YGN ETHICAL HACKER GROUP

Whitepaper

MULTIPLE TRICKY WAYS TO PROTECT SENSITIVE FILES & DIRECTORIES OF YOUR CRITICAL WEB APPLICATIONS

By

d0ubl3_h3lix

Tue Jan 29 2008

Abstract

Nowadays almost all kinds of full-featured web applications such in CMSes (Content Management Systems) include administrative interface where we can administer dynamic features provided. While the admin feature provides us much convenience, it surely attracts attackers like delicious honey for hungry bees. Typically, developers do not always make detailed attention to make every admin area secure because the user is an admin who is trusted and reliable. Over times, the more web applications are complicated, the more hidden security holes have to be existed. Hence, to protect this admin feature as much as we can is a must for critical applications. In this paper, I will show you multiple protection ways which may not be foolproof but may hinder attackers to a certain extent.

Assumption

I assume your web servers and underlying hosting infrastructure have been properly patched and secure as attackers can own your applications ultimately through even a single hole in each OSI layer. I assume the admin interface of the web application you are using has some hidden flaws. Since you are not a core developer of that third-party application, you may not know how to fix them up. In this case, your CMS is at risk of being attacked any time. You are not in helpless position. You have my suggested methods for protection.

Protection Methods

Through Fixed IPs

If you always use or manage your web applications from single IP (maybe your company IP), this method adds some protection. Restricting IPs with server-side methods to your admin files and folders force attackers to try to know admin's IP address if they are smart. An average attacker cannot even determine why they are banned. Nevertheless, this method is the lowest level protection mode if you deal with a lot of clients. Smart attackers may trick you to reveal your IP in some ways.

Through Obscurity

All ready-made and open-source web applications have default administrative file names or folder names. If you change those to non-dictionary and non-hacker-speaker words such as (admin to xd034a23e22G893xd44R4 not to 4dm1n), this may make attackers assume you might have removed or renamed admin files or folders. Using

particularly strange words in renaming sensitive files and folders protects brute force attacks using dictionaries or predefined wordlist mutations.

Through Weird Parameters

What would you do if unluckily attackers manage to know your admin files and folders through intensive brute force or a bug in applications? Well, the next defense is to set weird parameters to access like:

```
admin.php?iownu=03e4343x223232&usuck=4332343oI234i23&g9=34324xcxcx2
```

This forces attackers to brute force again even for accessing admin pages and folders. Again, they have fewer chance of owning your web application via admin interface.
If someone accesses your admin stuffs without sending those weird parameters, log his access attempts, deny his IPs.

Through Further Authentication Methods

Now smart attackers have successfully bypassed all the above restrictions. They now can see your admin login page. In this case, their to-do-lists are likely to be:

- Generating blind requests to what files and folders are existing
- Cracking admin login using brute force attack
- Cracking admin login using injection attempts
- Deliberately submit bad data to the application to show useful error messages

Must-do countermeasures are:

- Set a little complicated capcha image to protect brute forcing
- Set another secret question like ‘Why 25 June is great for me?’ in login method. If someone tries to be friendly with you and asks the above question through Social Engineering attack, you know he is the attacker actually breaking your application.
- Implement strong input filtering
- Suppress any kind of application debugging error messages