servers, along with open services on the network need to be implemented. In this paper, open source web services security will be investigated using steganography techniques. In particular an Apache web server configured to hide and allow transmission of data hidden within the web page will be considered. Open source web services and steganography in relation to how it can be used in a web environment will be discussed. Potential policies and certification practices will be discussed.

Keywords

Steganography, Open Source Software, Information Security, Web Services, Security Policies

1. INTRODUCTION

Over the past couple of years, steganography has been the source of a lot of discussion, particularly as it was suspected that terrorists connected with the September 11 attacks might have used it for covert communications [1]. While no such connection has been established, the concern points out the effectiveness of steganography as a means of obscuring data. Along with encryption, steganography is one of the fundamental ways by which data can be kept confidential [4]. There are numerous ways for implementing the security breaches within the infrastructure, most of which involved web services [2]. This involves the posting of content on a web page, mainly, pictures, images and the web pages themselves. The client would browse and view the web page, view the source page code, then save the page and/or the image in a local directory. With the proper software and codes, the client would be able to extract data from within the picture or image. With a little ingenuity, the client can take a web page, save the source page, and then with open source steganography programs, extract a message from within the page. It is very easy to put an operation in place, under the nose of unwary security policies [6].

In this paper, we will concentrate on achieving open source security assurance with steganography. In section two, methods of detecting data within the web page will be introduced. Section three will discuss a web page author's style of writing. Section four will describe policies which can be taken to assist in regulating security for web pages. Finally, section five draws some conclusions and points to areas of future work.

2. HIDING DATA IN THE WEB PAGE

There are many ways to hide data within a web page or server pages. With the use of steganography, data can be encrypted in images, music files, and video files, along with hiding in the web page itself. Hiding data in the web page is dependent on the size and complexity of the page itself [9]. The page can be very large in plain text, but provide little space for hiding information. To be able to hide a reasonable amount of data, there must be a large quantity of HTML tags available [10]. The difference between normal text and HTML tags is that HTML can be read in numerous ways, and still have the page look normal. When displaying text, the sentence or paragraph is displayed as written. If data was hidden within the page, decryption routines such as the Grill method [14] would need to be employed to find the hidden data. Therefore a re-arrangement of the sentence structure would not be the preferred methodology to hide information within a web page. As an example, let's consider the following, simple web page:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<! "http://www.w3c.org/TR/REC-html40/strict.dtd">
<html>
<head>
<center><b><font size=4>IRIX 6.5 Install procedure
<b> </b></font></b></center>
<i><b><p>  There are now 2 procedures for reloading a client
      <br> A) go to section 3a when the system is still on the network
      <br> B) go to section 3b when the system is off the network
</b></i></p>
```

```
<i><b><p>3a.)    Procedure for reloading the client, when the client is still on the network.
       <br> # Remove the root password on the client to be loaded
        <br># remove the "*" from the "guest" password field
        <br>## Clone the client tree.
        <br># rlogin ae9601  -l root
        <br># install_client
        <br> - Enter the hostname of the system to be upgraded
        <br>    - Answer yes to the question asking to continue the autoinst process
        <br>    - Answer yes to the question to push the init state
        <br># Go to step 4..
</b></i></p>
<i><b><p> 3B.)    Procedure for reloading the client, when the client is shutdown
        <br># rlogin &lt;client's hostname&gt; -l root
        <br># init 0
        <br>##      Clone the client tree.
<i><b>  <br># rlogin ae9601  -l root
        <br># install_client
        <br>        - Enter the hostname of the system to be upgraded
        <br>        - Answer yes to the question asking to continue the autoinst process
        <br>        - Enter the appropriate CPU Type.
        <br>        - Enter the appropriate Graphics Type.
        <br>       - Enter the size of the system disk in MB.  (Ex.  4.3 GB = 4000)
        <br># Go to Step 4
<i><b><p>4.)     From the PROM menu of the unix client, get into the command monitor, and type
        <br> setenv diskless 1
        <br> setenv netaddr &lt;client's IP&gt;
        <br> setenv OSLoader unix
        <br> setenv SystemPartition bootp()ae9601:/var/boot/&lt;client's hostname&gt;
        <br> auto
 <div class="footer"></div>
<p>Content updated by<i><a href="mailto:rjames2@ford.com"> Bob James </a></i>.
<br>Last updated: Tuesday, 17-Dec-2004 15:00:38 EST
</body>
</html>
```

This page has no available space to insert a message, due to the large amount of textual data, and the low number of HTML tags.

Webpage authors have their own way to make a web page, thus making it very easy and unobtrusive to catch the problem. The following lines of HTML have the same meaning:

```
<img alt=”A picture” src=”picture.jpg”>
<img src=”picture.jpg” alt=”A picture”>
```

Both statements, in a web page will display the same picture when looked at through the browser. The interesting piece is that each web author has a standard methodology of writing pages, as such, a signature. A specific program can analyze the page, line by line, to see if the statements are similar. If so, the results of the program can determine if the page is potentially hiding a message. As an example, let's consider the following web page:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<! "http://www.w3c.org/TR/REC-html40/strict.dtd">
```

```
<html>
<head>
 <TITLE>Client Implementation Menu</TITLE>
  <BASE href="http://www.client.com/t2000/">
 </head>
 <BODY bgcolor="#C0B0D0" background="graphics/menuback.gif" LINK="blue"
VLINK="blue">
  <P class=cntr> <center> <A href="index.html" target="_top"> <IMG border=0
        src="graphics/undcenter1.gif" alt="Client Home Page."></A></P>
   </center>
   <A href="Client_Implementation/index.html" style="text-decoration : none"
target="_top"><font color="white"> <b>Client Implementation</A></font><br>
 </BODY>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
```

```
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<br> 
<div class="footer"></div>
<p>Content updated by<i><a href="mailto:rjames2@ford.com"> Bob James </a></i>.
<br>Last updated: Tuesday, 17-Dec-2004 15:00:38 EST
</body>
</html>
```

By looking at the placement of each HTML tag, it is obvious that the tags are in the same order, on each of the lines. When designing a web page, the author will put similar commands in the same order, throughout the page. This allows for quicker assembling of a page, using “cut and paste” routines to replicate similar lines of code. The standardization is common, which could be possibly seen as a “signature” of the author.

Re-arranging the tags in a known order and series, the word “test” has been hidden in the following web page. In examining the page, we will only see the tags and not the word “test” as it is hidden.

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<! "http://www.w3c.org/TR/REC-html40/strict.dtd">
<html>
<head>
<TITLE>Client Implementation Menu</TITLE>
<BASE href="http://www.client.com/t2000/">
</head>
<BODY
VLINK="blue" background="graphics/menuback.gif" LINK="blue" bgcolor="#C0B0D0">
<P class=cntr> <center> <A target="_top" href="index.html"> <IMG alt="Client Home Page." border=0
src="graphics/undcenter1.gif"></A></P>
</center>
<A
target="_top" style="text-decoration : none" href="Client_Implementation/index.html"><font
color="white"> <b>Client Implementation</A></font><br>
</BODY>
<td VALIGN=CENTER BGCOLOR="#003399" ALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td BGCOLOR="#003399" VALIGN=CENTER ALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER BGCOLOR="#003399" ALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER ALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER ALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER ALIGN=CENTER BGCOLOR="#003399">
```

```
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td BGCOLOR="#003399" ALIGN=CENTER VALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER VALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER ALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER ALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td VALIGN=CENTER ALIGN=CENTER BGCOLOR="#003399">
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER BGCOLOR="#003399" VALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER BGCOLOR="#003399" VALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER BGCOLOR="#003399" VALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER BGCOLOR="#003399" VALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<td ALIGN=CENTER BGCOLOR="#003399" VALIGN=CENTER>
<div class="subwebtitle">Engineering Workstation Operations</div>
</td>
<br> 
<div class="footer"></div>
<p>Content updated by<i><a href="mailto rrmfjames@peoplepc.com"> Bob James </a></i>.
<br>Last updated: Tuesday, 17-Dec-2004 15:00:38 EST>
</body>
</html>
```

The only difference in the documents, when the text has been embedded, is the re-arrangement of the tags. This re-arrangement is done based on a formula, such as used in the program Deogol [15]. The program calculates the number of lines which can accept a tag “M”. As the program proceeds through the document, and encounters a line with “n” elements, It calculates “M' = M div n!” and “p” based on “p = M mod n!”. (The number p is a number between 0 and n!-1). The program, such as Deogol, will then transform or re-arrange the line of tags, based on this permutation.

3. DETERMINING WRITING STYLE

An author’s style of writing is akin to a painters brush stroke. As with a novel, a web page can be a passion of its writer and author**.** When assembling a web page, a writer, usually working alone or in small groups, will put together several lines, and then duplicate the lines elsewhere in the code with similar functionality. When reviewing this style several hypotheses can be formulated:

a. Lines of code written by the same author will have similar traits**.**
b. Sections of code performing similar functions will have similar traits
c. Pages of code, providing similar functionality will have similar traits.

By examining the authors style and finding changes in the style throughout the pages, a determination can be made that hidden messages could be in the documents.

In the evaluation of a web site there must be several tools available. First, a program to parse an HTML file to evaluate each line is needed. As the file is parsed, a comparison is made to the other lines. This comparison continues throughout the document, until all of the lines have been scored. The total score that could be assigned to a page is ten. All pages are evaluated. The total assigned for each web page is then compared with the other web pages contained within the web site.

To evaluate the web page, each line of code will be parsed and tags extracted from the lines. The parser will continue through the page evaluating each line. As the lines are evaluated, a score is given as follows:

2 points, if the tags are found in the same order.
1 point, if the majority of tags are found in the line, and in the same order
0 otherwise

Each line is given a score, and the page is given an overall score. The higher the score, the more likely the web page is considered to have the same similarities and thus having a "signature" of the web author. Evaluating the example in section 2, the first web page would be given a score of 480. The second example would be given a score of 120, showing that the page has very little similarities, thus requiring a closer look.

4. ASSURANCE WITH STEGANOGRAPHY

From the example outlined in section three, steganography can be used to hide data and transmit this data anywhere the internet is available. On the other hand, steganography can be used to verify that web pages have not been altered. Using the same techniques, known data can be hidden within the pages. This data can be certification dates, times and even a certification number assigned to the author or department. If the pages are checked on a regular basis, the data can be extracted and verified. If the certification data cannot be extracted, or the data is incorrect, the web site can be checked for intrusion and action taken to remedy the situation.

5. SECURITY POLICIES

With the use of steganography, the intended recipient can retrieve information from anywhere in the world, as could someone responsible for security. Albeit tedious, searching through web pages can verify if information is being passed into or out of the intranet [8]. To assist in identifying web authors who are working to provide their company with a value-added product, and to assist in identifying web sites, with potential security problems, the following policies are necessary:

- There must be an interest in securing data and web servers within the organization.
- A comprehensive policy must be made to assure steganography is addressed, and to assure the web authors that stiff penalties will be leveled for breach of security.
- There should be a method of registration for each web site in the company. This registration should include a department name with each site, and a manager, who would be responsibly for the content.
- All web pages in the company should adhere to a strict page format, with standard logos and page typeset. There must be an author name and a department contact on each page.
- A policy is needed to ensure each web site is registered each year and audited for content. This policy would include periodic audits of registered web pages to assure content integrity.
- A policy should be set to force security scans of the network for unregistered web servers. If found, the network port should be shutdown until the page is registered.

6. CONCLUSIONS

Steganography is a fascinating and effective method of hiding data that has been used throughout history. In this paper, a method to pass hidden data, using the internet, was explored. The reverse was also investigated, to assure the web pages have not been altered. Although both methods are simple and easy to implement, the altered page is still able to hide amongst other pages, virtually going undetected, “in the crowd”. Other methods that can be employed to uncover such tactics [11], but the first step is awareness that such methods even exist. There are many good reasons as well to use this type of data hiding, including watermarking [7] or a more secure central storage method for such things as passwords [3]. Regardless, the technology is easy to use and difficult to detect. The more we know about its features and functionality, the more ahead will be in the game.

Security issues are an ongoing problem related to the web serving process. Everyone wants to have their information shared across an internet or the extranet. Web servers can also be installed at will, even for help page utilization. The problem is that a web page can carry more information about your internal business structure, without us even knowing. Web servers, and open source make the web much more robust, with updates and enhancements daily. If we are involved with infrastructure security, it is our responsibility to identify and secure web sites on our wire. Intruders aiming at gaining access to a network or data will continue until they have succeeded. This means that staff responsible for security must know all of the tricks of the trade, and use them to secure your valuable data.

References

[1] Cole, Eric. “Hiding in plain sight, Steganography and the art of covert Communication”, Indianapolis, Indiana; Wiley Publishers, 2003, pp.9.

[2] Borders, Toni 16 December 2002, “Steganography Policies for protecting Your Web Site”, GIAC Security Essentials Certification, SANS Institute, 2003, pp. 8-12.

[3] D. A. Wheeler. (2003, November). How to evaluate open source software, [Online]. Available: http://www.dwheeler.com/oss_fs_eval.html

[4] (2004, January). Packet Storm Security, [Online]. Available: http://www.packstormsecurity.com/index.html

[5] Anderson, Ross. “Security in Open Versus Closed Systems- The Dance of Boltxmann, Coase and Moore, 2004, Cambridge University, England, pp. 10.

[6] N. Provos. (2004, January). [Online]. Available: http://www.outguess.org/index.html

[7] K. Westphal, (2003, April). Steganography Revealed. Security Focus, [Online]. Available: http://www.securityfocus.com/infocus/1684

[8] Cotse.net. (2004, November). [Online]. Available: http://www.cotse.com/tools/stega.htm

[9] B. Clair. (2001, October). Steganography, How to Send a Message, [Online]. Available: http://www.strangehorizons.com/2001/20011008/steganography.shtml

[10] Mr. Byte. (2004, November). Practical Privacy Guide: Steganography, [Online]. Available: http://www.all-nettools.com/library

[11] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. Paiz, and S. Pogreb. (2001). Applications for Data Hiding. IBM Systems Journal. [Online]. 39(3&4), pp. 547-568. Available: http://www.almaden.ibm.com/cs/people/dgruhl/afdh.pdf

[12] N. F. Johnson, Doric, Zoran, S. Jajodia. “On "Fingerprinting" Images for Recognition” Submitted to Fifth International Workshop on Multimedia Information Systems (MSIS'99) Palm springs, California, October 1999.

[13] N. F. Johnson, **Steganography**: Art & Science of Hidden Communication, Office of Naval Research (ONR) Naval-Industry Partnership Conference, Washington, DC, USA, August 2002.

[14] ThinkQuest Team 27158, "Basic Concept in Data Encryption, Classic Cryptography" http://library.thinkquest.org/27158/concept1_4.html, 1999

[15] Forrest, Stephen, Deogol web page, http://forrest.cx/projects/deogol/intro.html