Host Naming and URL Conventions

Pankaj Saharan
Helsinki University of Technology
psaharan@cc.hut.fi

Abstract

The most basic thing that needs to be considered by every organization before rolling out new servers or developing new web-based applications which would be accessed by customers and clients via Internet is to follow *host naming and URL referencing* conventions. There are no "silver bullets" when it comes to techniques used for securing web applications. In this paper, we will discuss the different attack vectors and techniques used by attackers to exploit the host naming and URL referencing techinques used for web applications by explaining different attack scenarios, suggest some best security practises to be followed and conclude about the practises to be followed for the effective solutions.

KEYWORDS: URL, DNS, Phishing, CSS (Cross Site Scripting), SQL Injection, Serial Host Naming.

1 Introduction

The way by which an organization names the public internet hosts and uses URL address references, is one of the simplest and easiest methods used by attackers to make a successful attack. The main reason behind an attacker's motivation for the attack is the lack of deployment of security best practices in the host naming and linking conventions, and lack of knowledge of latest attack vectors. The organizations generally don't utilize sufficient time for investigating the latest attack vectors and security practices followed before deploying the host naming and URL linking conventions.

There are many attack vectors used by malicious attackers to target an organization's customers or clients. The number and sophistication of the vectors used for carrying out the attacks are increasing year-on-year. But if we just limit the number of attacks carried out exploiting host naming and linking conventions, there are still a good range of attacks possible and are being carried out by attackers. The attacks range from social engineering to URL obfuscation [4] and domain hijacking [9]. The consequences of these attacks range from the loss of customers confidence from the online application, through to the manipulation or even actual compromise of the hosting environment.

The rest of the paper is organized in the following way. In Section 2, we analyze the different attacks possible due to weak host names and URL referencing conventions. In section 3, we define the different attack vectors and techniques used by attackers for conducting an attack. In section 5, we suggest the different securtiy practises to be used for securing web based applications using basic host naming and referencing techniques. In section 6, we summarize the information gained from this paper and suggest future implications for organizations to follow for host naming and URL referencing of their web based applications.

2 Understanding traditional threats

The first step that the organizations should follow, before planning to use the security practices to counterfeit the attacks, is to analyze the range of attacks resulting because of poorly thought out internet host names and URL referencing conventions.

2.1 Phishing

Phishing is a very common attack nowadays. Phishing attacks basically use social engineering schemes to steal customer's personal identity data and financial account credentials. These schemes mostly use spoofed emails to mislead customers to illegitimate websites hosted at an attacker's server and trick users to give personal and financial information such as credit card numbers, usernames, passwords, social security numbers etc. Their personal information and authentication credentials are recorded for later use in financial fraud or identity theft. The electronic message used can be in the form of an e-mail, web banner advertising or instant messaging. [10]

2.2 Mistyped names

The knowledge background of the users of a web application varies greatly. Many users, those even having good knowledge of web security practices, mistype host names and domain names. The attacker may use different permutations of an organization's domain and could register them for their own application. When the user mistypes the domain name, he is carried to the attacker's domain to an illegitimate application which can be used as a medium to capture user personal details and/or authentication details for financial fraud or identity theft, and also with the aim of discrediting the organization.

2.3 Cross-site scripting (CSS)

By CSS [12] attack, an attacker causes a legitimate web server to open a page in the user's web browser that consists of malicious java script or HTML of the intruder's choice.

The malicious script runs of the browser with the same privileges of a legitimate script originating from the legitimate web server. A user can easily loose his personal information, if he browses the Internet with scripting enabled.

2.4 Session Hijacking

As the name suggests, this attack results in the hijacking of the user session if an attacker is successful in capturing the session ID from a packet [6]. It can result in the unauthorized access to all resources available to the actual authenticated user. There are different kinds of techniques which can be used by attacker to acquire the session ID.

2.5 Bot-net building

The attacker's motivation of using a botnet [7] is not only to gain authentication and personal details of the users but also to install a remotely controllable agent. With the help of this agent running on the user's system, the attacker could monitor all the online activity of the user and capture the authentication details used by user for different online applications.

2.6 SQL Injection

SQL injection [8] is a technique used to take advantage of non-validated input vulnerabilities to pass SQL commands through a Web application for execution by a backend database. Attackers take advantage of the fact that programmers often chain together SQL commands with user-provided parameters, and can therefore embed SQL commands inside these parameters. The result is that the attacker can execute arbitrary SQL queries and/or commands on the backend database server through the Web application.

2.7 Example: One real attack scenario (Cross-site scripting)

On November 8th, 2006 Rajesh Sethumadhavan discovered a type 2 vulnerability in the social network site "Orkut" which would make it possible for the site's members to inject HTML and JavaScript into their profile. Rodrigo Lacerda used this vulnerability to create a cookie stealing script known as the "Orkut Cookie Exploit" which was injected into the site's profiles of the attacking member(s). By merely viewing these profiles unsuspecting targets had the communities they owned transferred to a fake account of the attacker. On December 12th, Orkut fixed the vulnerability. [2]

3 Attacker's toolkit

Every attacker has a finite set of techniques and vectors to select from when conducting an attack depending on his motivation. The most common and useful techniques for the attacker to abuse host names and linking conventions are:

Attack Technique	Example
Hyphenated Names	http://www.east-westernbank.com
Country Specific Registration	http://www.eastwesternbank.co.in
Legitimate Possibilities	http://www.secure-eastwesternbank.com
Mixed Wording	http://www.globaleastwesternbank.com
Long Host Names	http://www.globaleastwesternbank.com
Mixed-case Ambiguities	http://www.eastvvesternbank.com (notice double "v" in place of "w")
Common Misspellings	http://www.eastwesternbanks.com

Figure 1: Example naming scenarios of an example domain

3.1 Use of similar or identical Domain names

It is not a difficult task for an attacker to register a domain name through an international domain registrar. But if he is successful in registering a domain name which is similar to an organization's already registered domain, he can use for it for its evil motives ranging from discrediting the organization to misguiding the user to his own personal web application thereby getting the user personal and authentication details.

Here are some of the case scenarios of using similar domain names:

1. **Diversion to a different website**

   The attacker registers a very similar domain name to already registered organization's domain name and then diverts the traffic to its own website.

2. **Creating a mirror image website**

   The attacker registers a very similar domain name to organization's domain name and then maintains that site which is usually a mirror image of its own website i.e. has the same content as the original domain's website.

3. **Registration but No Use of a similarly named Website**

   The attacker registers a very similar domain name to organization's domain name and then sits on it i.e. there is no content at the relevant URL. This is usually done to discredit the organization.

4. **Using a Competitor's Product Name or URL as a Search Engine Adword**

   The attacker pays a search engine to bring up its website when someone making a search enters the target organization's name or a well-known product or brand name.

To understand various permutations of this attack vector, let us assume that the target organization's name is "East Western Bank" and their web application is "www.eastwesternbank.com". The common permutation techniques used are shown in Figure 1.

3.2 Complex and confusing URLs

[ref1] Organizations normally implement long and complex URLs in order to provide access to specific components of the online application to their users. This is due to poorly thought out application development processes and over dependency on third party integration tools. The first obvious

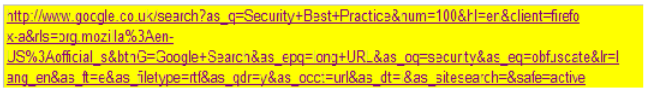

Figure 2: An example of a long URL

Figure 3: An example of SessionID information embedded in URL

result from this is that the long URLs are difficult to type and typos become quite common. Also, these URLs don't support usability i.e. user is not able to repeat and remember them easily. These weaknesses are utilized by the attacker and he can easily add malicious code in the URL address and is unlikely to be detected at the time of user clicking.

Figure 2 will show an URL which will not be shown the web-browser's address bar and serves as an excellent attack platform for an attacker.[11]
Some applications also contain Session ID information in their URLs which allows the user to carry out different varieties of attacks ranging from brute-force to preset session hijacking.

The following figure 3 [11] shows the Session ID information embedded into URL, and it can be inferred that the application makes use of "BroadVision" [1] content delivery platform.

Also, several third- party organizations are offering free services to organizations for coping with undesirably growing URL size and problem with transferring the URLs in email systems. The most popular web sites providing this functionality for free are *http://smallurl.com* and *http://tinyurl.com*. If the organizations use long and complex URLs, it becomes easy for the attacker to misguide customers by registering to a fake site using any one of the free short URL name service provider.

4 Suggested Practices

It is a well known fact that there are no "silver bullets" in information security. It is also been proved that the simpler the security practice followed, the more effective it is against the attacks. This goes the same for the security best practices to be followed for threats and attack vectors described in the previous section.

The most basic step that should be implemented by the organizations is to keep host names as short and simple as possible for the users to understand and the host names should be combined with short and simple URL references.

Use:	Instead of:
http://www.eastwesternbank.com/ebank	http://www.eastwesternbank-ebank.com
http://www.eastwesternbank.com/UK	http://www.eastwesternbankuk.com
https://www.secure.eastwesternbank.com	https://www.secure-eastwesternbank.com

Figure 4: Examples of how to use TLD effectively

Use:	Instead of:
http://www.eastwesternbank.com/UK	http://www.eastwesternbank.co.uk
https://www.secure.eastwesternbank.com/UK	https://www.secure.eastwesternbank.com.uk
http://www.eastwesternbank.com/ZA/New-user	http://www-eastwesternbank-new-user.za

Figure 5: Examples of Redirecting Regional Domains

4.1 Domain Names and host services

All organizations with online web content have a registered domain name and possibly own a number of similar domain names. The domain names generally identify the organization's main service or business, or particular application information.

The problem generally arises because the users using the domain names of the organizations have different knowledge backgrounds. It is generally very difficult for non-technical users to follow and remember complex host names and URLs. So these basic practices should be used for domain naming and host referencing:

1. **Using same Top Level Domain(TLD)**

   The organizations should same TLD for all online web applications. This becomes necessary because of the fact that the users generally hesitate to trust the custom domains and loosely associated URLs with the TLD. So it is important that the users should gain access to organization's services through well known and trusted domain.

   Here are some examples to show how to use TLD effectively for different service as shown in figure 4

2. **Redirecting Regional Domains**

   It should be noted by the organizations spread internationally that host names should identify the country specific application or service offered and the user should easily identify the reference from the root domain (TLD). This process not only secures the application but also enable the organization to do global load balancing.

   The following table in figure 5 shows some best practice examples:

3. **Keep It Simple (KIS)**

   "Keep it simple" (KIS) formula should be used while naming hosts and URL combinations. Adequate time and effort is required to think of the naming conventions to be used before rolling out the new servers for online applications and it should be noted that the selected host name should reflect key aspects of the service.

   The following table in figure 6 provides some best practice examples:

Use:	Instead of:
https://www.secure.eastwesternbank.com/investor	https://www.eastwesternbank.com/secureinvestor
http://www.news.eastwesternbank.com/UK	http://www.eastwesternbank.co.uk/onlinebanking/news
https://www.secure.eastwesternbank.com/ZA/personal	https://personal.eastwesternbankinvestor.co.za/secure/elogin

Figure 6: Examples of KIS techinque

Use:	Instead of:
https://secure.eastwesternbank.com	https://www.eastwesternbank.com
http://invest.eastwesternbank.com/UK	http://www.InvestorAteastwesternbankuk.com
http://www.eastwesternbank.com/invest	http://investment.eastwesternbank.com

Figure 7: Examples of using Representive Naming

4. **Using Representative Naming**

   Additionally, if the service uses secure services e.g. SSL-based HTTPS, the recommended practice is to embed the protection service used in the host name.

   The following table in figure 7 shows some best practice examples:

5. **Avoid Host numbering**

   The organizations should not let the host numbers visible on the internet. This not only creates confusion and increases complexity for the user, but also can be used by attackers for planning an attack by discovering insecure hosts. For example, "www.eastwesternbank.com" should be used in place of "www3.eastwesternbank.com".

4.2 URL Referencing

Not only for host names, the organizations should also take care while handling URLs used for the web based application. URLs are used for accessing navigate and access specific application process or functionality and organizations should carefully review the URLs used. If the complexity of URLs is reduced, it also reduces the chances of an effective attack.

Best security practices for URL referencing include:

1. **Use Small URLs**

   Organizations should make sure that they don't overload the use of HTTP GET requests in the URLs and should be replaced by HTTP POST requests. This helps in reducing the complexity of the URLs as well as makes it difficult for the attacker to carry out many attacks. [5]

   Organizations should only use URL's to direct customers to key application components or services ideally binding all environment details to their unique SessionID (e.g. customer identity, application preferences, etc.). All other details and data submissions can be handled through the use of HTTP POST requests through HTML forms. A combination of HTML forms and client-side scripting can enable any possible request that would traditionally be managed through HTTP GET requests alone. [11] The application developers should concentrate to implement session management solution

```
http://www.mybank.com/ebanking/transfers/doit.aspx?funds=34000&agent=kelly02&sessionid
=898939289834
```

Figure 8: HTTP GET request

```
<FORM METHOD=POST ACTION="ebanking/transfers/doit.aspx">
<INPUT TYPE="hidden" NAME="funds" VALUE="34000">
<INPUT TYPE="hidden" NAME="agent" VALUE="kelly02">
<INPUT TYPE="submit" NAME="Transfer">
```

Figure 9: POST method for making application requests

   for the web applications rather than focusing on individual page references. So in this way, the URLs could become more memorable to the user, transportable to different systems e.g. using e-mails and it would be easy for the customers to detect any attacks.

   But its worthy to note that using HTTP POST requests is not a full-proof solution. An attacker, for example, can use personal proxies or hacking tools to manipulate the data sent via POST method.

2. **Remove Session Information from URLs**

   URLs of a web application should never contain session related information. Every web session between the user and the application server should have a specific Session ID. This Session ID related information should be stored in the cookies of the user's web browser. [13]

   [11] For example, instead of using the following HTTP GET request as shown in figure 8:

   Application developers should use the more secure and robust HTTP POST method to make application requests. For example, the code behind the customers page request may look like in the following figure 9

   Which could result in the following HTTP POST from the client browser as shown in the figure 10

3. **Remove application variable from URLs**

   Just like session information, application variables should not be visible with the URLs. The application variables should be handled on the server side associated with the Session ID of the user and if not, should be handled through hidden HTML form submissions.

4.3 Serial Host Naming

Many organizations use serial host naming [11] for serial naming the host servers. Attackers can use the concept of serial hosts by discovering forgotten or insecure hosts. Attackers can use these hosts' names to connect directly to the server and attempt to compromise their weaknesses for carrying out an attack. The reason being that when organizations implement serial host naming and name their load-balanced hosts typically through a well known host name or URL, some of the hosts would not be configured as well as others, which are later exploited by attackers.

```
POST /ebanking/transfers/doit.aspx HTTP/1.1
Referer: http://www.mybank.com/ebanking/transfers/balance
Accept Language: en gb
Content-Type: application/x-www-form-urlencoded
Proxy-Connection: Keep-Alive
User Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: www.mybank.com
Content Length: 46
Pragma: no cache
Cookie: sessionid=898939289834

funds=34000&agent=kelly02&action=submit
```

Figure 10: HTTP Post Method

For example, an attacker notices that there are two public servers called *grumpy.eastwesternbank.com*and *sleepy.eastwesternbank.com*. So there would not be much problem for the attacker to try other seven dwarf names to carry out his attack.

A three phased defensive practice is recommended [11]:

- Use unrealistic host names, try to name then using some logic just like passwords, instead of using serialized names (as shown in above example of naming using seven-dwarfs' names).
- Do not provide Internet accessible forward or reverse DNS entries for hosts that do not actually require named access over the Internet. Access to these hosts can be governed through appropriate load balancing technologies and address translation.
- Manage the authoritative DNS servers correctly to ensure that only authorised hosts appear within the public DNS entries and that the DNS server itself is correctly configured to disallow zone transfers.

4.4 Domain Registering Monitor

There are many commercial domain monitoring services available for the organizations nowadays. These services provide monitoring of the organization's domain for safety, notify when similar domains are registered, notify if any changes are made to the domain instantly, watch other similar domains that are expired, notify when a domain name is released. There are also commercial services that keep track of hacking forums and discussions of attacks related to host naming and URL conventions. It is strongly suggested that organizations should register to such third-party agencies for domain safety. [3]

5 Conclusion

With the help of this paper, we analyzed the threats posed to the organizations if they fail to follow the best security practices for host naming and URL referencing conventions. On the basis of the attacks discussed, we can easily infer the importance of following host naming and URL conventions in the web based applications of organizations. The attacker's motivations are the same but they are gaining additional methods in their tool kit of attacks. So as shown in this paper, the best way to secure the web based applications is to follow the simple best security practices. The simpler the security practice followed, the more effective it is against the attacks.

The most basic step that should be implemented by the organizations is to keep host names as short and simple as possible for the users to understand and the host names should be combined with short and simple URL references. By adopting these security practices discussed and some common sense strategies, the organization could definitely prevent the future attacks and could definitely increase their customers' confidence in their online service offerings.

References

[1] Broadvision. http://searchcio.techtarget.com/sDefinition/0,,sid19_gci214592,00.html.

[2] Cross-site scripting. http://en.wikipedia.org/wiki/Cross-site_scripting.

[3] Domain monitoring service. http://registration.premierwebsitesolutions.com/.

[4] How to obscure any url. http://www.pc-help.org/obscure.htm.

[5] Methods get and post in html forms - what's the difference? http://www.cs.tut.fi/~jkorpela/forms/methods.html.

[6] Session hijacking. http://www.imperva.com/application_defense_center/glossary/session_hijacking.html.

[7] D. Dittrich. Five steps for beating back the bots. http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci1068834,00.html.

[8] S. Friedl. Sql injection attacks by example. http://ocliteracy.com/techtips/sql-injection.html.

[9] R. from ICANN's Security and S. A. C. (SSAC). Domain name hijacking: Incidents, Threats, Risks, and Remedial Actions. Technical report, July 2005. http://www.icann.org/announcements/hijacking-report-12jul05.pdf.

[10] G. Ollmann. The phishing guide. http://www.nextgenss.com/papers/NISR-WP-Phishing.pdf.

[11] G. Ollmann. Security best pratise: Host naming and url conventions. http://www.ngssoftware.com/papers/NISR-BestPracticesInHostURLNaming.pdf.

[12] J. Rafail. Cross-site scripting vulnerabilities. www.cert.org/archive/pdf/cross_site_scripting.pdf.

[13] P. J. Windley. Cookies and Privacy. Technical report. http://www.windley.com/docs/Cookies%20and%20Privacy.pdf.