

THE ACUNETIX BLOG > WEB SECURITY ZONE

Windows Short (8.3) Filenames – A Security Nightmare?



Bogdan Calin | July 3, 2012

Each time you create a new file on Windows, the operating system also generates an MS-DOS-compatible short file name in 8.3 format, to allow MS-DOS-based or 16-bit Windows-based programs to access files which have a long name. You can see these MS-DOS-compatible short file names by using the **/X** switch with the **dir** command. On my system I get something like this:

```
3,987 BACKUP~1.PNG backup-options-saveserver.png
6,558 DIRECT~1.PNG directory-name.png
4,648 FILE-N~1.PNG file-name.png
122,230 VULNER~1.PNG vulnerability.png
845 WINDOW~1.TXT windows-short-names.txt
138,268 bytes
```

There have been a lot of security problems in the past related to short file names. Just yesterday, I found another paper that talks about this subject. The paper was written by Soroush Dalili and is called [Microsoft IIS tilde character “~” Vulnerability/Feature – Short File/Folder Name Disclosure](#).

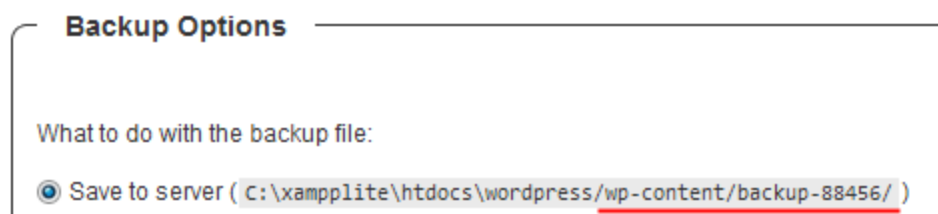
When using IIS, you can get a list of all the shortnames (both files and directories) from a certain directory. This can be a big problem if you can manage to guess, or bruteforce the full file or directory names from these short names. While working on a security script for [Acunetix Web Vulnerability Scanner](#), I thought “Why you have to guess the full names once you have the short names? Why you cannot use the short names? It turns out that IIS doesn’t accept short names for a variety of reasons,

Think of following scenario; a web application running on Apache on Windows, is creating a file with a long name that should not be guessed by an attacker. For example it creates a session file or an SQL backup file. In this case the security of this application relies on the fact that the name of this file cannot be guessed.

Let's assume that this file name is **backup-082119f75623eb7abd7bf357698ff66c.sql**. Windows will create a short name for this file, **BACKUP~1.SQL**. If I can access this file using the short file name then all the security is broken. I just request BACKUP~1.SQL and get the file, which includes a backup of an SQL database.

Being curious if this problem is a real life problem, I looked at two of the most popular backup plugins for WordPress. Both of them are affected by this problem, which is explained in detail below.

After installing one of the plugins, I have requested a backup of my WordPress blog:



The plugin creates a custom directory for this backup (**backup-88456**). Once the backup is completed, the directory contained a file named **wordpress_wp_20120702_576.sql**, which is the WordPress database backup. This should be pretty hard to guess. We have 5 numbers in the directory name (100 000 combinations) plus the date and plus **3** more numbers. In total it should be at least **100,000,000** combinations if we ignore the date. What do you think are the short names for this directory and file? Using short names this is pretty easy to guess.

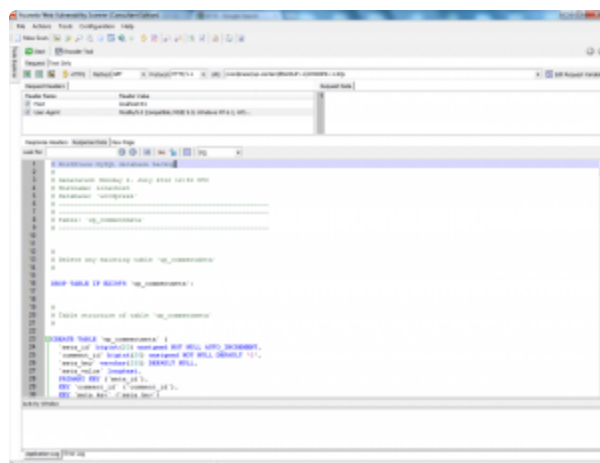
```
C:\xampplite\htdocs\wordpress\wp-content>dir /X *
Volume in drive C has no label.
Volume Serial Number is 5-440-5777

Directory of C:\xampplite\htdocs\wordpress\wp-content

07/02/2012  03:40 PM    <DIR>          .
07/02/2012  03:40 PM    <DIR>          ..
07/02/2012  03:50 PM    <DIR>          BACKUP~1      backup-88456
01/06/2012  05:00 PM    <DIR>          index.php
07/02/2012  03:47 PM    <DIR>          plugins
07/02/2012  03:43 PM    <DIR>          themes
07/02/2012  03:47 PM    <DIR>          upgrade
               1 File(s)              20 bytes
               6 Dir(s)  388,385,599,400 bytes free
```

```
Directory of C:\xampplite\htdocs\wordpress\wp-content\backup-88456

07/02/2012  03:50 PM    <DIR>          .
07/02/2012  03:50 PM    <DIR>          ..
07/02/2012  03:40 PM    <DIR>          0
07/02/2012  03:50 PM    <DIR>          index.php
07/02/2012  03:50 PM    <DIR>          320,440 WORDPRESS~1.SQL wordpress_wp_20120702_576.sql
               2 File(s)             320,440 bytes
               2 Dir(s)  388,385,599,400 bytes free
```



What can you do to protect yourself against this problem, and who's fault is it?

Is Microsoft's fault that they still support the short names in 2012? Maybe. I'm not sure but legacy and security do not go well together. Or is it Apache's fault that they support the short names? Maybe. I don't think it is the fault of the person who wrote the WordPress plugin.

The solution

There is a way to disable Windows 8.3 short name creation. You can create a registry key named **NtfsDisable8dot3NameCreation** in HKLM\SYSTEM\CurrentControlSet\Control\FileSystem and set it to 1. That should disable short names creation. Refer to this [Microsoft TechNet article](#) to read more about the solution.

Acunetix

Get the latest content on web security
in your inbox each week.

Subscribe

We respect your [privacy](#)

SHARE THIS POST



THE AUTHOR



Bogdan Calin

Acunetix developers and tech agents regularly contribute to the blog. All the Acunetix developers come with years of experience in the web security sphere.

Related Posts:



Four ways to combat the cybersecurity skills gap

[Read more →](#)



Common password vulnerabilities and how to avoid them

[Read more →](#)



How Acunetix addresses HTTP/2 vulnerabilities

[Read more →](#)

Acunetix
by Invicti

```

graph LR
    subgraph Attacker_Botnet [Attacker's Botnet]
        direction TB
        AB[Attacker's Botnet]
        AS[Attacker's Server]
    end

    subgraph Website [Website]
        direction TB
        WB[Website Browser]
        WVC[Website's Vulnerable Code]
    end

    AB -- "1 GET http://www.example.com/index.html" --> WVC
    WVC -- "2 GET http://www.example.com/index.html" --> WB
    WB -- "3 GET http://www.example.com/index.html" --> WVC
    AS -- "4 GET http://www.example.com/index.html" --> WVC
    WVC -- "5 GET http://www.example.com/index.html" --> AS
  
```

The diagram illustrates a Denial of Service (DoS) attack on a website using a botnet. The components and flow are as follows:

- Attacker's Botnet:** Contains the **Attacker's Botnet** and the **Attacker's Server**.
- Website:** Contains the **Website Browser** and the **Website's Vulnerable Code**.
- Flow of the Attack:**
 - The **Attacker's Botnet** sends a **GET http://www.example.com/index.html** request to the **Website's Vulnerable Code**.
 - The **Website Browser** sends a **GET http://www.example.com/index.html** request to the **Website's Vulnerable Code**.
 - The **Website's Vulnerable Code** sends a **GET http://www.example.com/index.html** request back to the **Website Browser**.
 - The **Attacker's Server** sends a **GET http://www.example.com/index.html** request to the **Website's Vulnerable Code**.
 - The **Website's Vulnerable Code** sends a **GET http://www.example.com/index.html** request back to the **Attacker's Server**.

The diagram shows that the **Website's Vulnerable Code** is overwhelmed by the requests, leading to a **Denial of Service (DoS)**.

Acunetix
by Invicti

[← Older](#)

Newer →

Subscribe

Vulnerability Assessment vs Pen Testing

Blog Categories

- Articles
- Web Security Zone
- News
- Events
- Product Releases
- Product Articles

PRODUCT INFORMATION

- AcuSensor Technology
- AcuMonitor Technology
- Acunetix Integrations
- Vulnerability Scanner
- Support Plans

LEARN MORE

- White Papers
- TLS Security
- WordPress Security
- Web Service Security
- Prevent SQL Injection

USE CASES

- Penetration Testing Software
- Website Security Scanner
- External Vulnerability Scanner
- Web Application Security
- Vulnerability Management Software

COMPANY

- About Us
- Customers
- Become a Partner
- Careers
- Contact

WEBSITE SECURITY

- Cross-site Scripting
- SQL Injection
- Reflected XSS
- CSRF Attacks
- Directory Traversal

DOCUMENTATION

- Case Studies
- Support
- Videos
- Vulnerability Index
- Webinars

Terms of Use

Sitemap



© Acunetix 2024, by Invicti