

# **EVIL COLON NAPADI 101**

Leon Juranic





# EVIL COLON 101

OTKRIO SAM IH TOKOM SOURCE CODE AUDITA JEDNE JAVA APLIKACIJE

MANIPULACIJA STAZA NA DISKU - NAPADI SLIČNI POISON NULL BYTE  
NAPADIMA

NULL BYTE NAPADI ISKORIJENJENI IZ SVIH POPULARNIJIH PROGRAMSKIH  
JEZIKA U ZADNJIH 10 GODINA

EVIL COLON NAPADI - MICROSOFT WINDOWS (NTFS FILESYSTEM)

SLUŽE ZA MANIPULACIJU STAZA NA DISKU TE KREIRANJA ARBITRARY  
FAJLOVA I NJIHOVIH EKSTENZIJA



# POISON NULL BYTE -> 0x00 -> \0

NAPADI KOD KREIRANJA STAZE NA DISKU

`"/var/www/htdocs/" + userInput + ".jpg"`

`userinput -> imefajla.jsp\0`

KOD OBRADJE OVAKVE STAZE NA NIVOU JEZIKA (KOJI JE OBIČNO NAPISAN U C-U) ILI API-a ISPOD HAUBE - NULL BYTE (\0) OZNAČAVA STRING TERMINATOR ODNOSNO KRAJ STRINGA

`"/var/www/htdocs/" + "imefajla.jsp\0" + ".jpg"`

DODAVANJE .JPG EKSTENZIJE JE ZANEMARENO JER NULL BYTE OZNAČAVA KRAJ STRINGA KOD ZAPISIVANJA NA FILESYSTEM

`"/var/www/htdocs/imefajla.jsp"`

KREIRANJE FAJLOVA NA DISKU SA ARBITRARY EKSTENZIJAMA (WEBSHELL, itd.)



# EVIL COLON

RANJIVE SU APLIKACIJE DEPLOYANE NA MICROSOFT WINDOWS (NTFS)

ISKORIŠTAVANJE ADS (ALTERNATE DATA STREAM) FUNKCIONALNOSTI ZA MANIPULACIJU STAZAMA NA DISKU SA NTFS FILESYSTEMOM

*"An Alternate Data Stream is a little-known feature of the NTFS file system. It has the ability of forking data into an existing file without changing its file size or functionality. Think of ADS as a 'file inside another file'."*

ADS SE OBIČNO KORISTIO I ZA SKRIVANJE MALWARE-A

ZAŠTO EVIL COLON NAPAD - ZBOG ZNAKA COLON ODNOSNO DVOTOČKE ':' KOJA JE KLJUČNA ZA ADS

MOGUĆE ISKORIŠTAVANJE SIGURNOSNIH RANJIVOSTI I ZAOBILAŽENJE SANITIZACIJE USER INPUTA U SVRHU MANIPULACIJE FAJLOVIMA NA DISKU TE KREIRANJU ARBITRARY FAJLOVA, WEBSHELL-OVA, ITD.

# ALTERNATE DATA STREAM (ADS)

Command Prompt

```
C:\Users\leon\ADS>echo TESTADS > imefajla.jsp:ADSIIME.JPG
```

```
C:\Users\leon\ADS>dir
Volume in drive C is OS
```

Directory of C:\Users\leon\ADS

```
09/20/2023  03:25 PM    <DIR>        .
09/20/2023  03:19 PM    <DIR>        ..
09/20/2023  03:25 PM                0 imefajla.jsp
                                0 bytes
1 File(s)
```

```
C:\Users\leon\ADS>dir /R
```

Volume in drive C is OS

Directory of C:\Users\leon\ADS

```
09/20/2023  03:25 PM    <DIR>        .
09/20/2023  03:19 PM    <DIR>        ..
09/20/2023  03:25 PM                0 imefajla.jsp
                                10 imefajla.jsp:ADSIIME.JPG:$DATA
1 File(s)                                0 bytes
```

# EXAMPLE- WriteFile.jsp

<%

```
// Create file on filesystem, vulnerable to Evil-Colon to cut off appending of .txt extension to file and create arbitrary extensions of filesystem
if (request.getParameter("filename") != null && request.getParameter("data") != null)
{
    String append_string = ".txt";
    FileOutputStream fout=new FileOutputStream("c:\\testing\\webdata\\" + request.getParameter("filename") + append_string );
    String s = request.getParameter("data");
    byte b[]=s.getBytes();
    fout.write(b);
    fout.close();
}

File f = new File("c:\\testing\\webdata");
String[] pathnames = f.list();

Integer x=0;
for (String pathname : pathnames)
{
    x++;
    response.getOutputStream().write(("<a href='WriteFile.jsp?appendID=" + x + "&data=TEST'> name: " + pathname + " - id: " + x.toString() + "</a>
    <br>").getBytes());
    response.getOutputStream().flush();
}

// Additional modification of previously created files, if Evil-Colon is used on created file, it is easy to modify its content
x = 0;
for (String pathname : pathnames)
{
    x++;
    if (request.getParameter("appendID").equals(x.toString()))
    {
        response.getOutputStream().write(("<br><br>Appending to file name: " + pathname + " - id: " + x.toString() + "</a>").getBytes());
        response.getOutputStream().flush();
        FileOutputStream fout=new FileOutputStream("c:\\testing\\webdata\\" + pathname);
        String s = request.getParameter("data");
        byte b[]=s.getBytes();
        fout.write(b);
        fout.close();
    }
}
```



# EVIL COLON & WriteFile.jsp

SKRIPTA KREIRA FAJL NA TEMELJU USER INPUTA U DIREKTORIJU `c:\testing\webdata\`

SANITIZACIJA USER INPUTA -> **.txt** EKSTENZIJA SE DODAJE NA SVAKI KREIRANI FAJL, ŠTO ONEMOGUĆAVA KLASIČNO ISKORIŠTAVANJE RANJIVOSTI KREIRANJA ARBITRARY EKSTENZIJE

EVIL COLON NAPAD -> NA KRAJ USER INPUTA SE DODAJE COLON ZNAK -> : TE APLIKACIJA ONDA .TXT EKSTENZIJU SPREMA U NTFS ADS, A NA DISKU SE KREIRA FAJL SA ARBITRARY EKSTENZIJOM .JSP -> **`"c:\testing\webdata\" + "evil.jsp:" + ".txt"`**

`http://localhost:8080/examples/WriteFile.jsp?filename=evil.jsp:&data=TEST_31337`





# EVIL COLON DOMINATION

UKOLIKO RANJIVA APLIKACIJA IMA FUNKCIONALNOSTI NAKNADNE MANIPULACIJE NAD SADRŽAJEM KREIRANIH FAJLOVIMA MOGUĆE JE KOMPLETNO ISKORISTITI EVIL COLON NAPAD (RECIMO ČESTO CMS-OVI, ITD.)

OBIČNO APLIKACIJE RADE DALEKO MANJE SANITIZACIJA NAD FAJLOVIMA KOJI SU VEĆ KREIRANI NA FILESYSTEMU

VEĆ SPOMENUTO KREIRANJE WEBSHELL-A







# EVIL COLON COVERAGE I ZAŠTITA

ISPROBAO SAM I VERIFICIRAO EVIL COLON NAPADE NA ZADNJIM VERZIJAMA VELIKOG BROJA MODERNIH JEZIKA (NPR. JAVA, PHP, PYTHON, ITD.)

EVIL COLON NAPADI FUNKCIONIRAJU U SVIM JEZICIMA KOJE SAM ISPROBAO (OSIM ASP.NET-a)

ZAŠTITA APLIKACIJA - FILTRIRATI ODNOSNO SANITIZIRATI COLON ZNAK, DVOTOČKU IZ BILO KAKVOG KREIRANJA STAZE NA DISKU SA USER INPUTOM DA BI SPRIJEČILI ZLOUPOTREBU



**HVALA NA PAŽNJI!**

PITANJA?