

RCA - CVE-2017-12561

Analysis through binary diffing using Ghidra. The idea is trying to locate where the bug was located by diffing the patched version with an earlier version of the software.

iMC PLAT 7.3E0504 and iMC PLAT 7.3E0504P04

Introduction

The structure of this document relies to some extent on:

<https://raw.githubusercontent.com/googleprojectzero/0days-in-the-wild/main/0day-RCAs/template.md>

The Basics

Disclosure

- 2017-06-27 - Vulnerability reported to vendor
- 2017-10-03 - Coordinated public release of advisory

Product:

Hewlett Packard Enterprise Intelligent Management Center

Affected Versions:

HPE IMC v7.3 (E0504)

Reporter:

Steven Seeley (mr_me) of Offensive Security

The Vulnerability

Bug class:

RCE(Remote Code Execution) through UAF(Use-After-Free)

Vulnerability details:

<https://securitytracker.com/id/1039495>

A remote authenticated user can send specially crafted data to the dbman service on TCP port 2810 to trigger a use-after-free in the mibFileServlet servlet to delete arbitrary files on the target system [CVE-2017-12561].

This resource seems to require a user to be authenticated to trigger the vulnerability.

<https://www.zerodayinitiative.com/advisories/ZDI-17-836/>

This resource states that authentication is not required to exploit this vulnerability.

From my tinkering so far, it seems that the authentication is not required. But default firewall rules on my test environment(WinServer2012R2) prevent reaching dbman port 2810 i.e., prevent sending payloads to it.

Analysis - Process

Started osinting to find out what is out there about this vulnerability CVE-2017-12561.

There are a number of vulnerabilities related to this one.

OSINT

[ZDI-17-836 / CVE-12561]

- A bunch of links for vulnerabilities revolving around this one,
 - Querying some of this sites gives an idea:
 - exploitdb
 - cvedetails
 - cvesearch
 - https://vulners.com/nessus/HP_INTELLIGENT_MANAGEMENT_CENTER_7_3_E0504P04.NASL
 - https://cxsecurity.com/cveproduct/7/18987/intelligent_management_center/

Structure of the analysis - technical

Here is the high level view of how I am approaching this stuff(excluding preliminary reading/analysis/osinting):

0. Setup the hosting environment

1. Setup the software (Hewlett Packard Enterprise Intelligent Management Center)
2. Try to confirm the bug
3. Refine previous step

Environment Setup - Installation

The following are the steps that I've taken to deploy the iMC software.

Host

My host is:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.3 LTS
Release:        20.04
Codename:       focal
```

I am using a type 2 hypervisor: Oracle VirtualBox, version/package:

```
$ dpkg-query -W virtualbox
virtualbox 6.1.26-dfsg-3~ubuntu1.20.04.
```

This is the version from Canonical's repo, not directly from Oracle.

Test Environment

I am using:

- **OS:** MS Windows Sever 2012 R2
- **Additional software for DB:** DotNet 4.5.2
- **Software DB:** SQLServer2014Express
- **Software:** iMC_PLAT_7.3_E0504_Ent_Win-003(vuln)
- **Software:** iMC_PLAT_7.3_E0504P04_Win-002(patched)

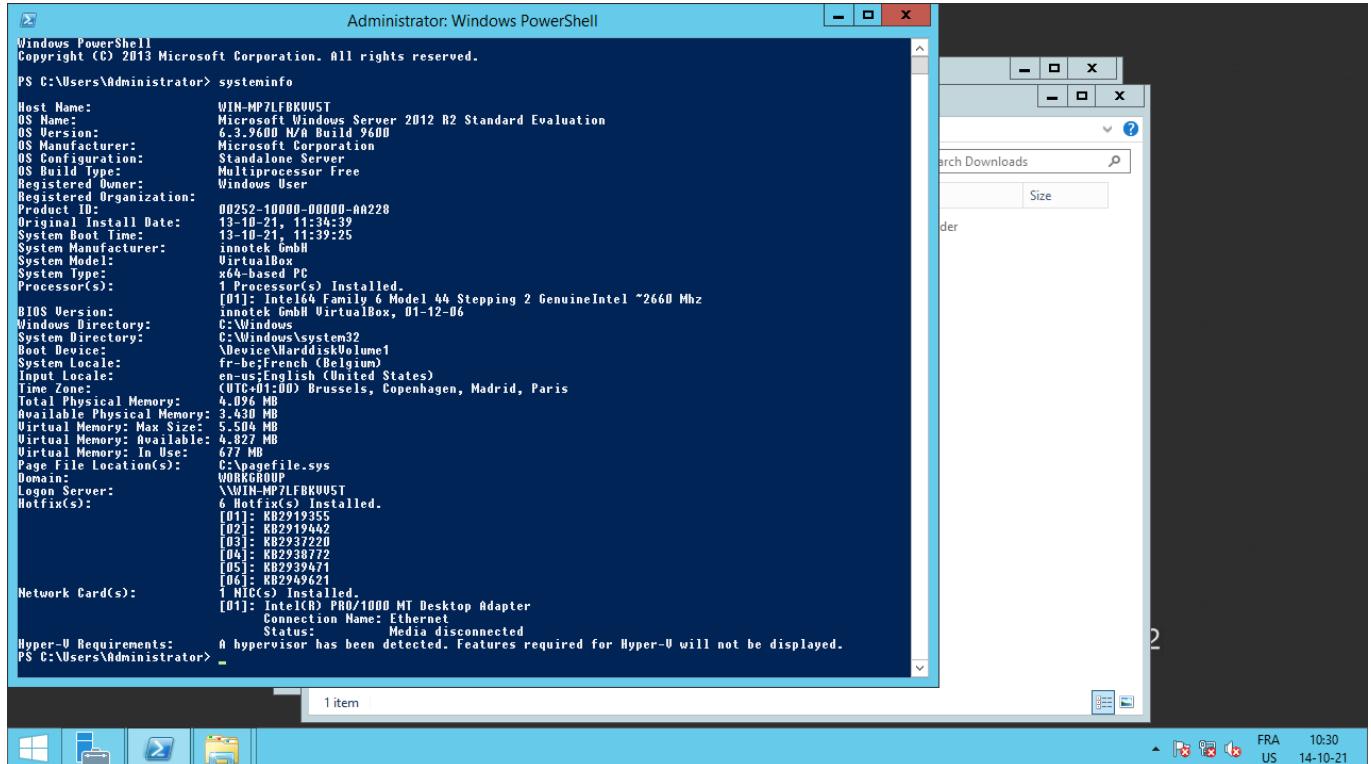
After deeper analysis, the version E0504 is affected by a stack-based buffer overflow or stack buffer overrun in msecdbg nomenclature. Whereas version E0504P04 introduces a potential remote code execution(RCE) in a specific function.

OS:

MS Windows Sever 2012 R2:

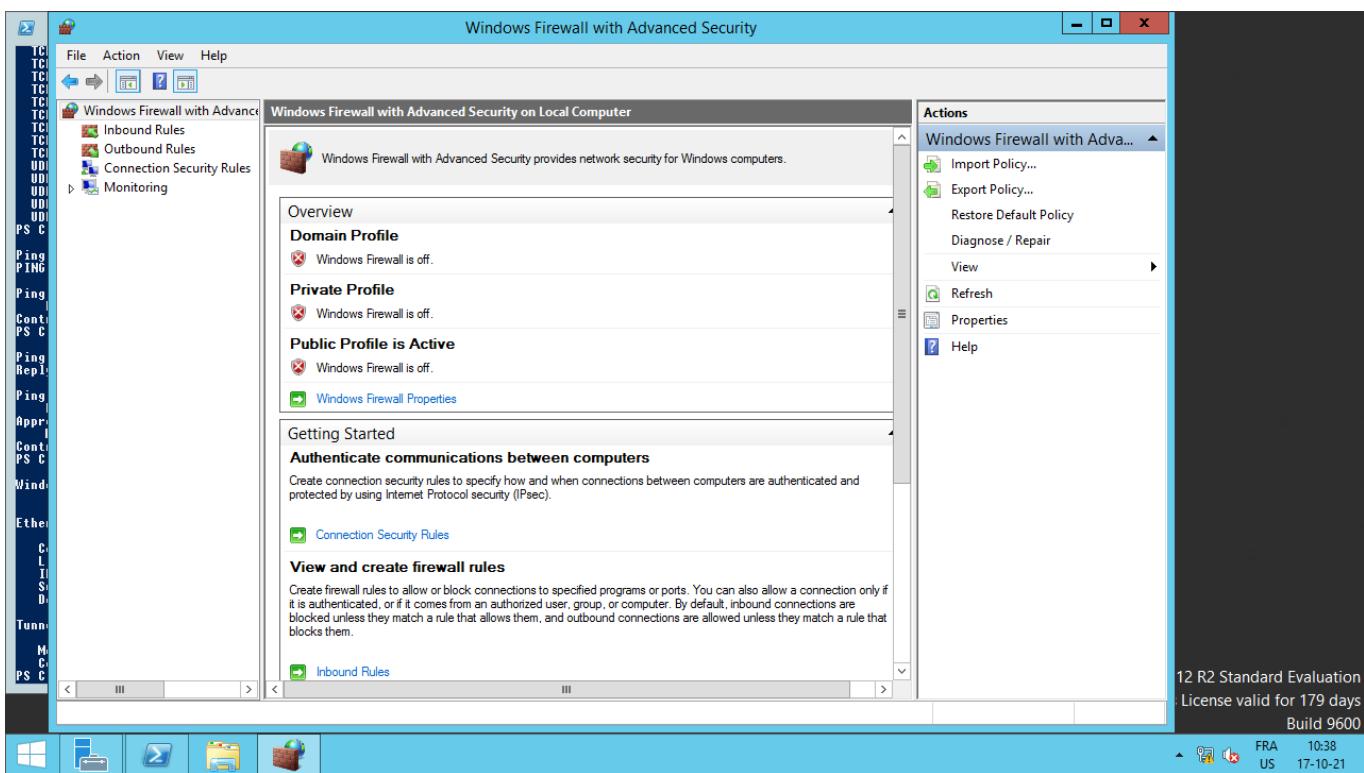
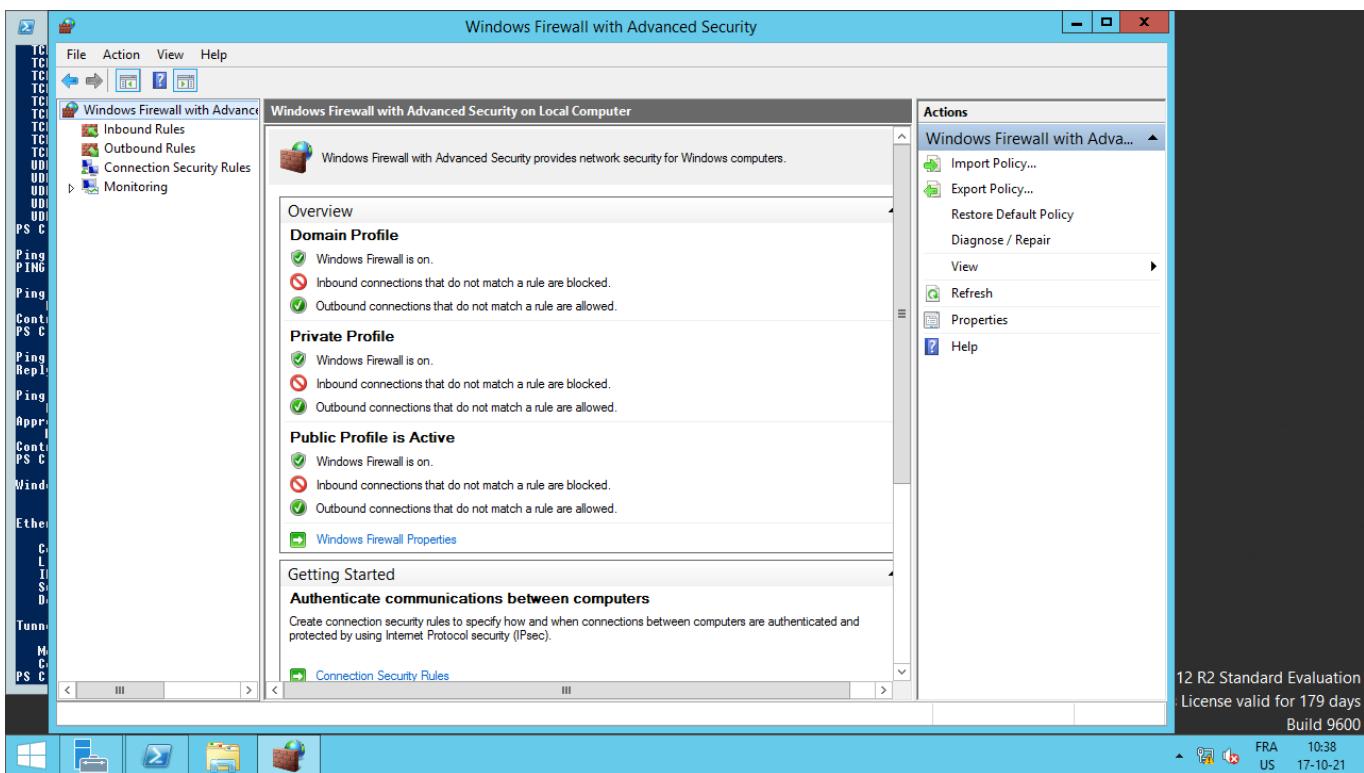
```
$ shasum 9600.17050.WINBLUE_REFRESH-140317-1640_X64FRE_SERVER_EVAL_EN-US-IR3_SSS_X64FREE_EN-US_DV9.ISO
849734f37346385dac2c101e4aacba4626bb141c  9600.17050.WINBLUE_REFRESH-140317-1640_X64FRE_SERVER_EVAL_EN-US-IR3_SSS_X64FREE_EN-US_DV9.ISO
```

The installation process is all standard, nothing fancy. I just avoided enabling Internet connectivity for old/"legacy" systems. One exception to this was during the installation of SQLServer2014.



[systeminfo]

By default the firewall is enabled on the three profiles: domain, private, public. If we want to reach *dbman* we need to disable the firewall or add a specific rule for it.



All red, we're good to go.

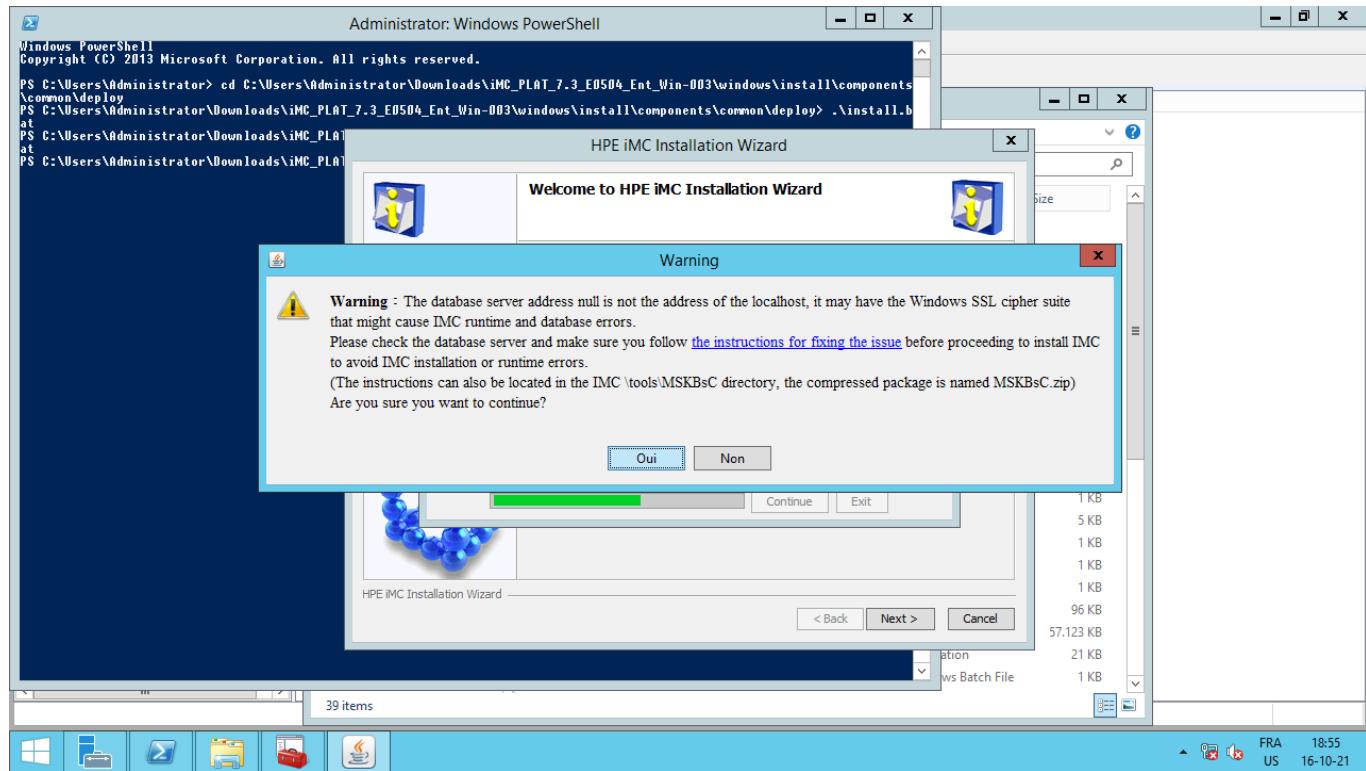
Software:

DotNet 4.5.2:

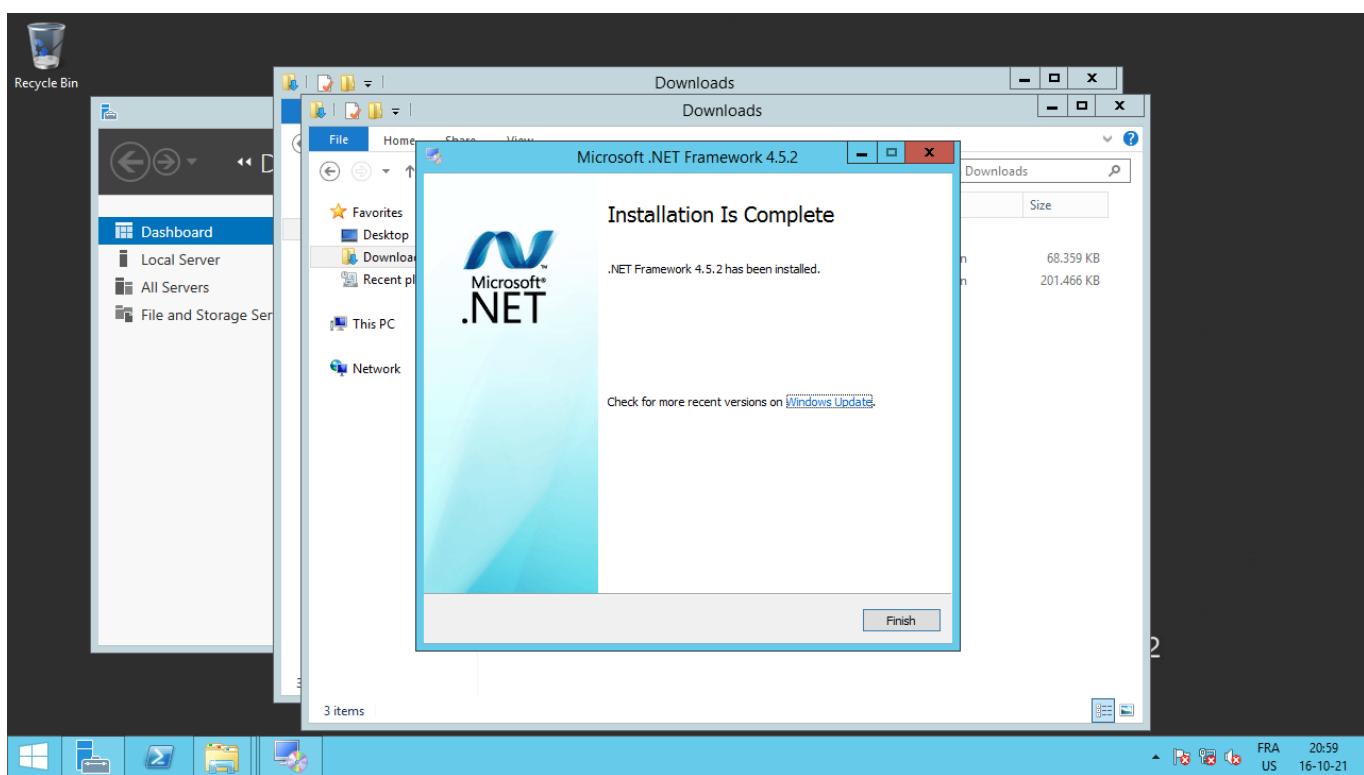
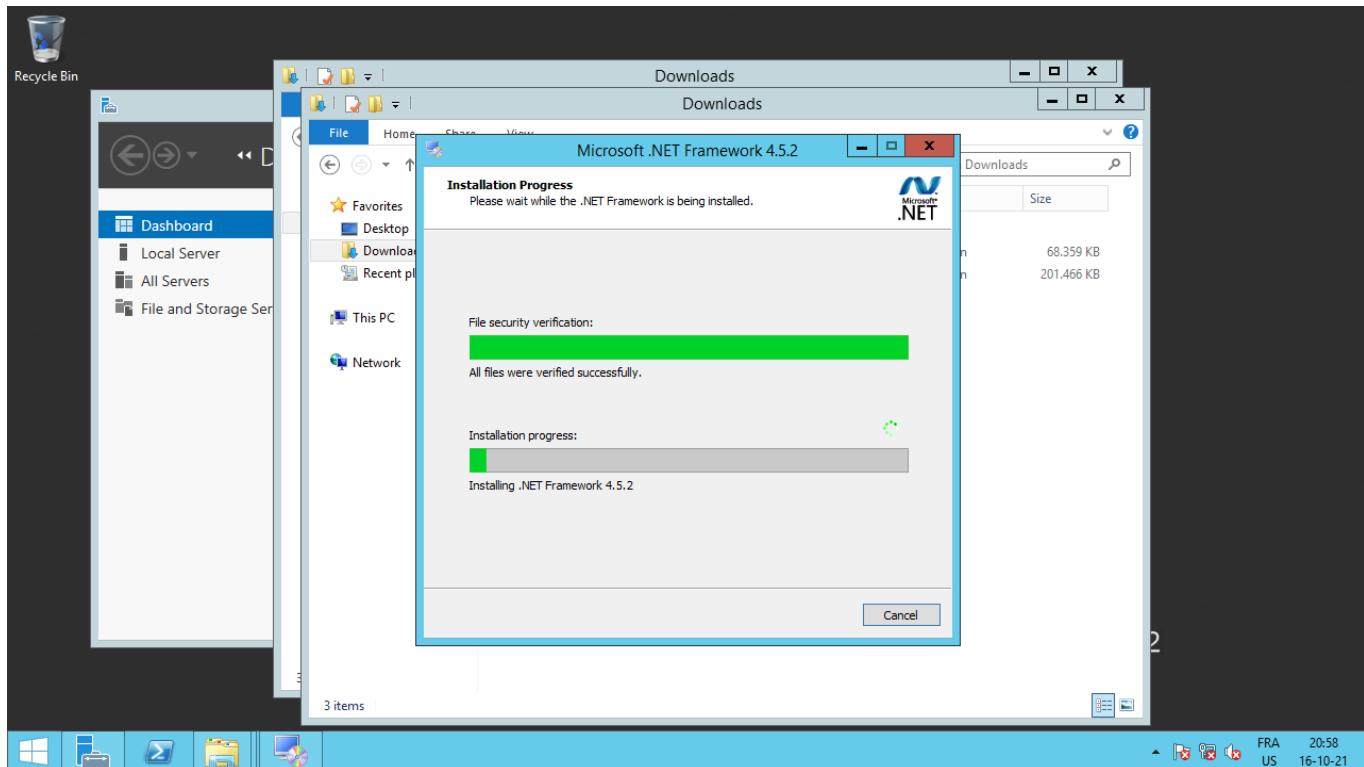
```
$ shasum NDP452-KB2901907-x86-x64-Allos-ENU.exe
89f86f9522dc7a8a965facce839abb790a285a63  NDP452-KB2901907-x86-x64-Allos-ENU.exe
```

<https://dotnet.microsoft.com/download/dotnet-framework/net452>

This is needed in order to avoid SSL/TLS error when launching the iMC installer. In my experiments the iMC installer database check fails if there isn't a bunch of updates ; something like:



Installation steps:

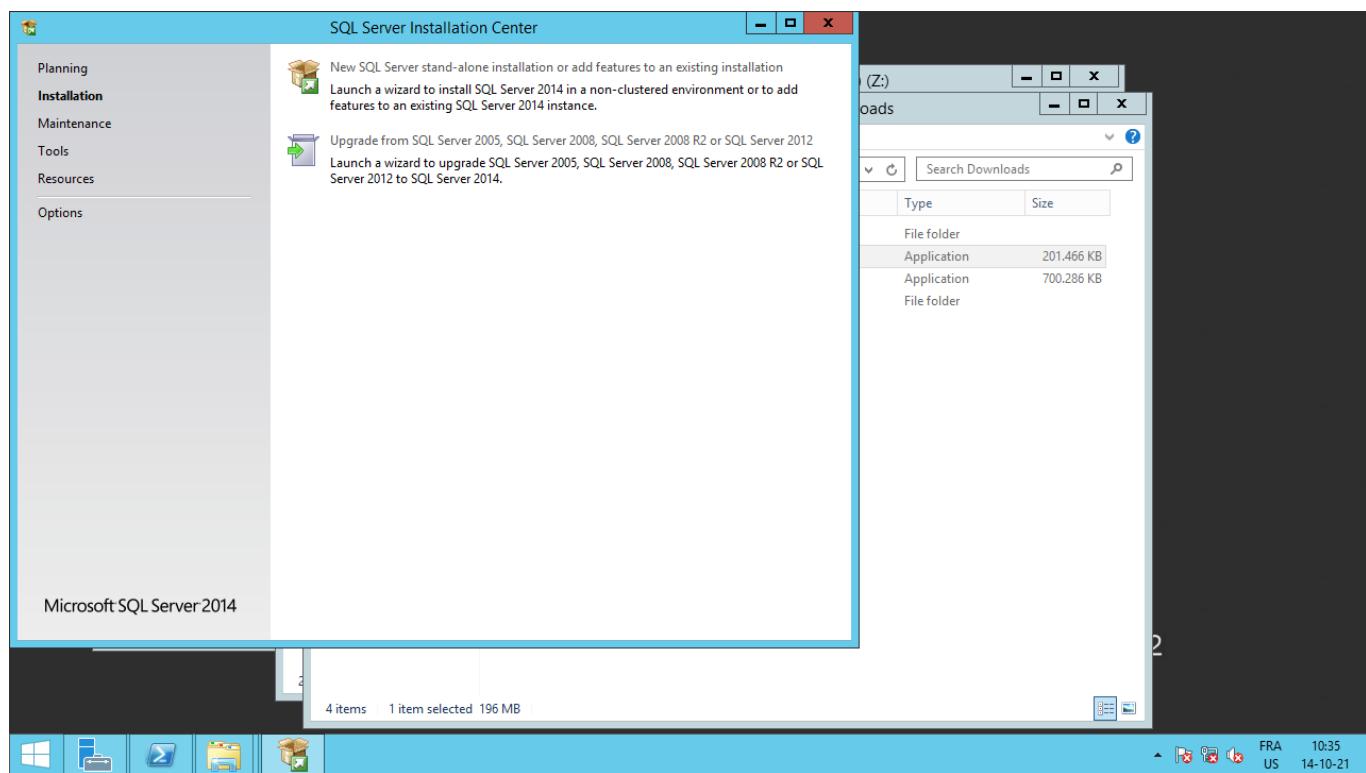
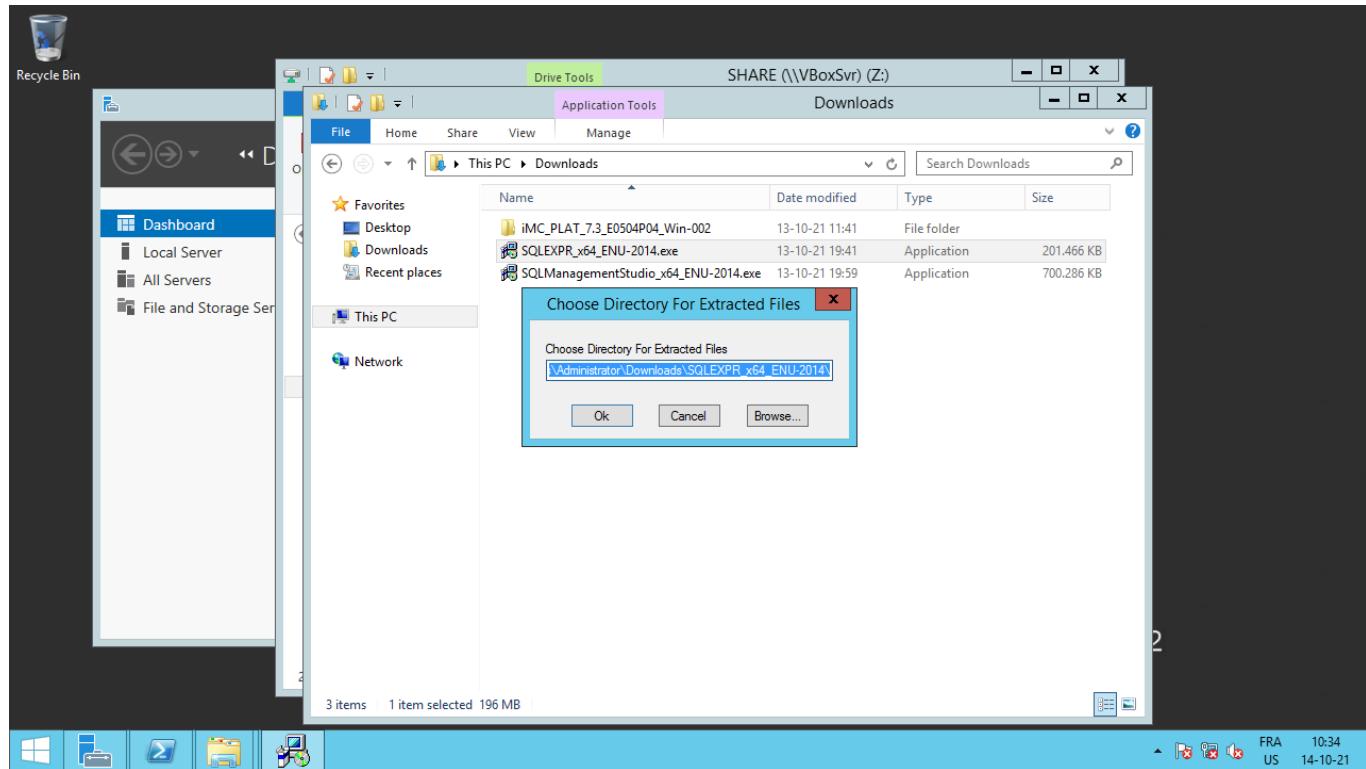


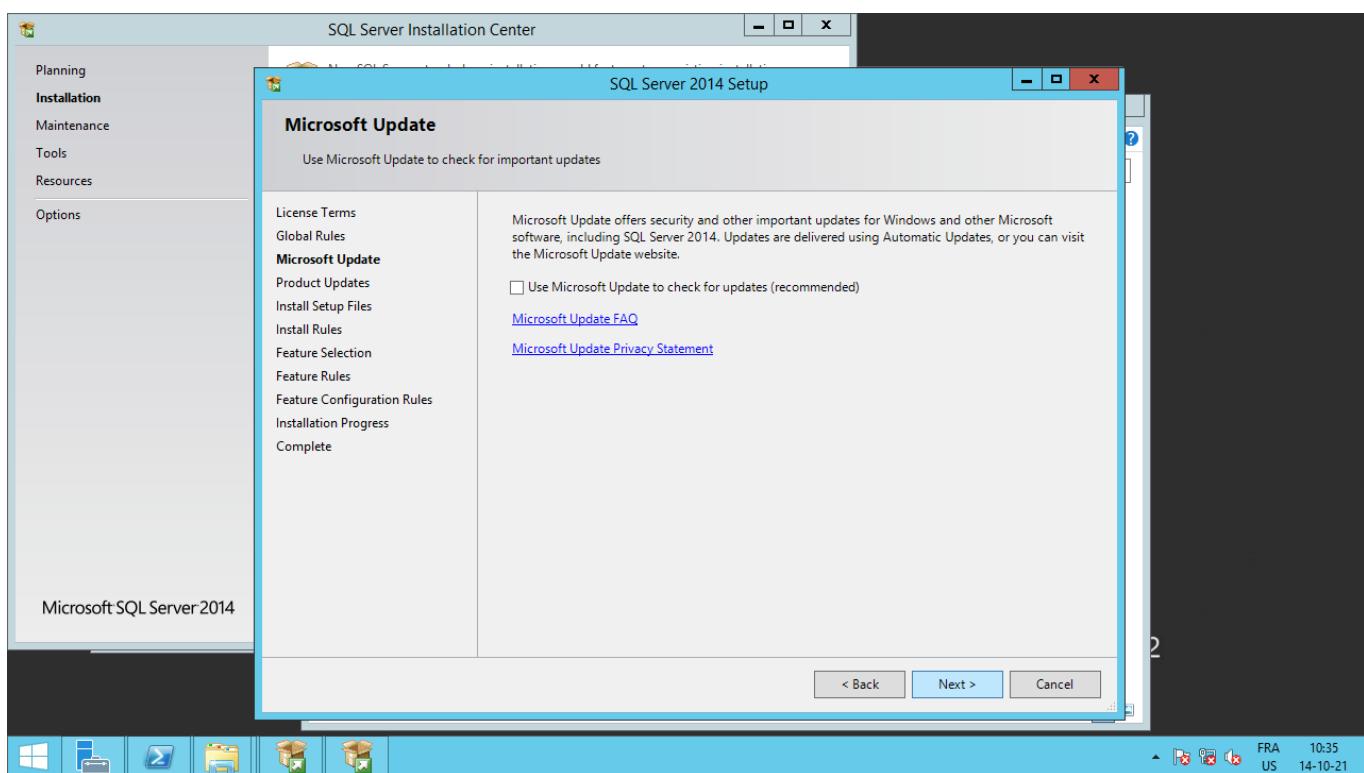
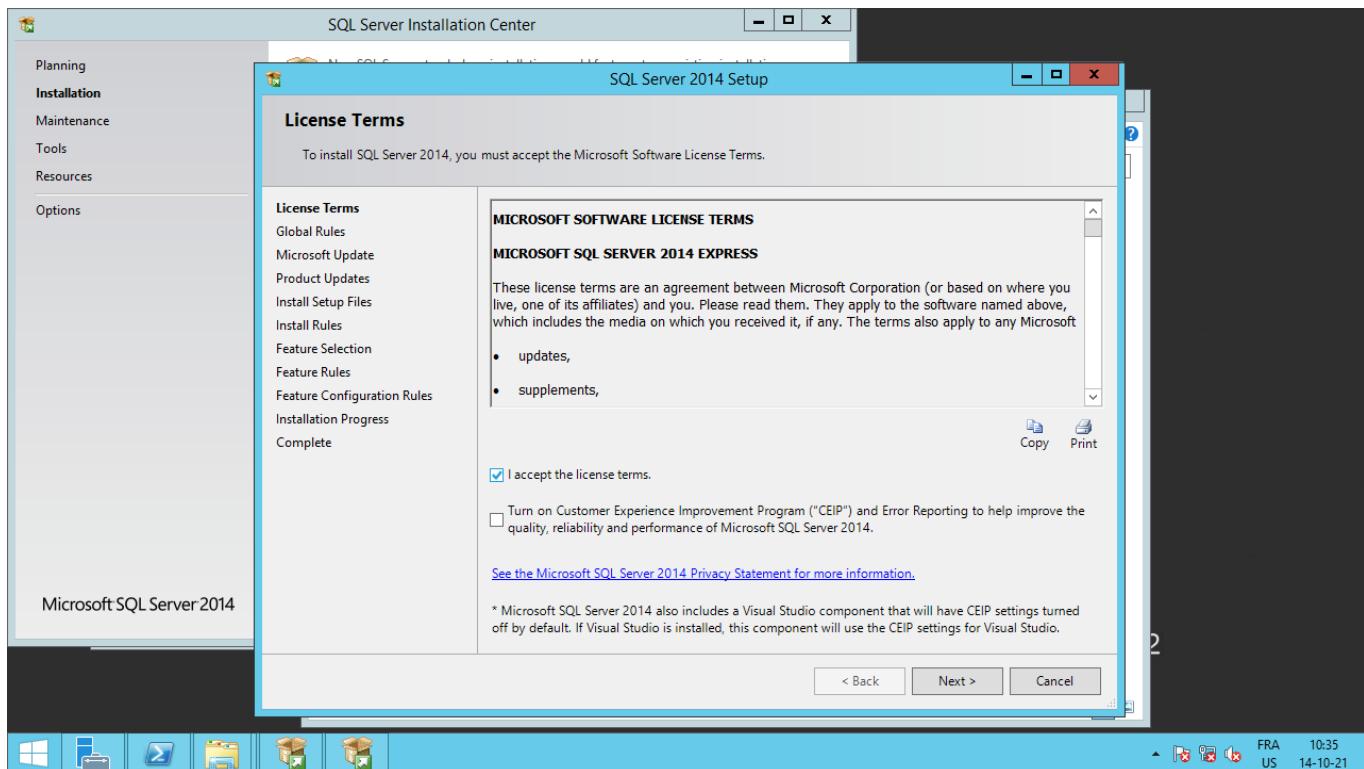
SQLServer2014Express:

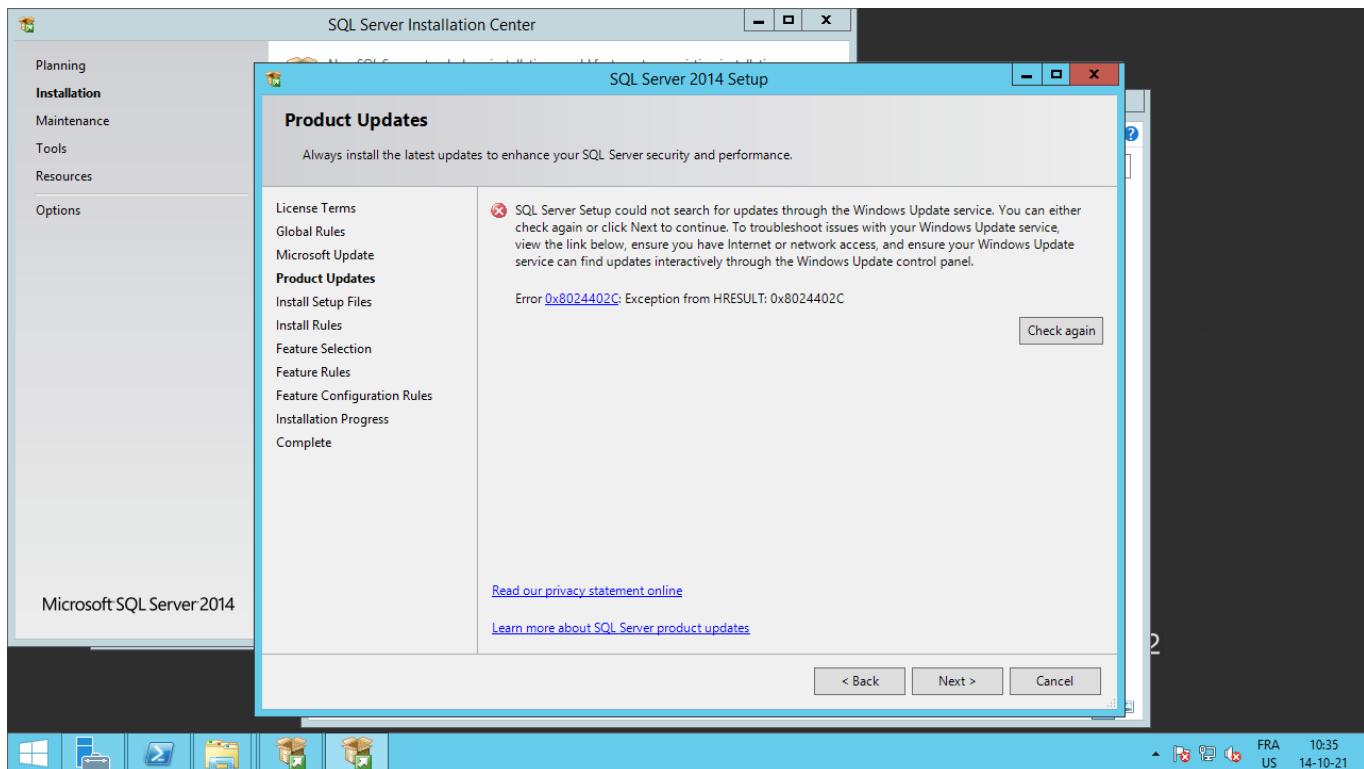
```
$ shasum SQLEXPR_x64_ENU-2014.exe
38e0d08f06af2f907e1aeabdca0444b828ae356b5  SQLEXPR_x64_ENU-2014.exe
```

<https://www.microsoft.com/en-us/download/details.aspx?id=42299>

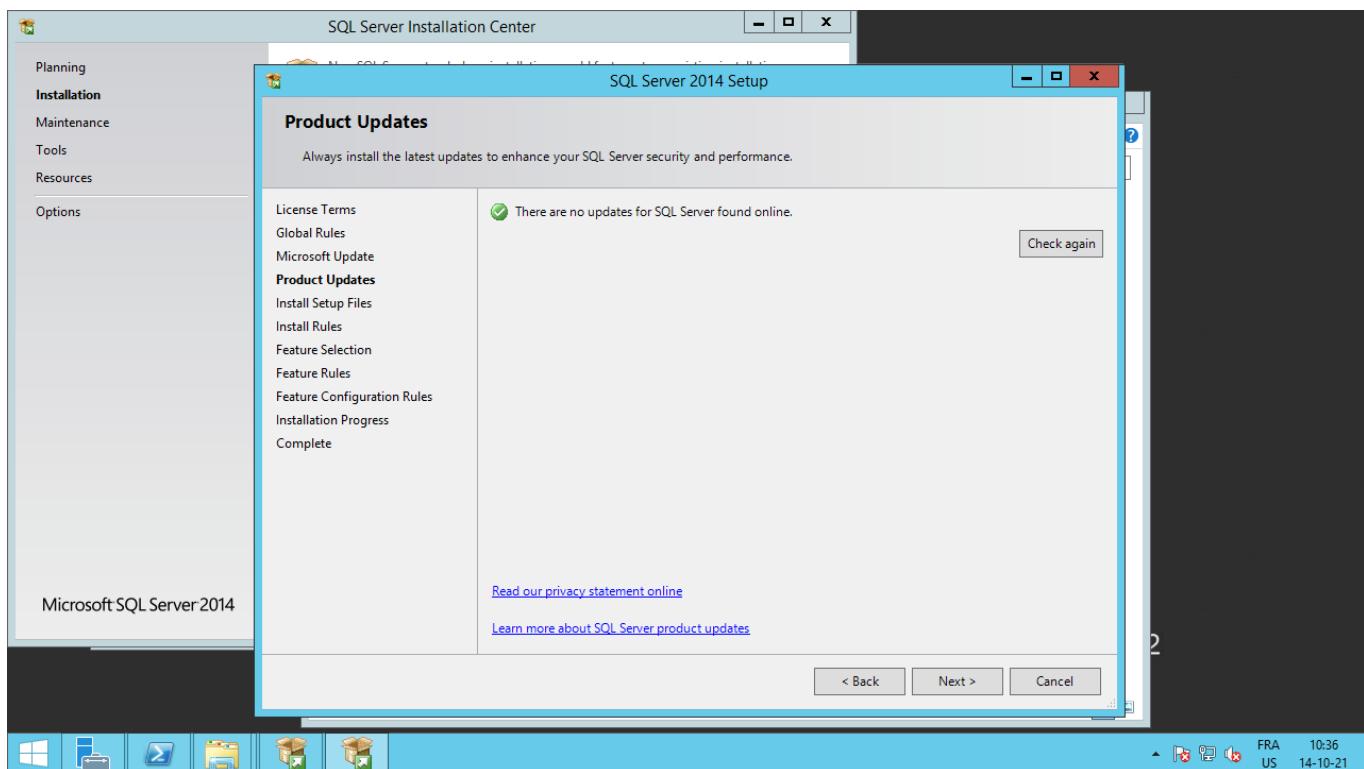
Installation steps:

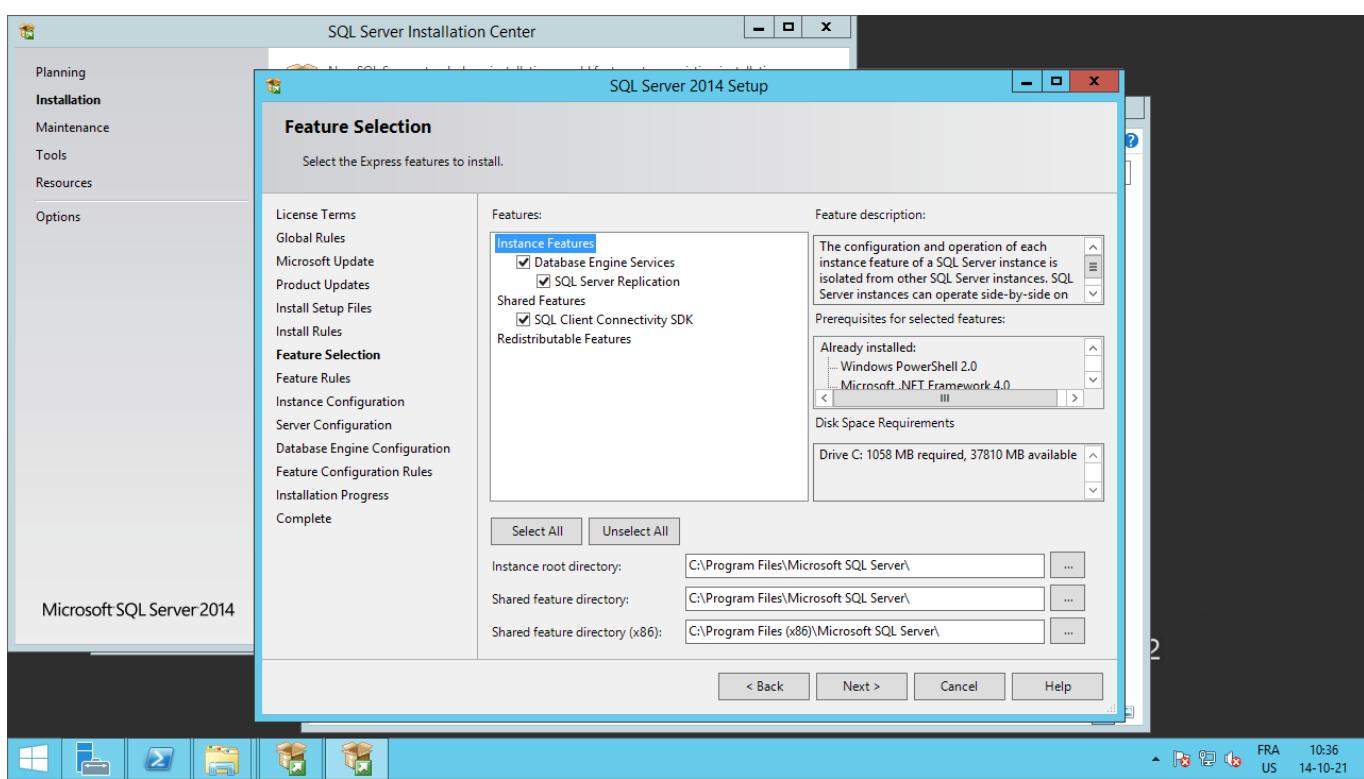
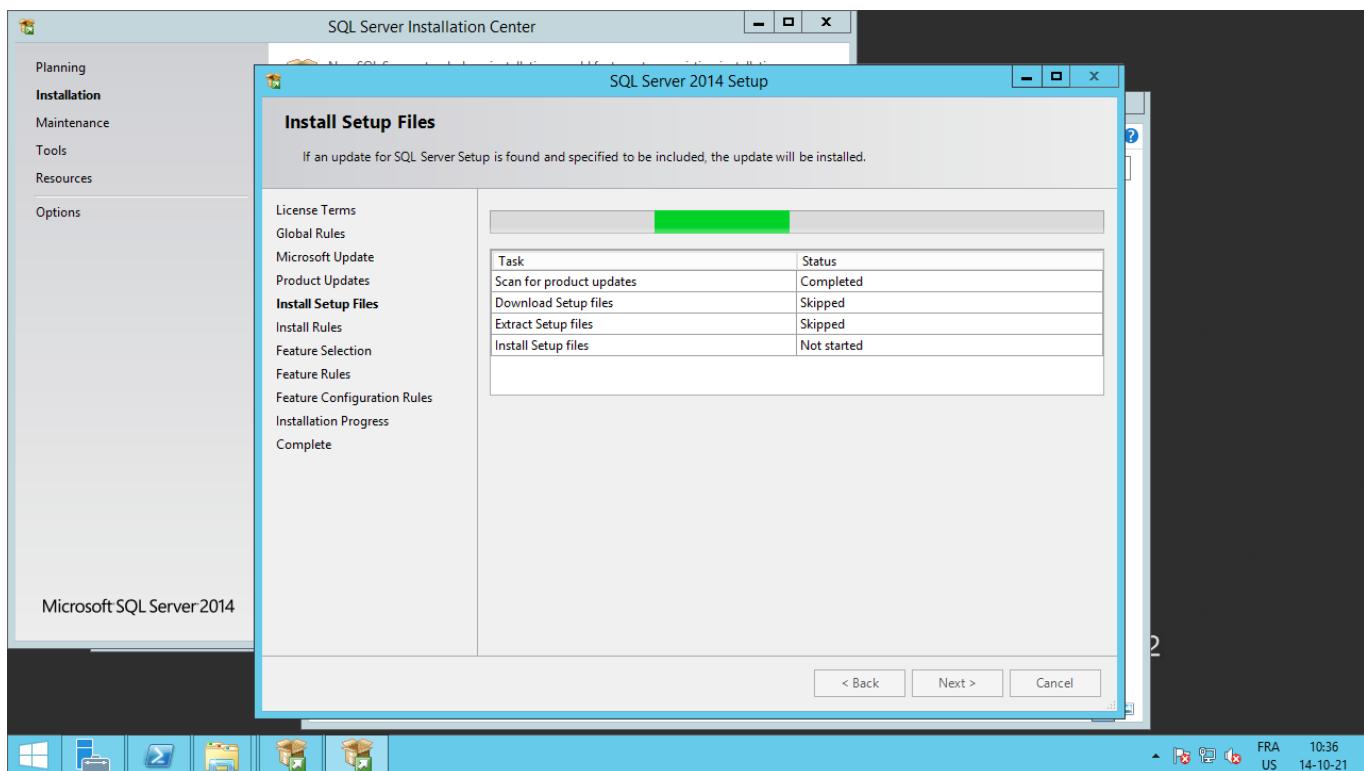


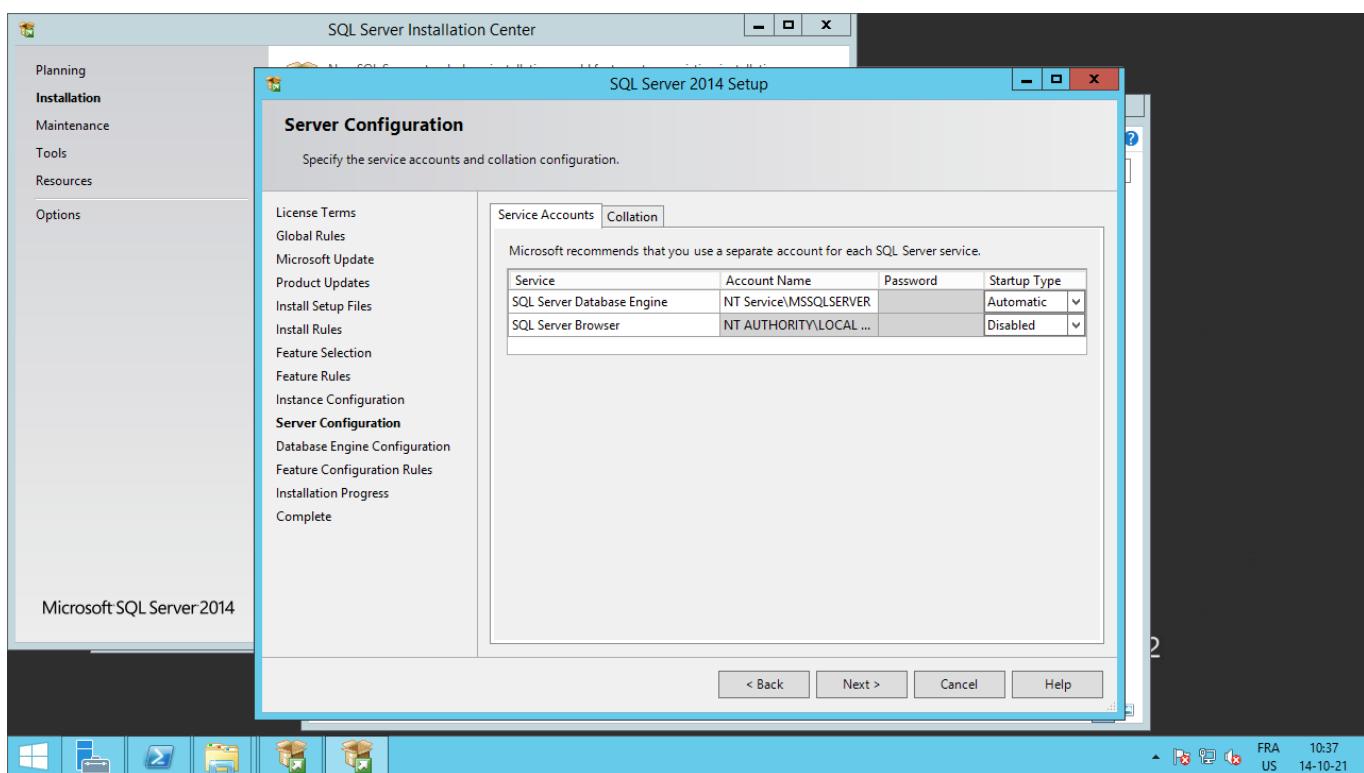
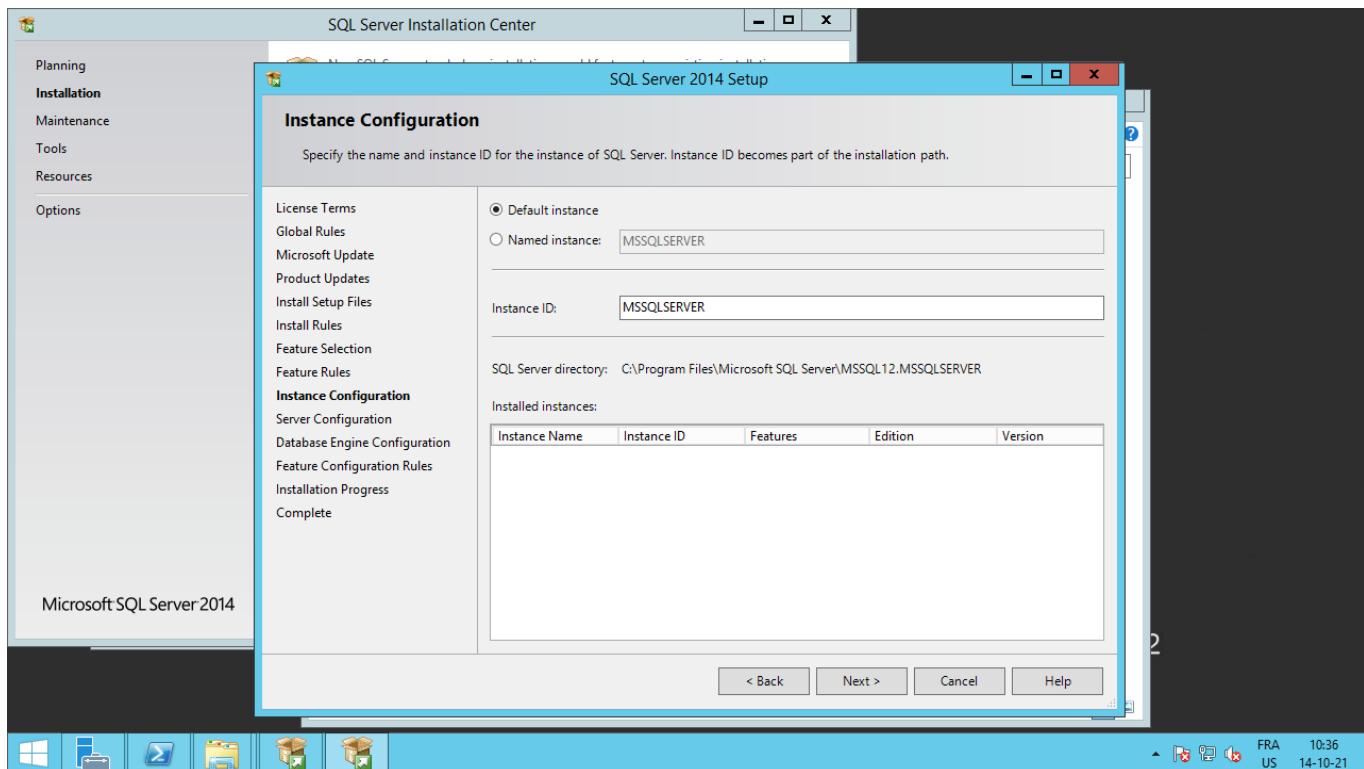


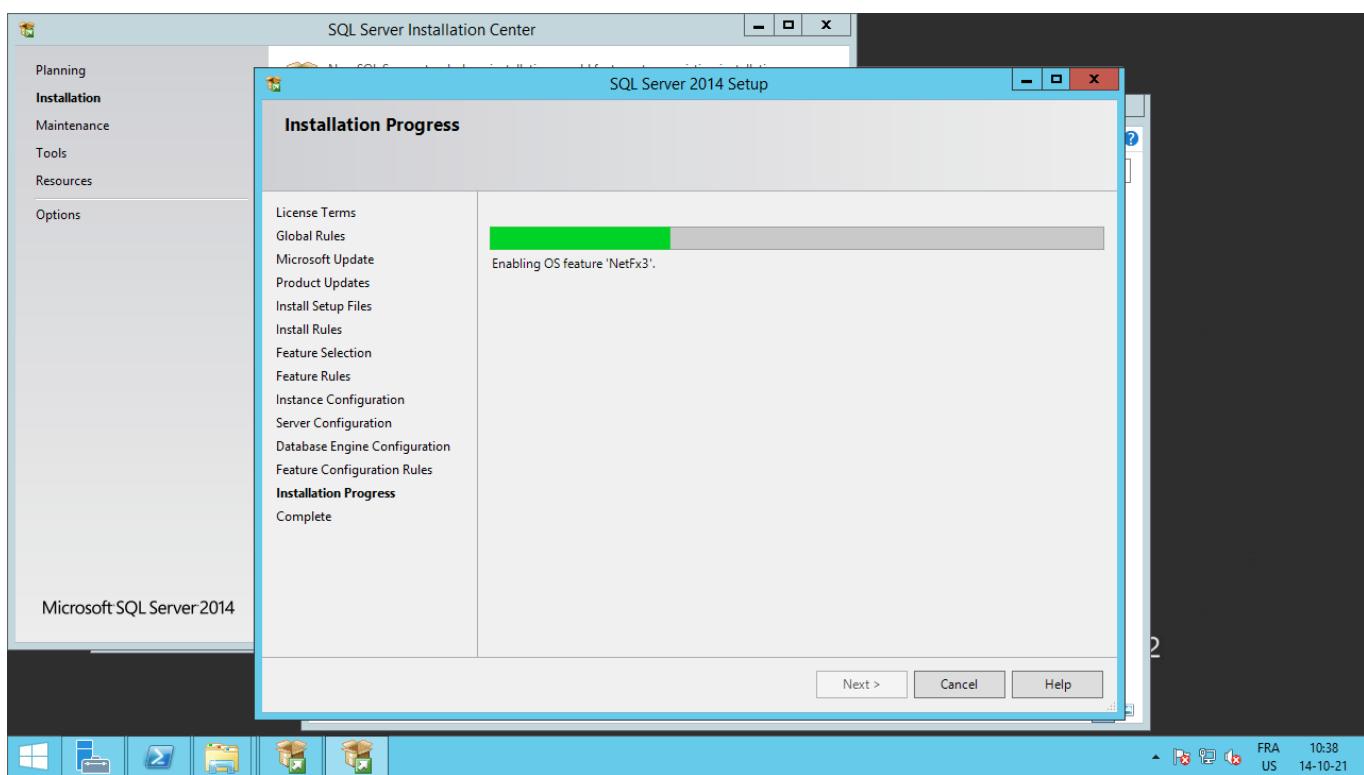
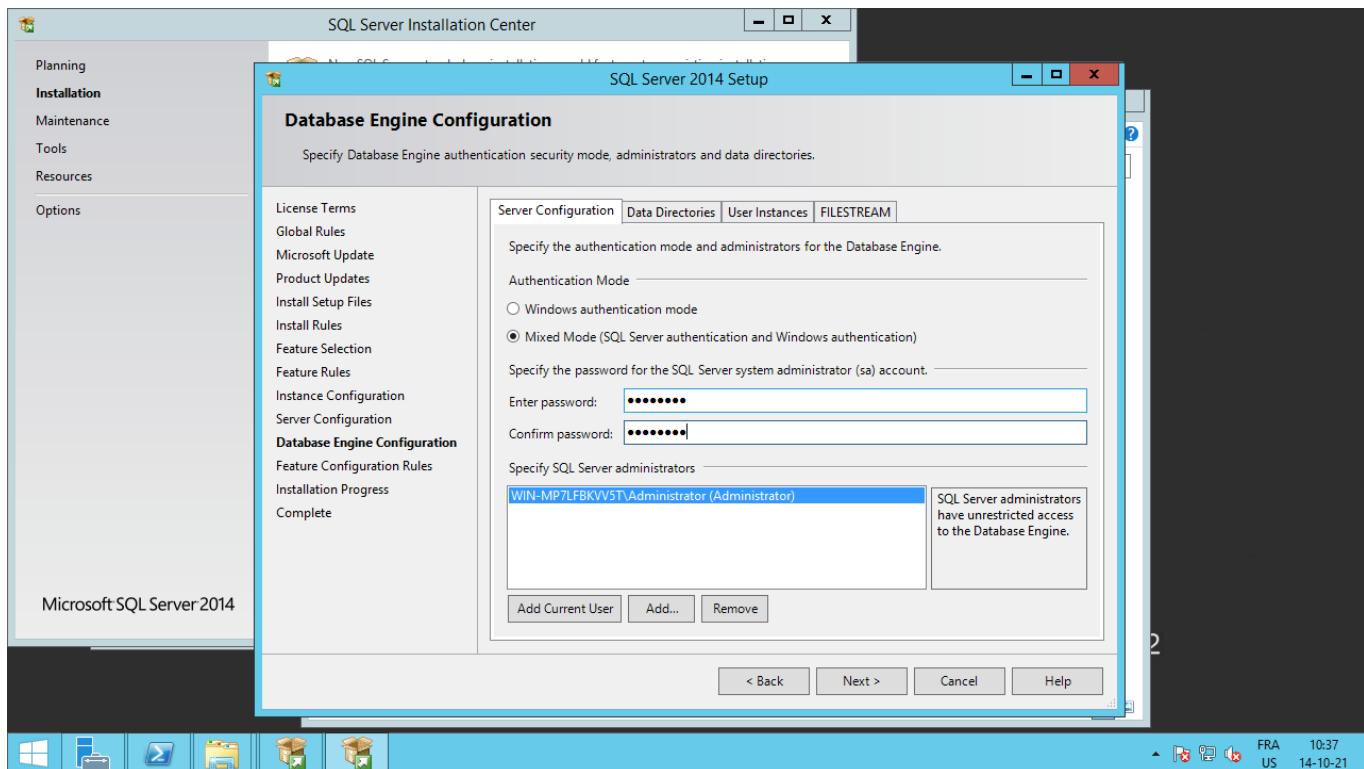


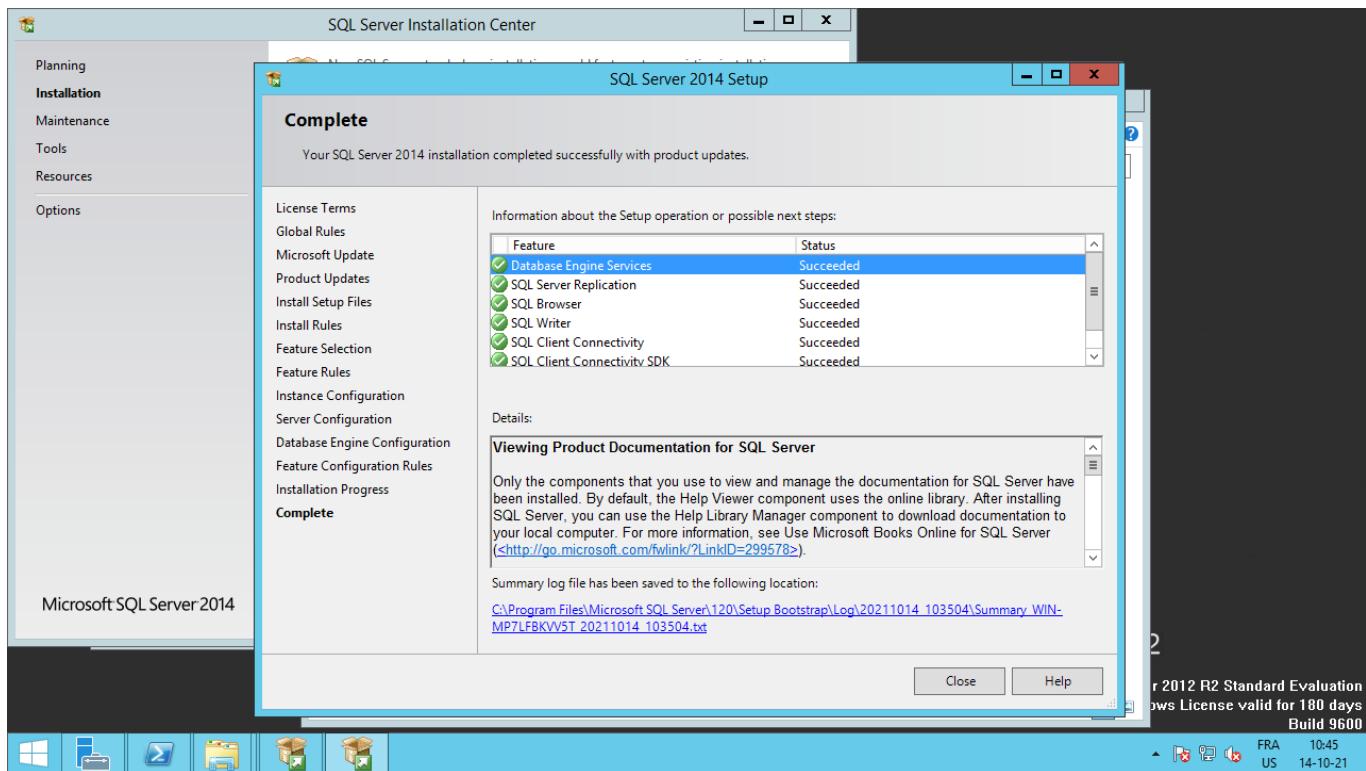
At this point, I enable Internet connectivity so that it can perform the check and also needed later on for the "NetFx3" install.











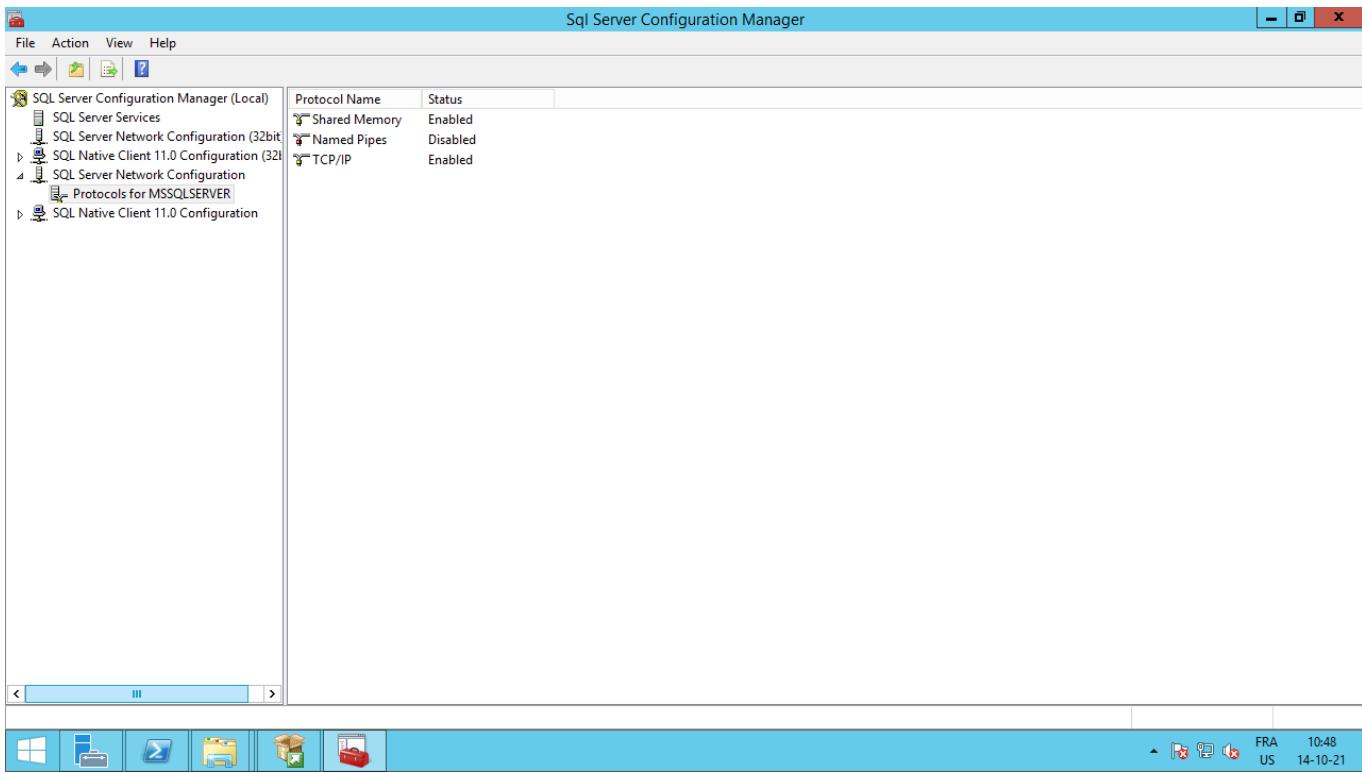
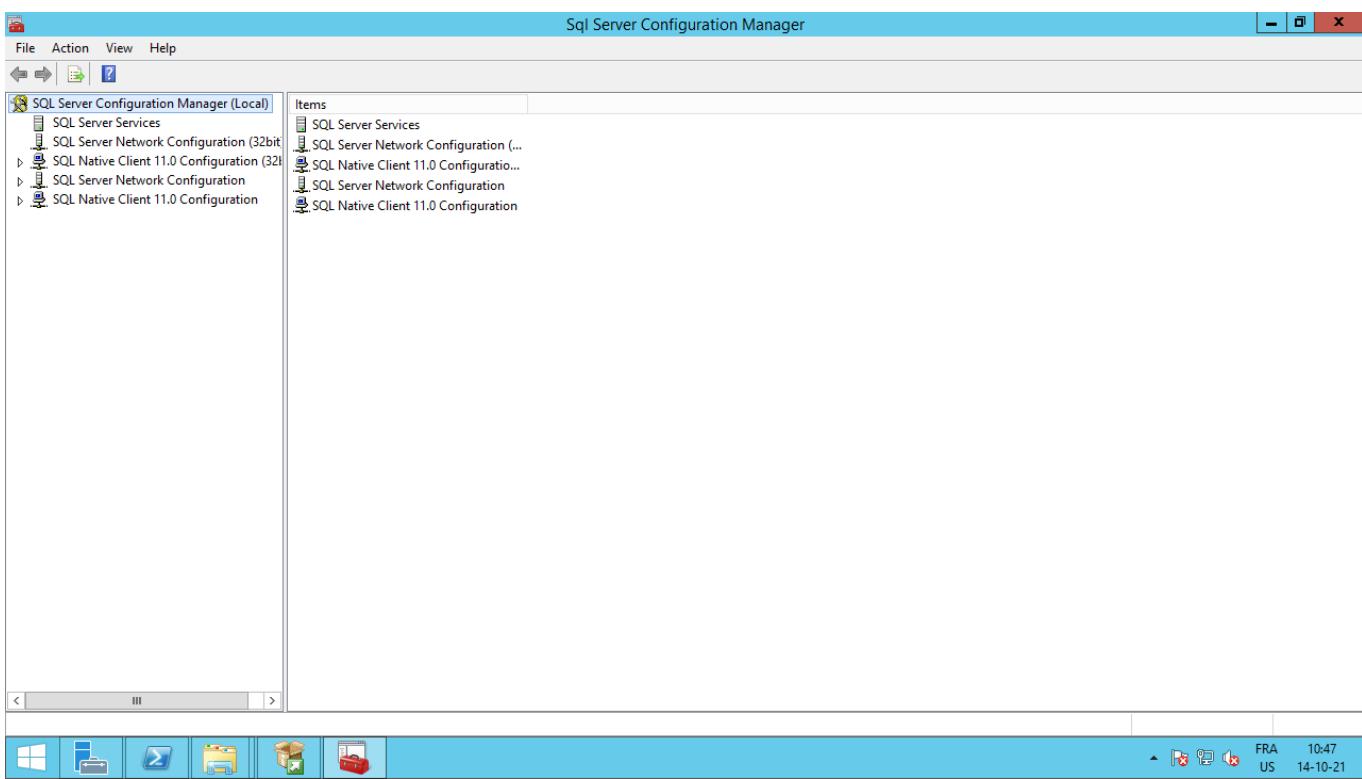
At this point, the db is installed but not yet exposed on port 1433:

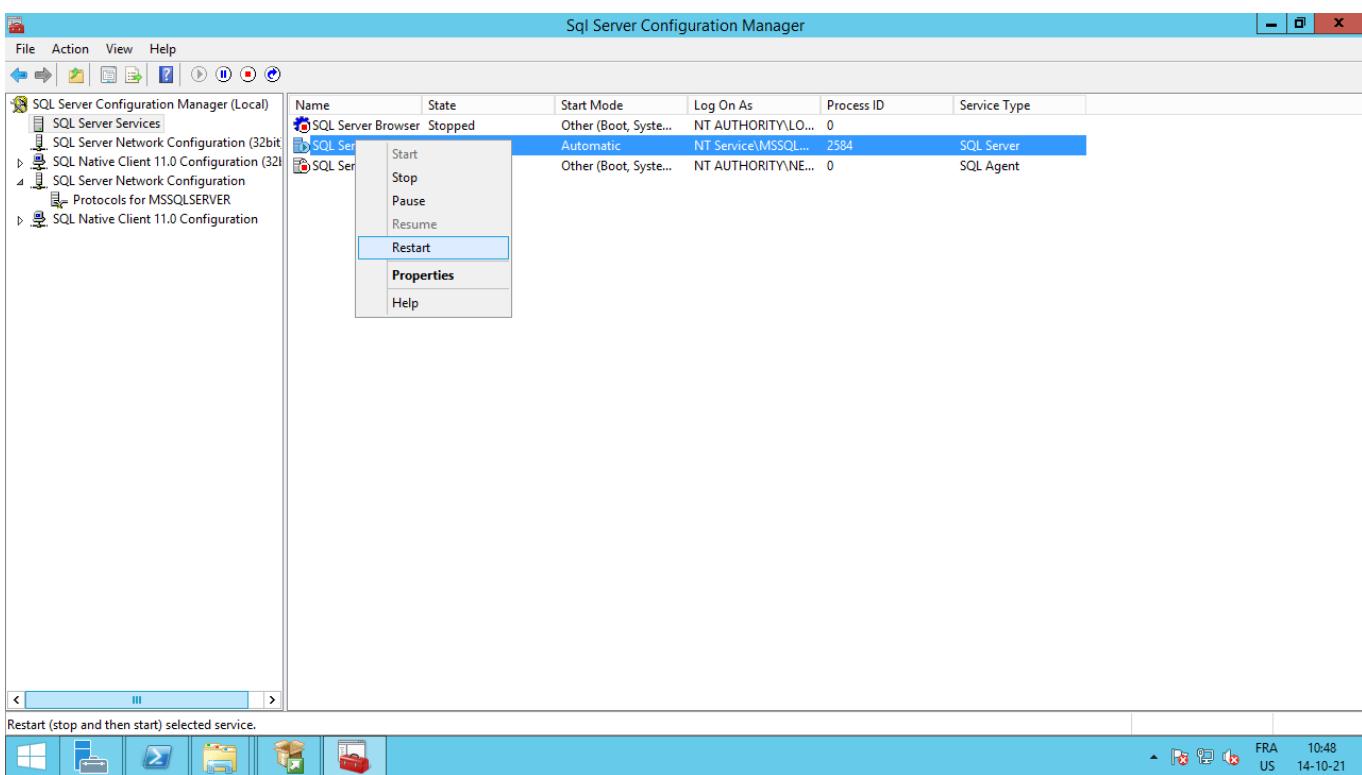
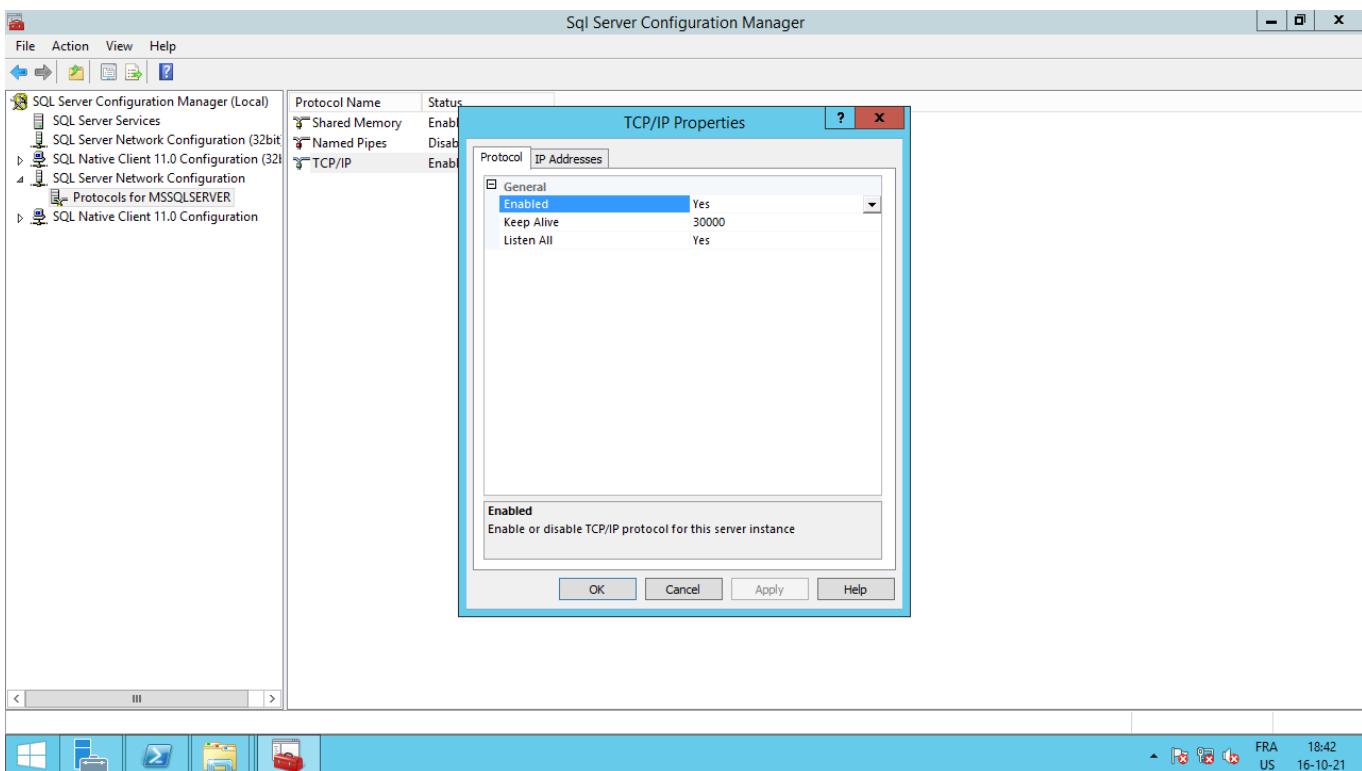
```
Administrator: Windows PowerShell
PS C:\Users\Administrator> netstat -a

Active Connections
Proto  Local Address          Foreign Address        State
TCP    0.0.0.0:49156           WIN-MP7LFBKV5T:0      LISTENING
TCP    0.0.0.0:49157           WIN-MP7LFBKV5T:0      LISTENING
TCP    10.0.2.15:139           WIN-MP7LFBKV5T:0      LISTENING
TCP    10.0.2.15:49189         WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:138                WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:445                WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:5985               WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:47001              WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:49152              WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:49153              WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:49154              WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:49155              WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:49156              WIN-MP7LFBKV5T:0      LISTENING
TCP    [::]:49157              WIN-MP7LFBKV5T:0      LISTENING
UDP   0.0.0.0:5355             *.*.
UDP   10.0.2.15:137           *.*.
UDP   10.0.2.15:138           *.*.
UDP   [::]:5355               *.*.

PS C:\Users\Administrator>
```

This can be achieved by enabling TCP/IP through the SQL Server Configuration Manager:





Administrator: Windows PowerShell

```
TCP 0.0.0.0:445 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:5985 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:47001 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:49152 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:49153 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:49154 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:49155 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:49156 WIN-MP2LFBKU5T:0 LISTENING
TCP 0.0.0.0:49157 WIN-MP2LFBKU5T:0 LISTENING
UDP 0.0.0.0:5355 *:*
UDP 10.0.2.15:137 *:*
UDP 10.0.2.15:138 *:*
UDP [::]:5355 *:*
PS C:\Users\Administrator> netstat -a

Active Connections

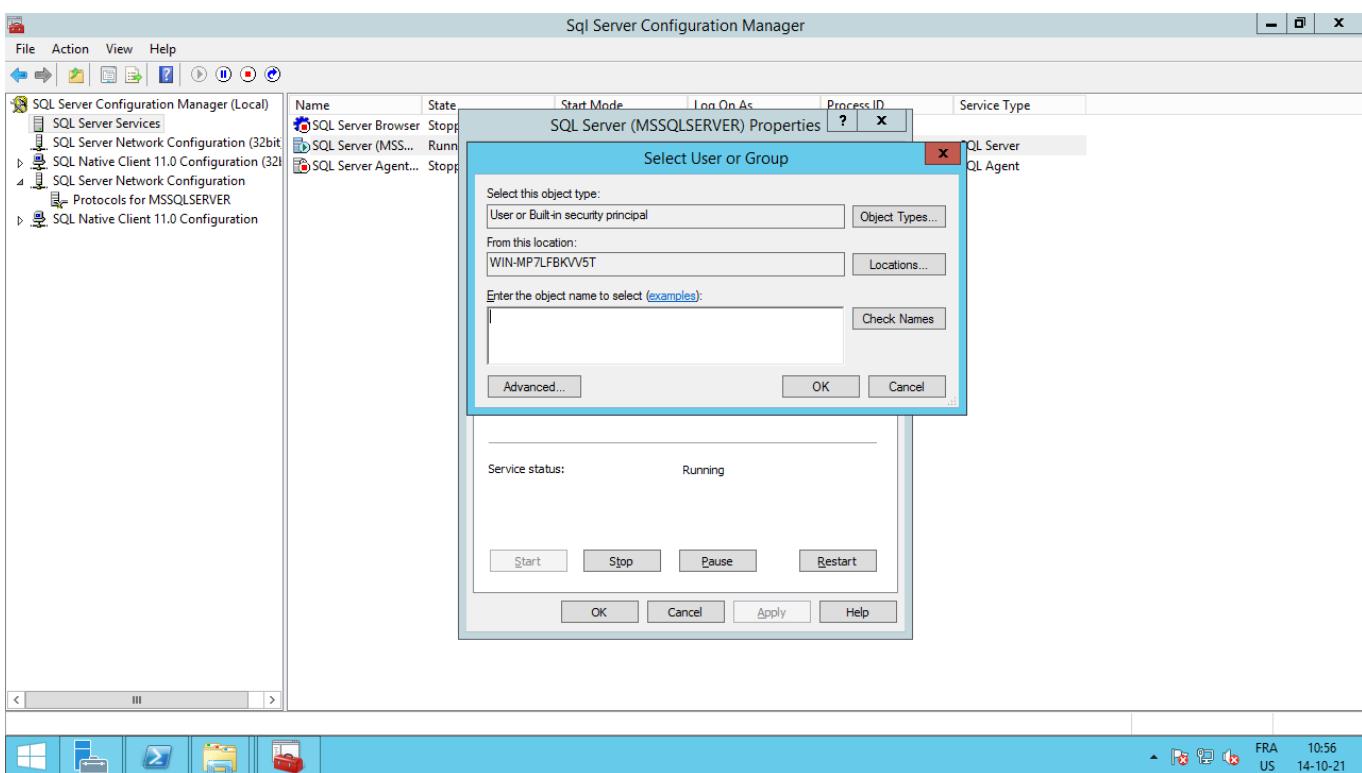
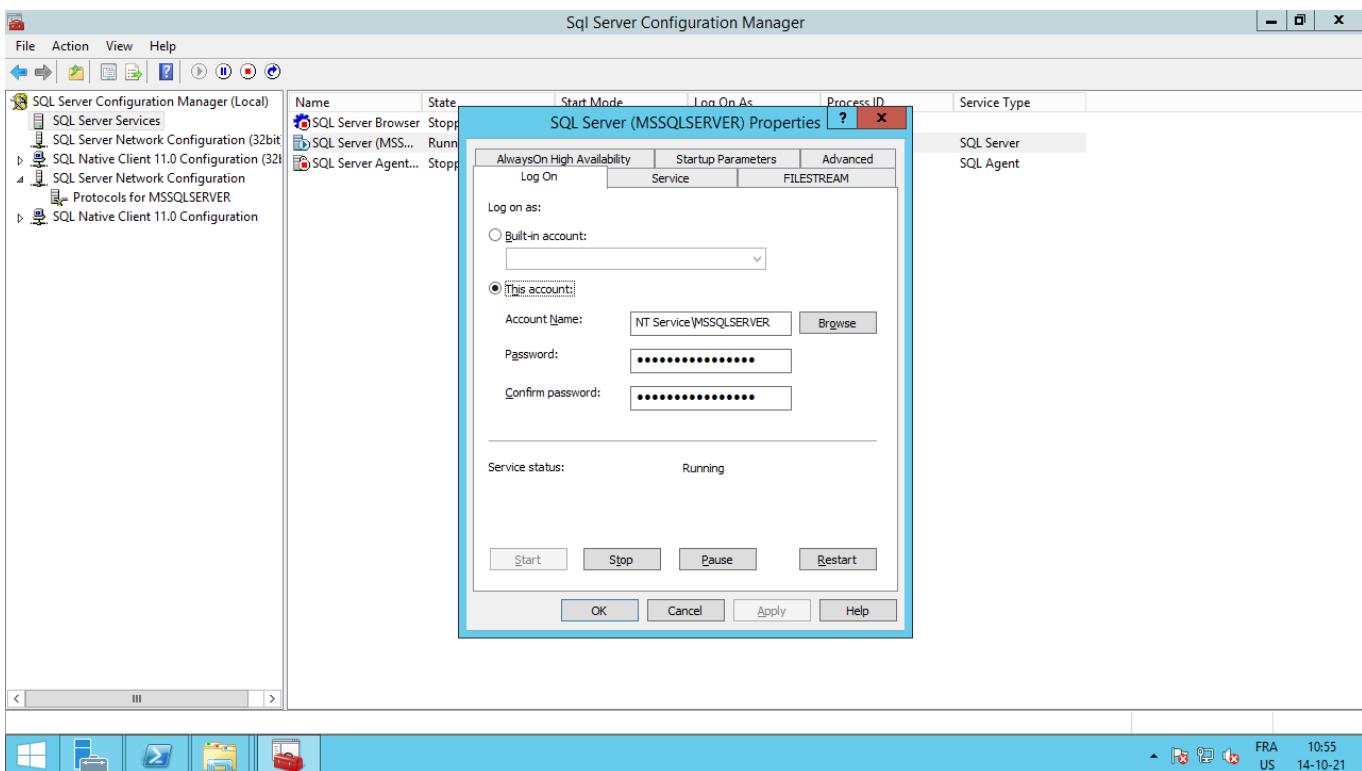
  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135           WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:445           WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:47001         WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:49152         WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:49153         WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:49154         WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:49155         WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:49156         WIN-MP2LFBKU5T:0  LISTENING
  TCP    0.0.0.0:49157         WIN-MP2LFBKU5T:0  LISTENING
  TCP    10.0.2.15:139         WIN-MP2LFBKU5T:0  LISTENING
  TCP    10.0.2.15:49190       51.143.49.66:https ESTABLISHED
  TCP    [::]:135              WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:445              WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:47001             WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:49152             WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:49153             WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:49154             WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:49155             WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:49156             WIN-MP2LFBKU5T:0  LISTENING
  TCP    [::]:49157             WIN-MP2LFBKU5T:0  LISTENING
  UDP  0.0.0.0:5355           *:*
  UDP  10.0.2.15:137          *:*
  UDP  10.0.2.15:138          *:*
  UDP  [::]:5355               *:*
PS C:\Users\Administrator> netstat -a | findstr 1433
TCP 0.0.0.0:1433           WIN-MP2LFBKU5T:0  LISTENING
TCP [::]:1433                WIN-MP2LFBKU5T:0  LISTENING
PS C:\Users\Administrator>
```

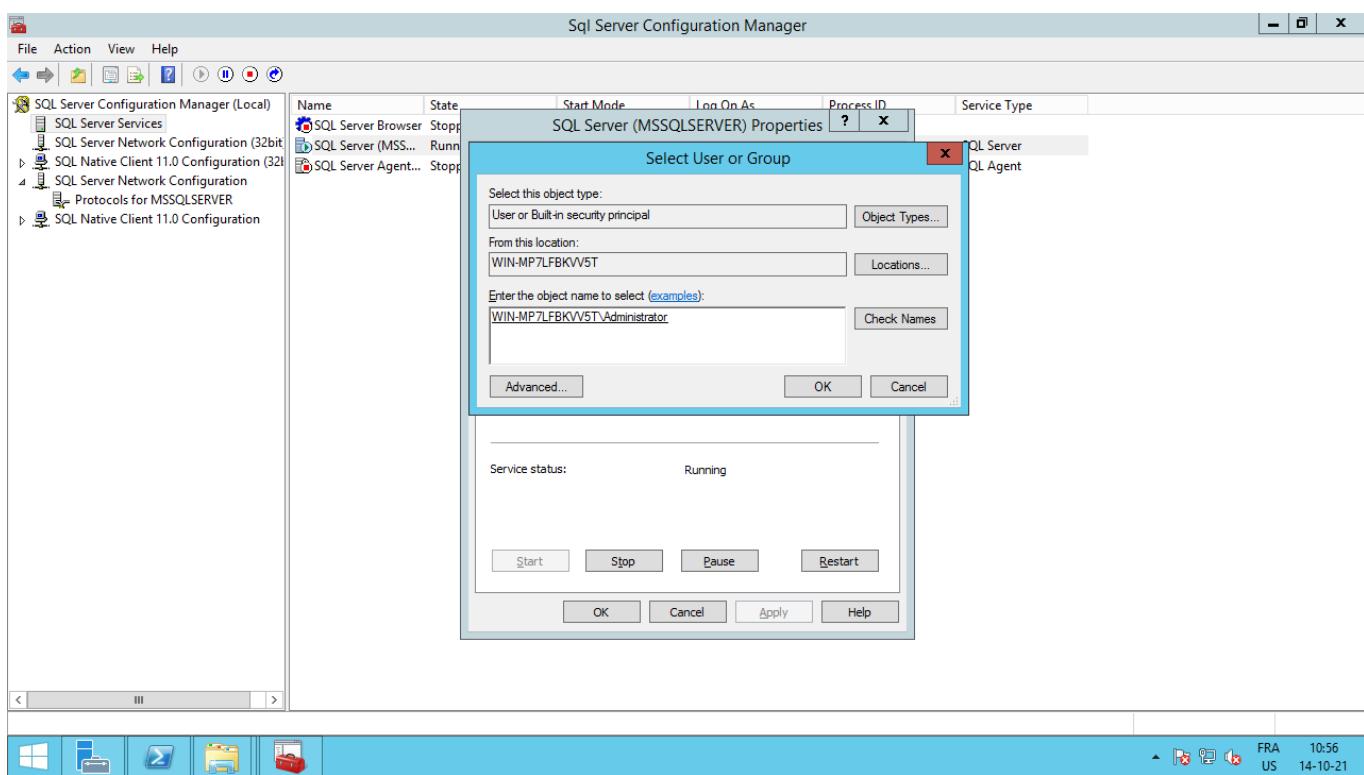
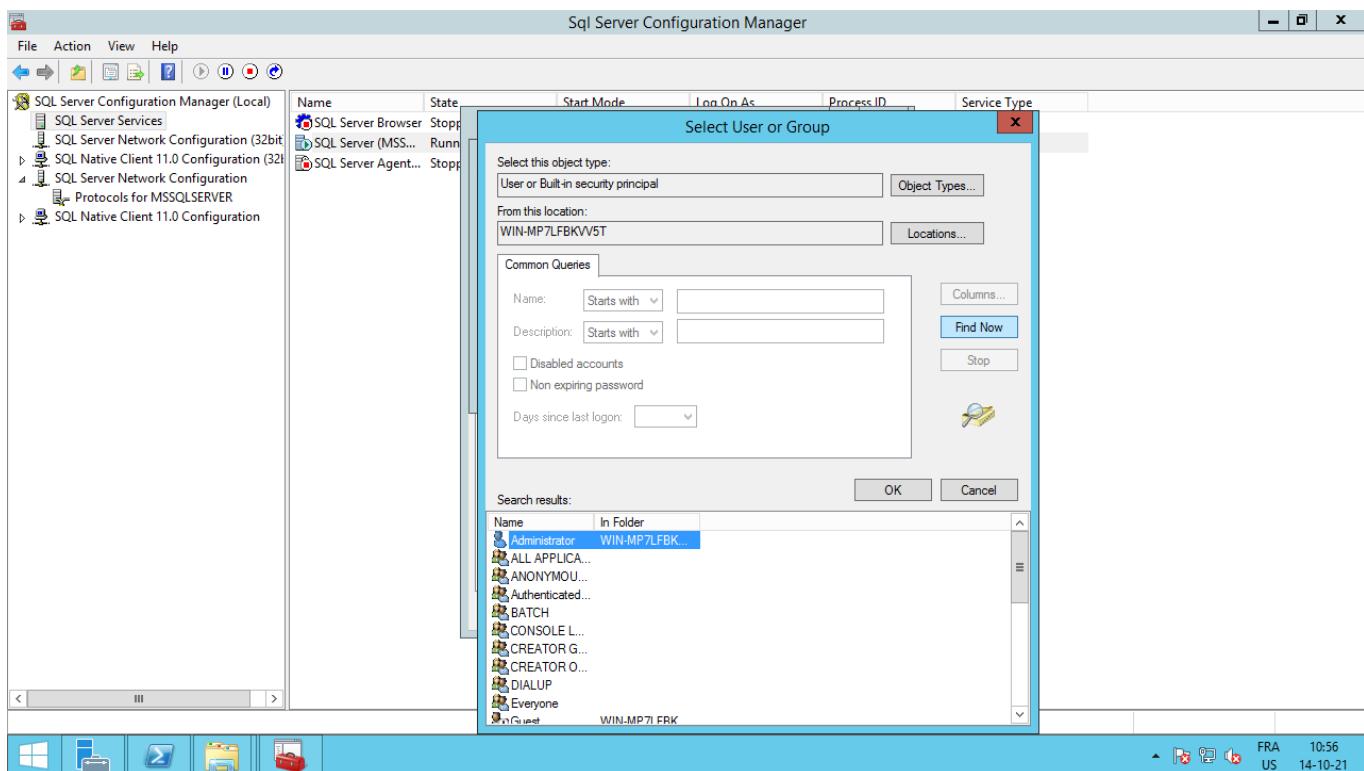
At this point, the SQLServer2014Express is installed and running on port 1433 with default configuration.

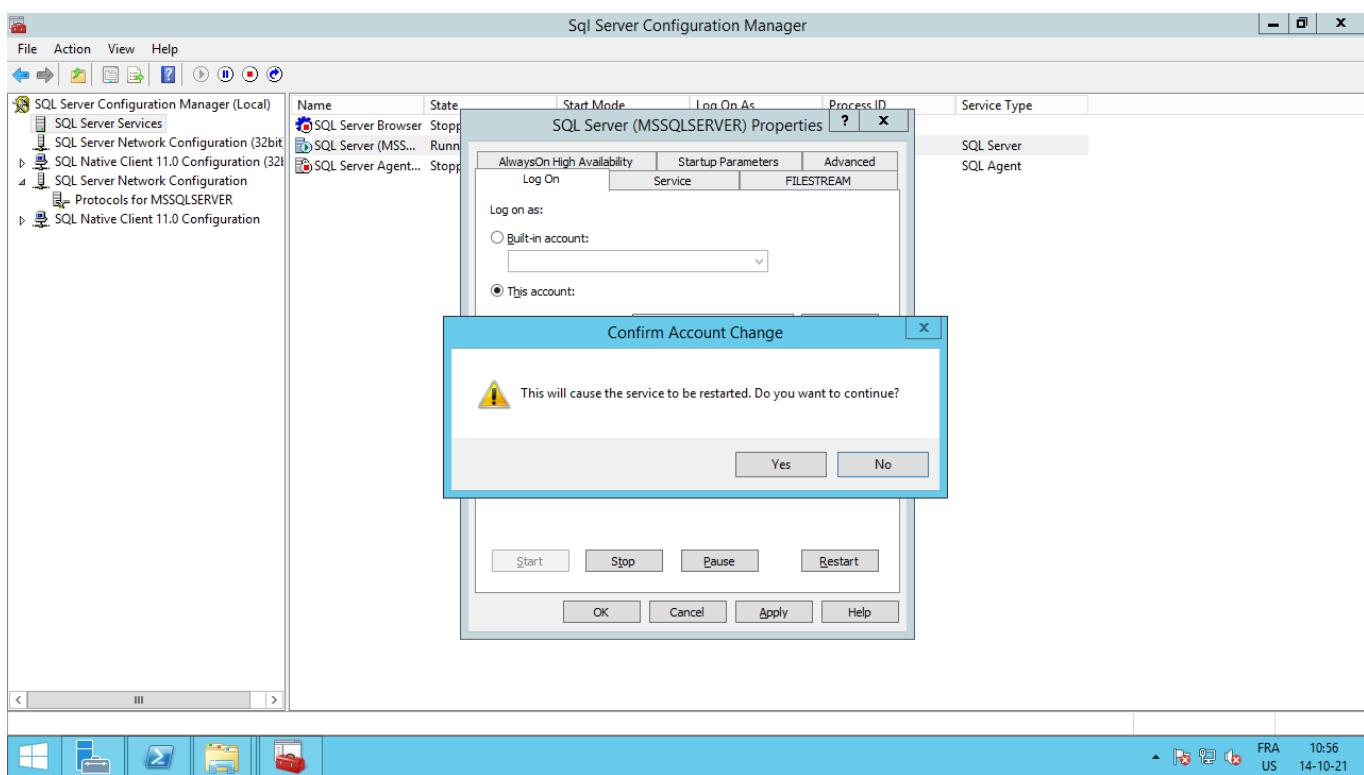
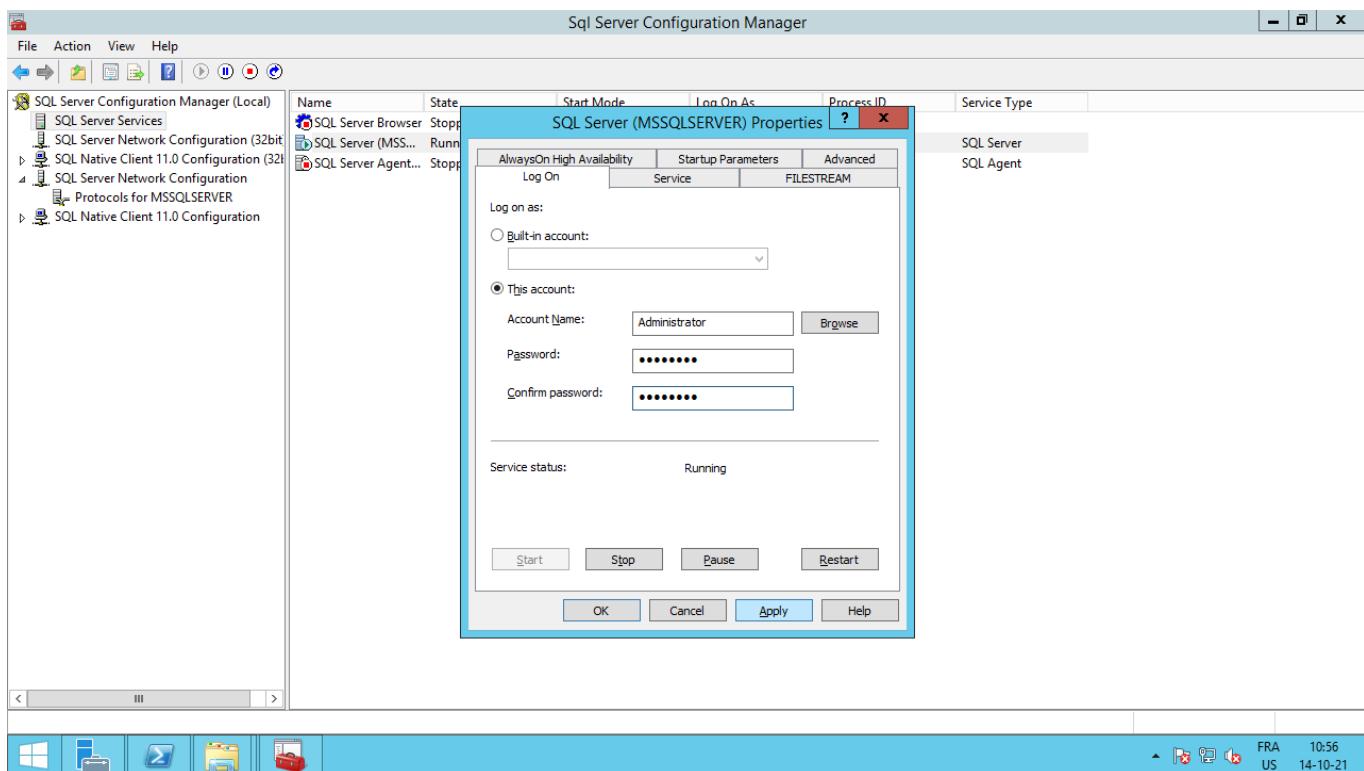
However, if the iMC installer is launched it will complain that the current user doesn't have the correct privileges, something like:

The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell" with several network connection logs. Overlaid on it is the "HPE iMC Installation Wizard" window. The wizard's main screen displays "Welcome to HPF iMC Installation Wizard" and "Checking installation parameters". It includes dropdowns for "Database Type" (Microsoft SQL Server), "Instance Name" (Default Instance), and "Superuser" (sa). A progress bar at the bottom indicates "Checking database" is in progress. A modal dialog box titled "Error" is centered, stating: "Failed to create database. Please confirm you have already created the local data folder "C:\Program Files\imcdata", and the user with the SQL server service enabled has write privilege to the folder." Buttons for "OK" and "Cancel" are present. At the bottom of the wizard window, there are "Next >" and "Cancel" buttons.

To circumvent this issue, we change the account with which SQLServer is accessed, from the default NT service account to the administrator:







Given this configuration, we should be able to move flawlessly through the iMC installer.

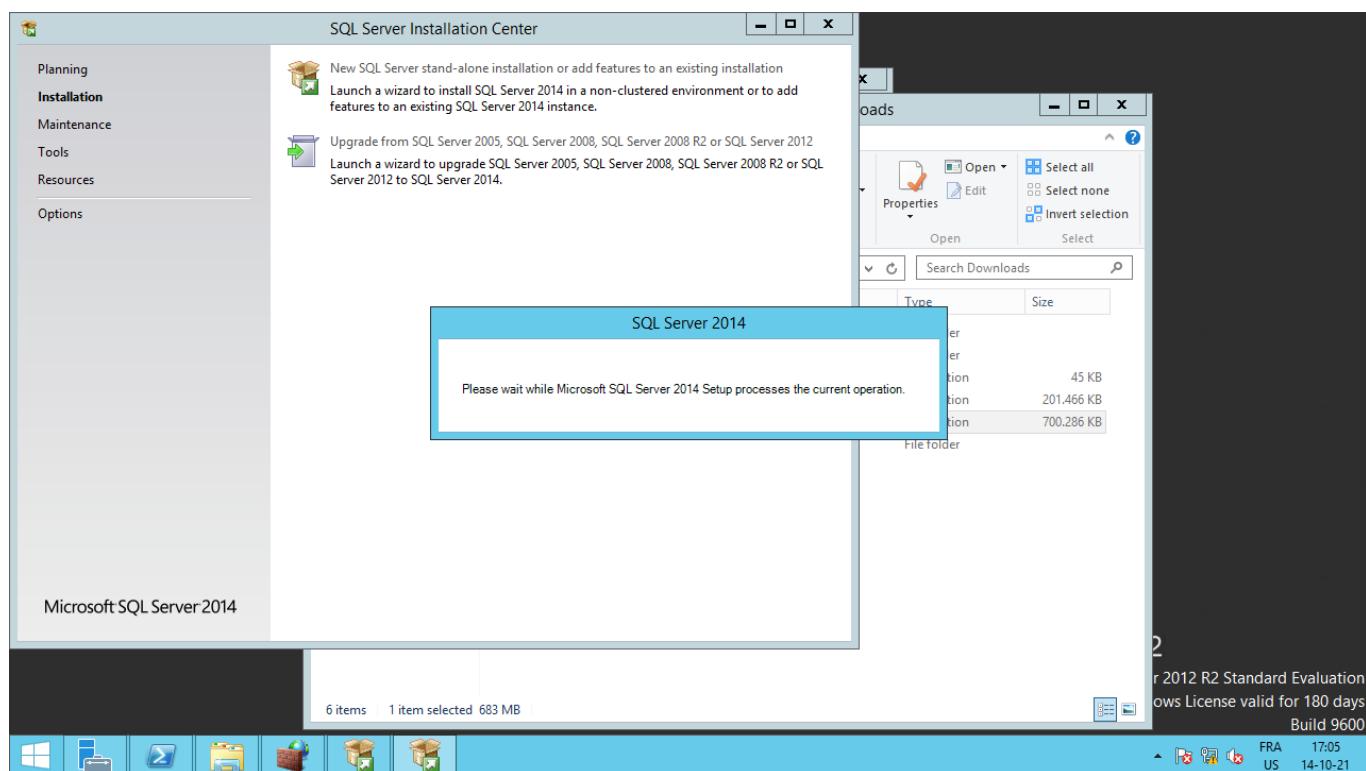
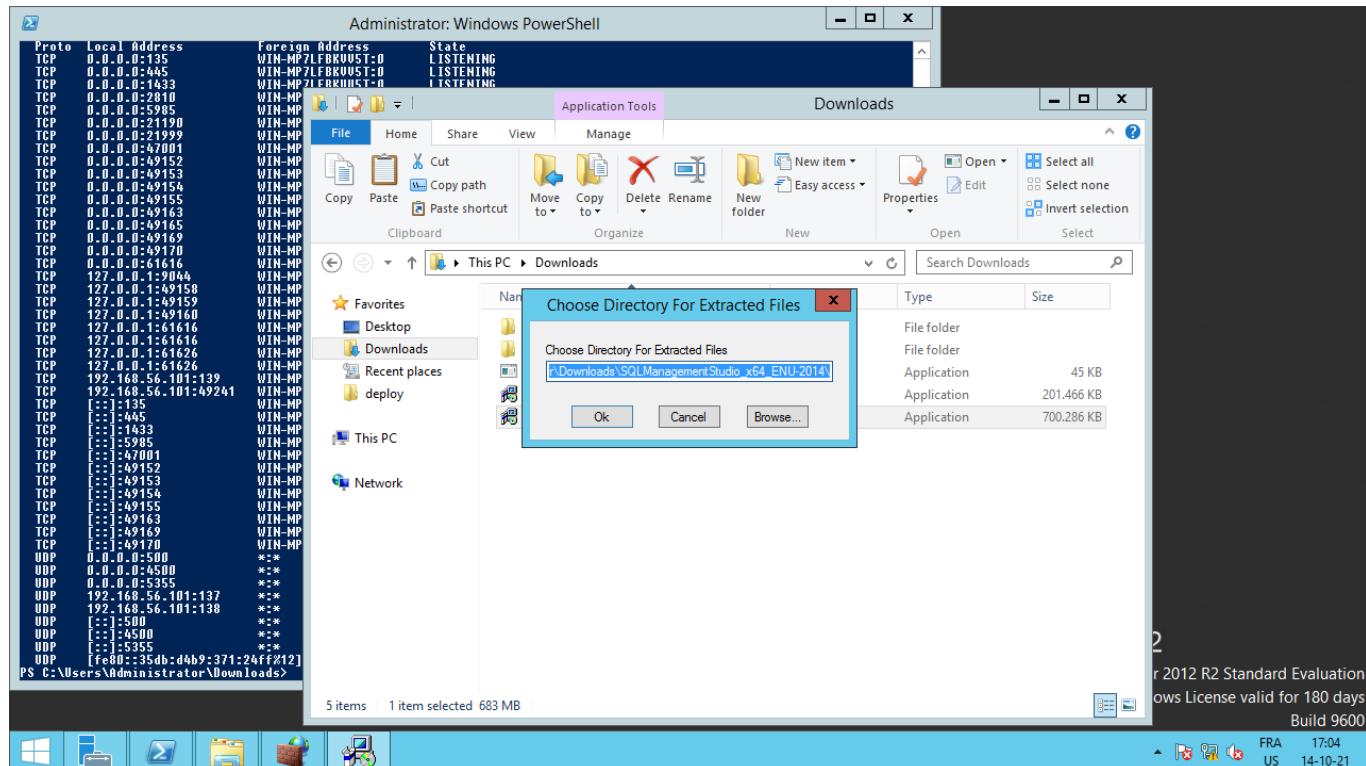
SQLManagementStudio2014:

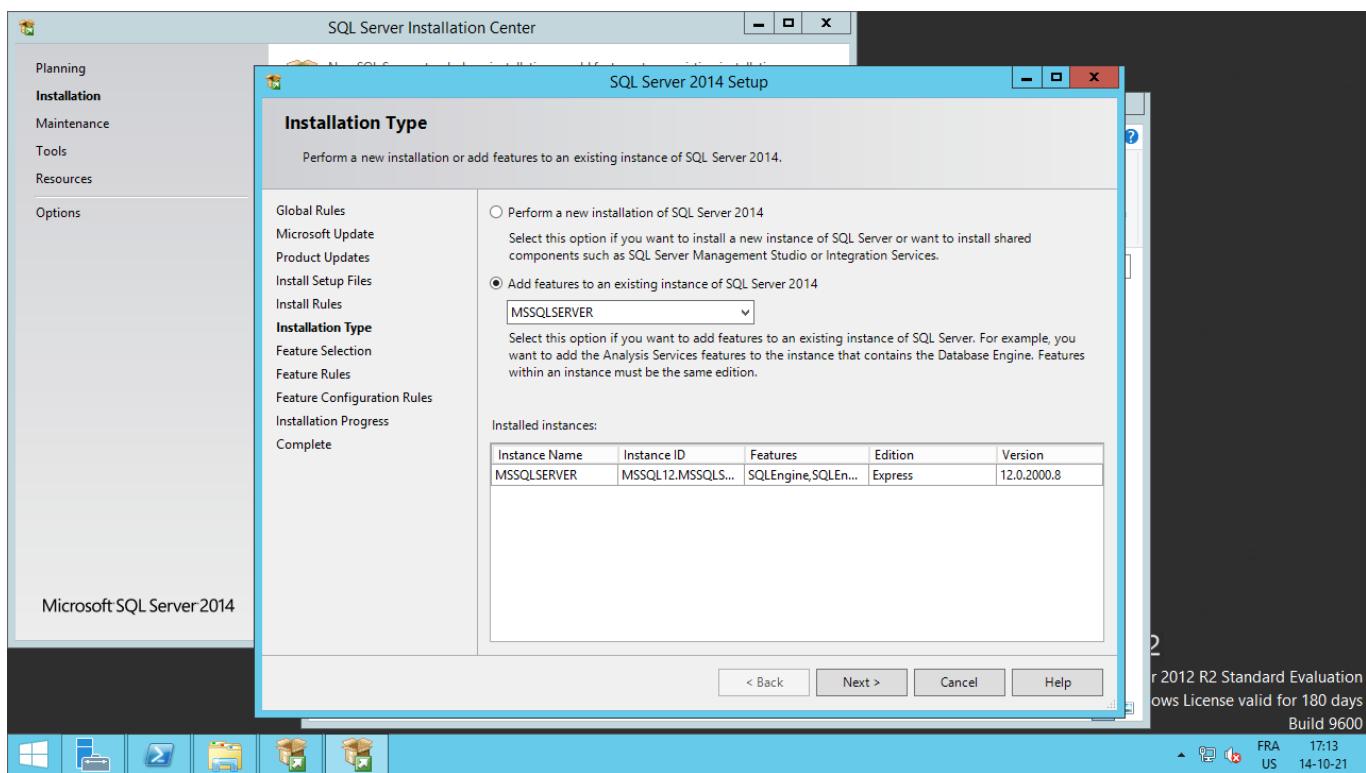
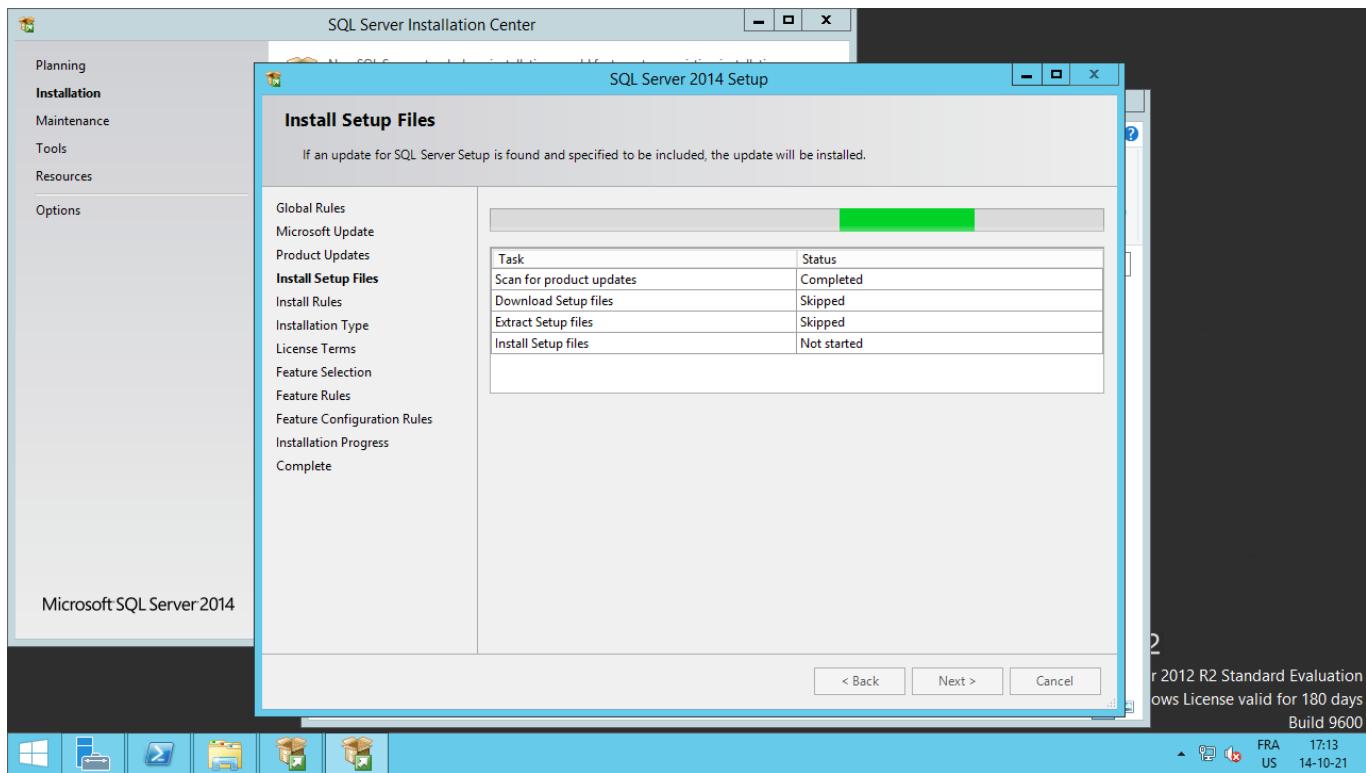
```
$ shasum SQLManagementStudio_x64_ENU-2014.exe
406a6dd55552d35fd99229c4d13e43375f131828 SQLManagementStudio_x64_ENU-2014.exe
```

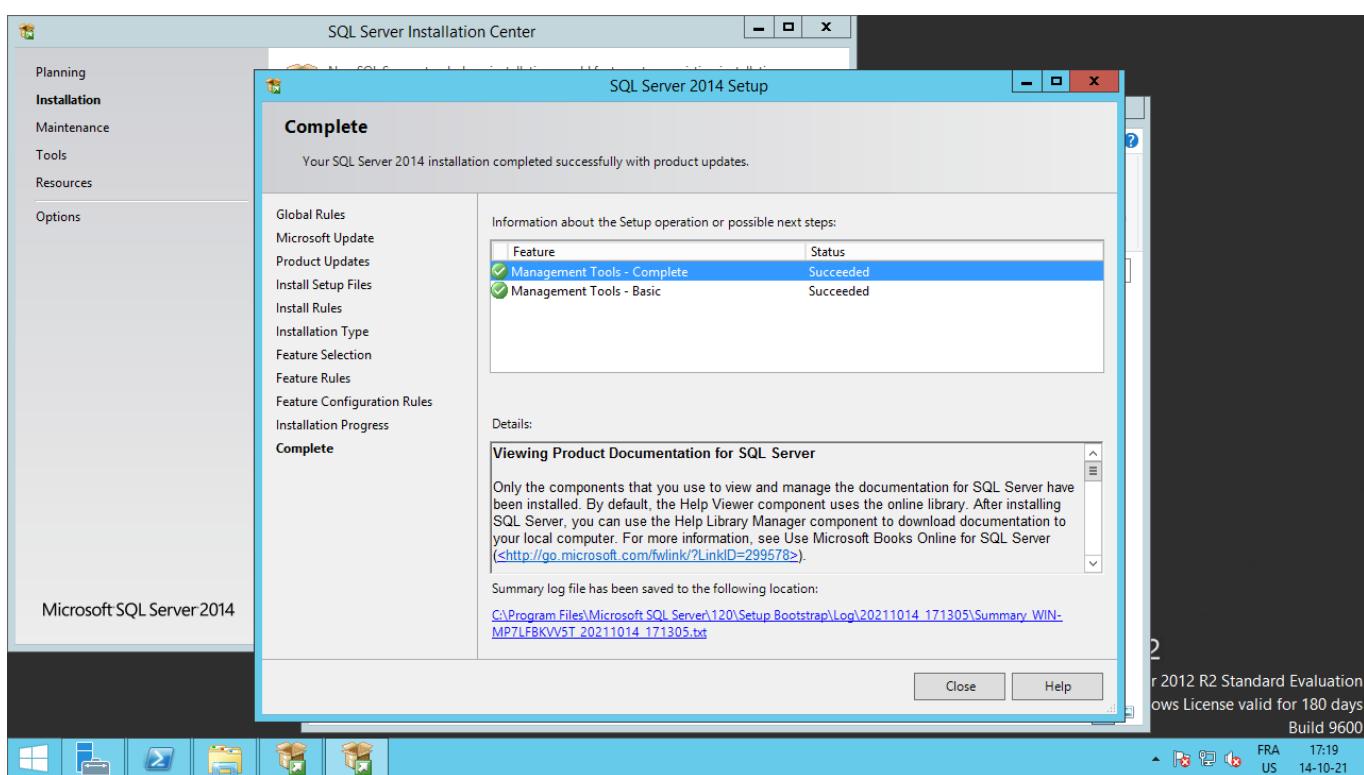
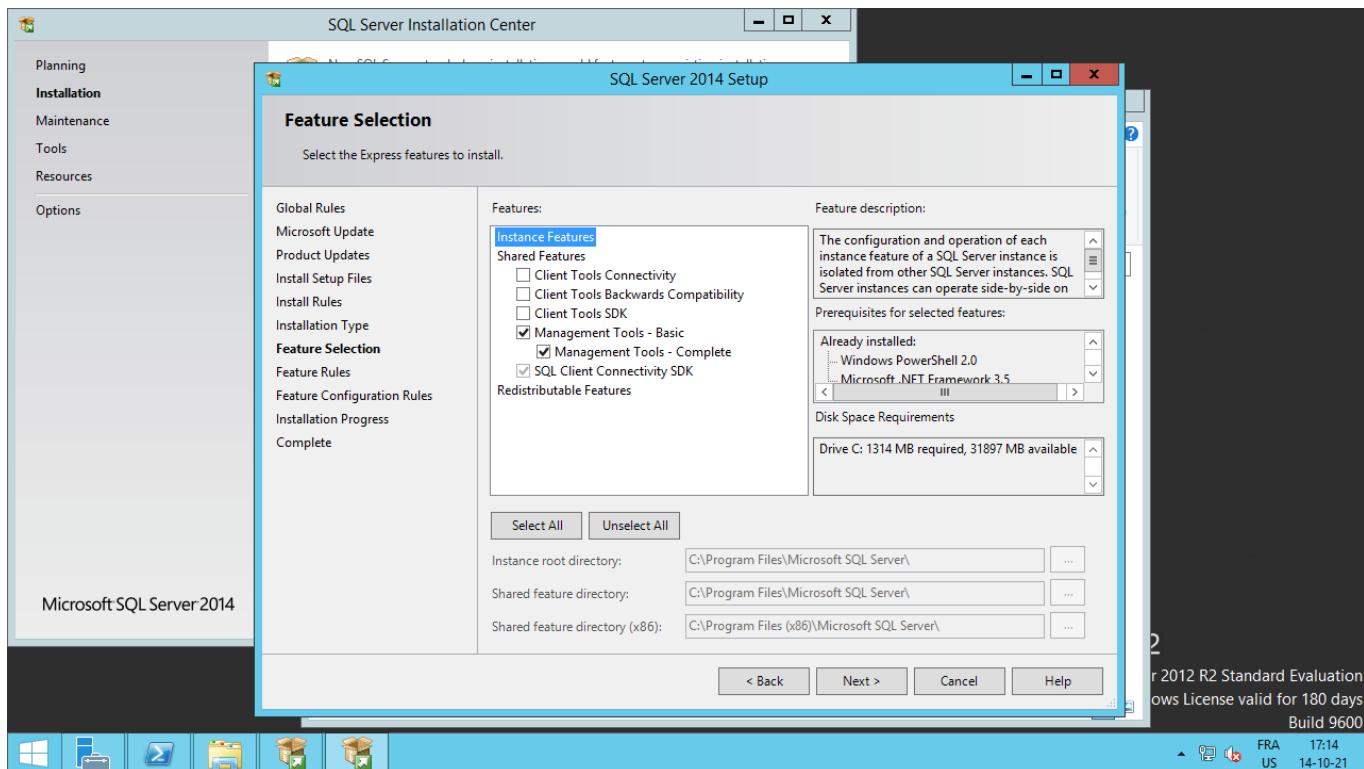
<https://www.microsoft.com/en-us/download/details.aspx?id=42299>

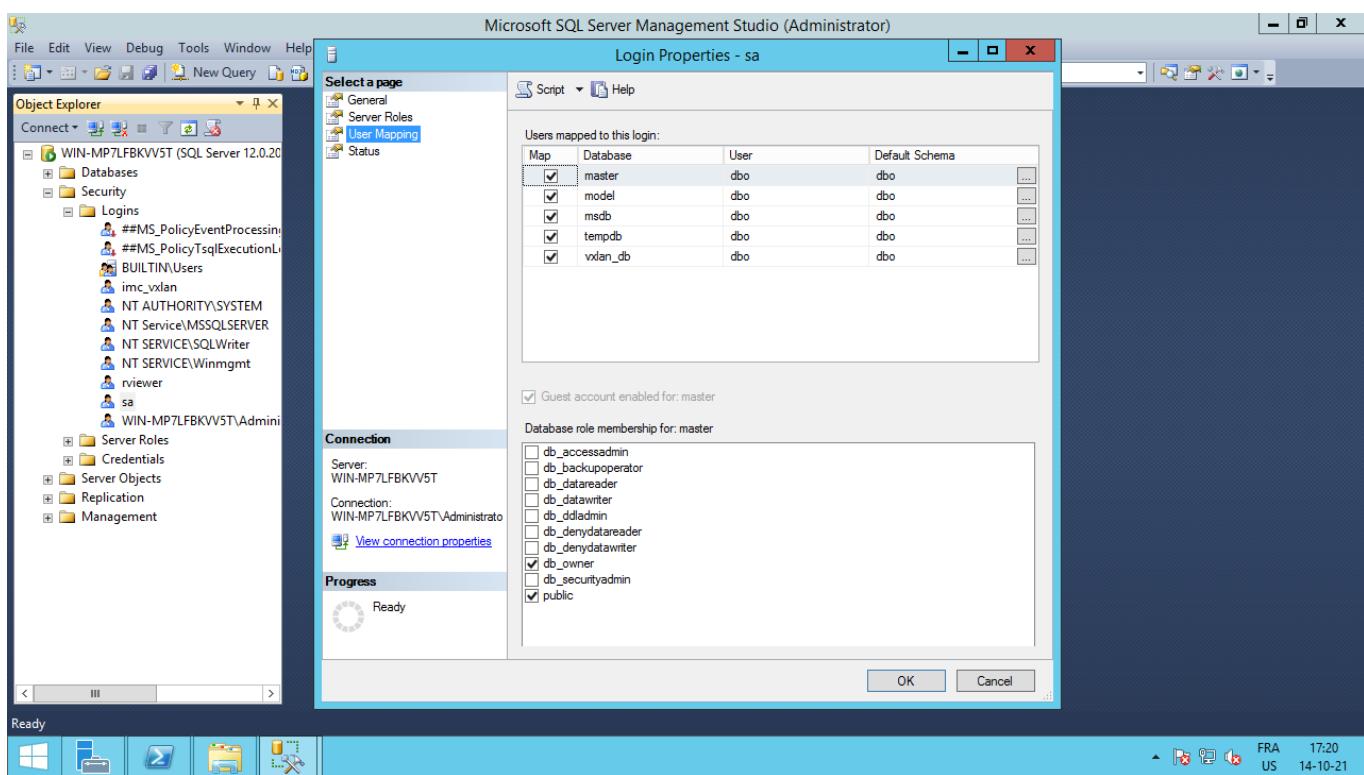
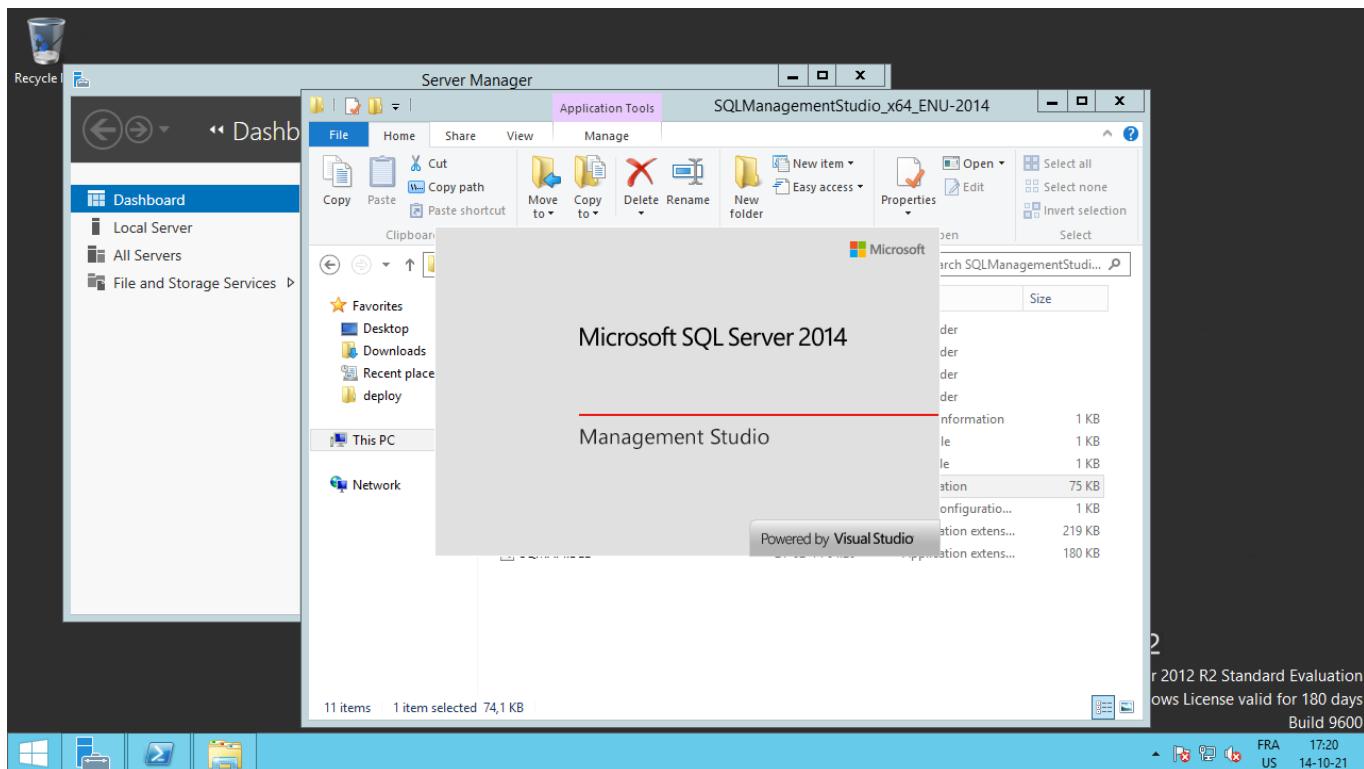
SQLManagementStudio is actually not needed to run iMC but gives the ability to explore the sa account.

Installation steps:









iMC_PLAT_7.3_E0504_Ent_Win-003(vuln) or iMC_PLAT_7.3_E0504P04_Win-002(patched)

The installation of iMC is identical for either version. However, there are discrepancies between the content of both folder/archive.

There at least a folder(named db) that is present in:

iMC_PLAT_7.3_E0504_Ent_Win-003/

which isn't in the other. This is also the case for at least tool/utility like "envcheck.zip".

```
ixidor@pr0c:~/exp/iMC_PLAT_7.3_E0504_Ent_Win-003/windows/install/components/common$ ls -l ; tree db
total 16
drwxrwxr-x 2 ixidor ixidor 4096 Jan 23 2017 client
drwxrwxr-x 2 ixidor ixidor 4096 Jan 23 2017 db
drwxrwxr-x 3 ixidor ixidor 4096 Jan 23 2017 deploy
drwxrwxr-x 2 ixidor ixidor 4096 Jan 23 2017 server
db
├── install32.bat
├── install64.bat
├── SQLExpr_x64_ENU.exe
├── SQLExpr_x86_ENU.exe
├── SQLServer2008R2-KB3045313-x64.exe
├── SQLServer2008R2-KB3045313-x86.exe
├── uninstal32.bat
├── uninstal64.bat
├── update32.bat
└── update64.bat

0 directories, 10 files
ixidor@pr0c:~/exp/iMC_PLAT_7.3_E0504_Ent_Win-003/windows/install/components/common$ ls ../../../tools/
cluster_tools.zip  dhcp-plug-windows.zip iHATool.zip  iMC_DVM.zip  InstallGO  lldp-agent-ubuntu.zip  MSKBSC  SCOM.zip  vrm-plug-linux.zip
dhcp-plug-linux.zip  envcheck.zip  iMC-MIB-Download_Windows.zip  iMC_tools.exe  lldp-agent-redhat.zip  lldp-agent-windows.zip  openstack-plug.zip  VPD  vrm-plug-windows.zip
ixidor@pr0c:~/exp/iMC_PLAT_7.3_E0504_Ent_Win-003/windows/install/components/common$
```

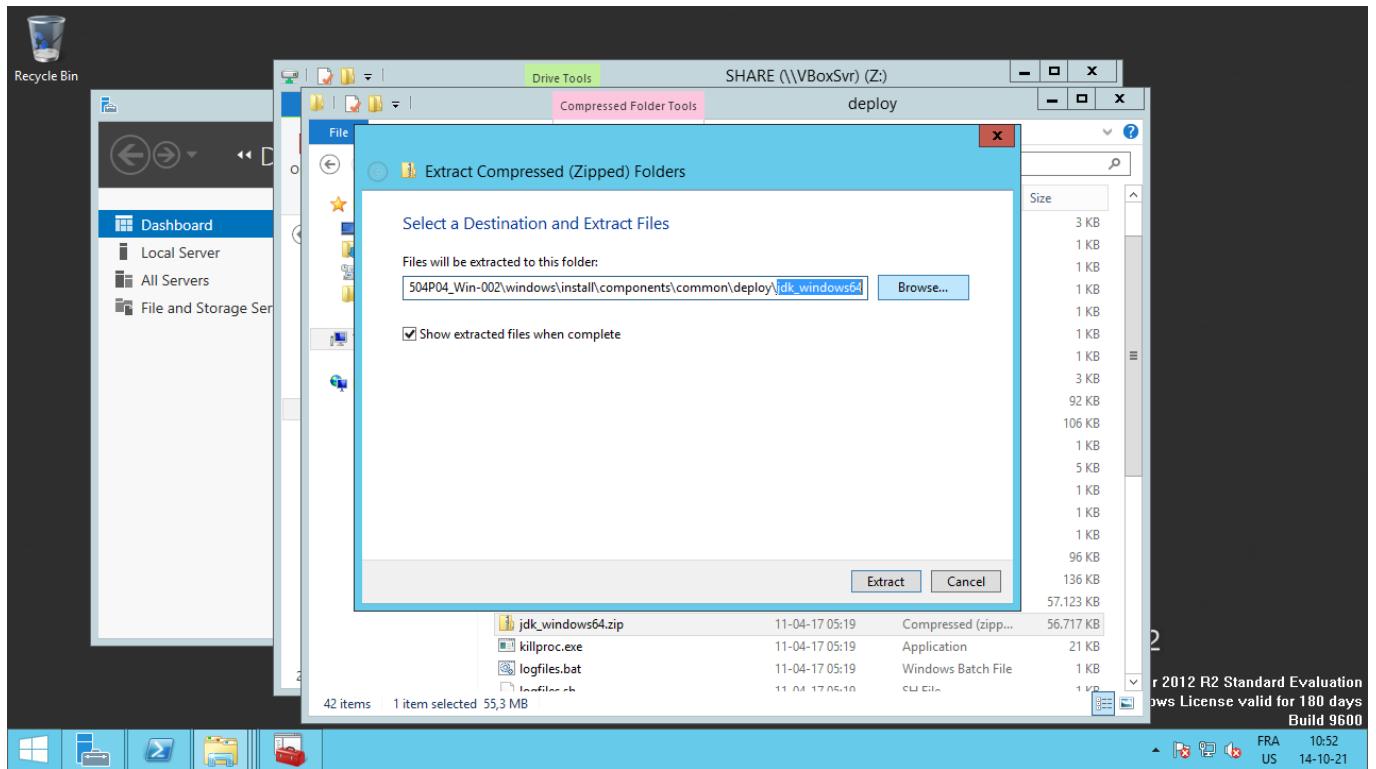


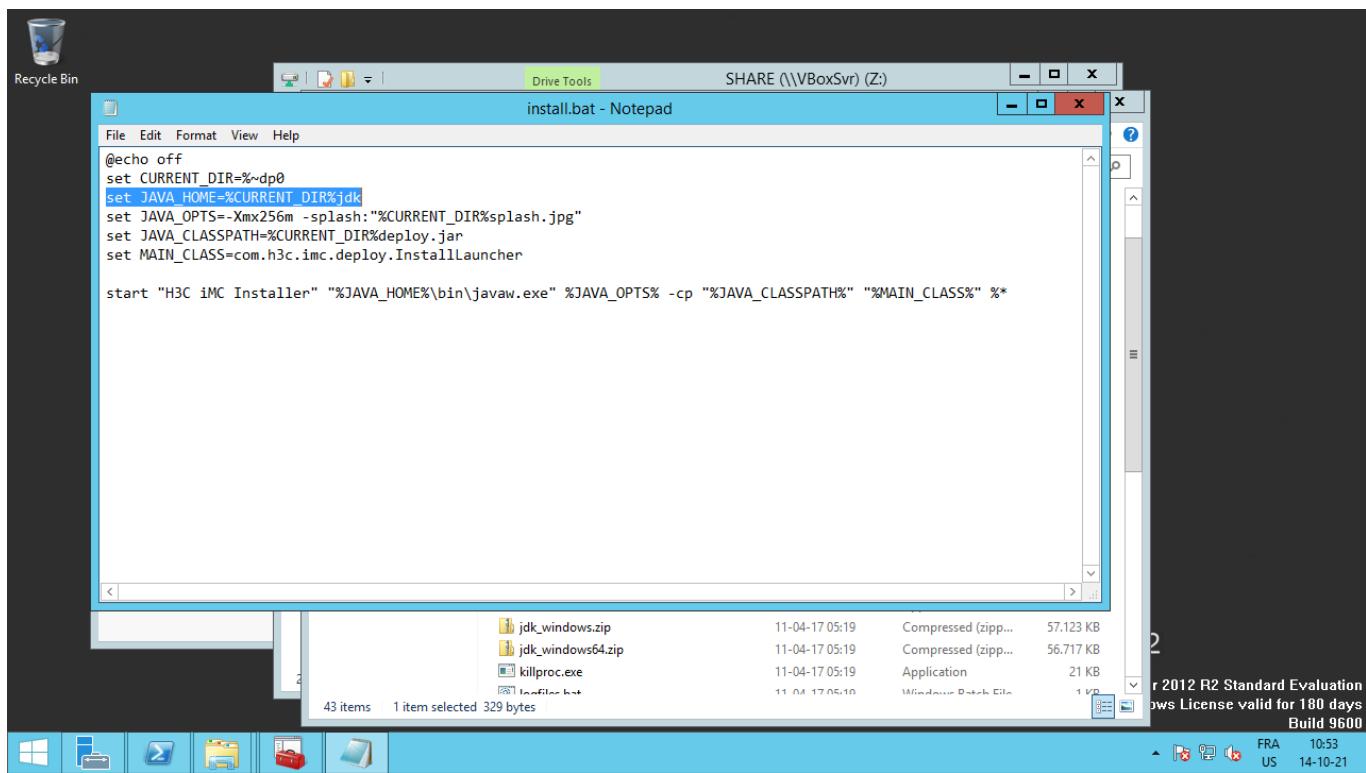
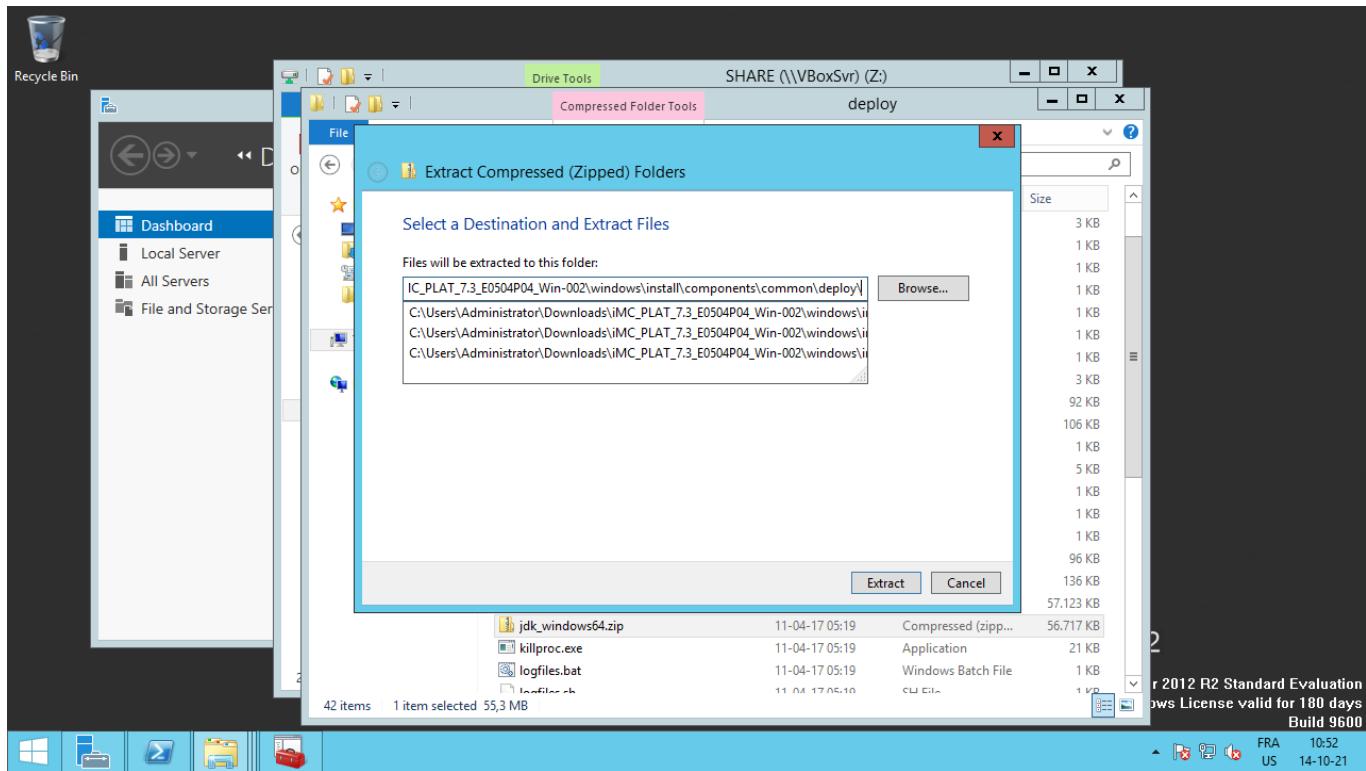
```
ixidor@pr0c:~/exp/iMC_PLAT_7.3_E0504P04_Win-002/windows/install/components/common$ ls -l
total 12
drwxrwxr-x 2 ixidor ixidor 4096 May  2 2017 client
drwxrwxr-x 3 ixidor ixidor 4096 May  2 2017 deploy
drwxrwxr-x 2 ixidor ixidor 4096 May  2 2017 server
ixidor@pr0c:~/exp/iMC_PLAT_7.3_E0504P04_Win-002/windows/install/components/common$ ls ../../../tools/
dhcp-plug-linux.zip  iHATool.zip  iMC-MIB-Download_Windows.zip  lldp-agent-ubuntu.zip  MSKBSC.zip  vrm-plug-linux.zip
dhcp-plug-windows.zip  iMC_MIB_Download_Linux.zip  lldp-agent-redhat.zip  lldp-agent-windows.zip  openstack-plug.zip  vrm-plug-Windows.zip
ixidor@pr0c:~/exp/iMC_PLAT_7.3_E0504P04_Win-002/windows/install/components/common$
```

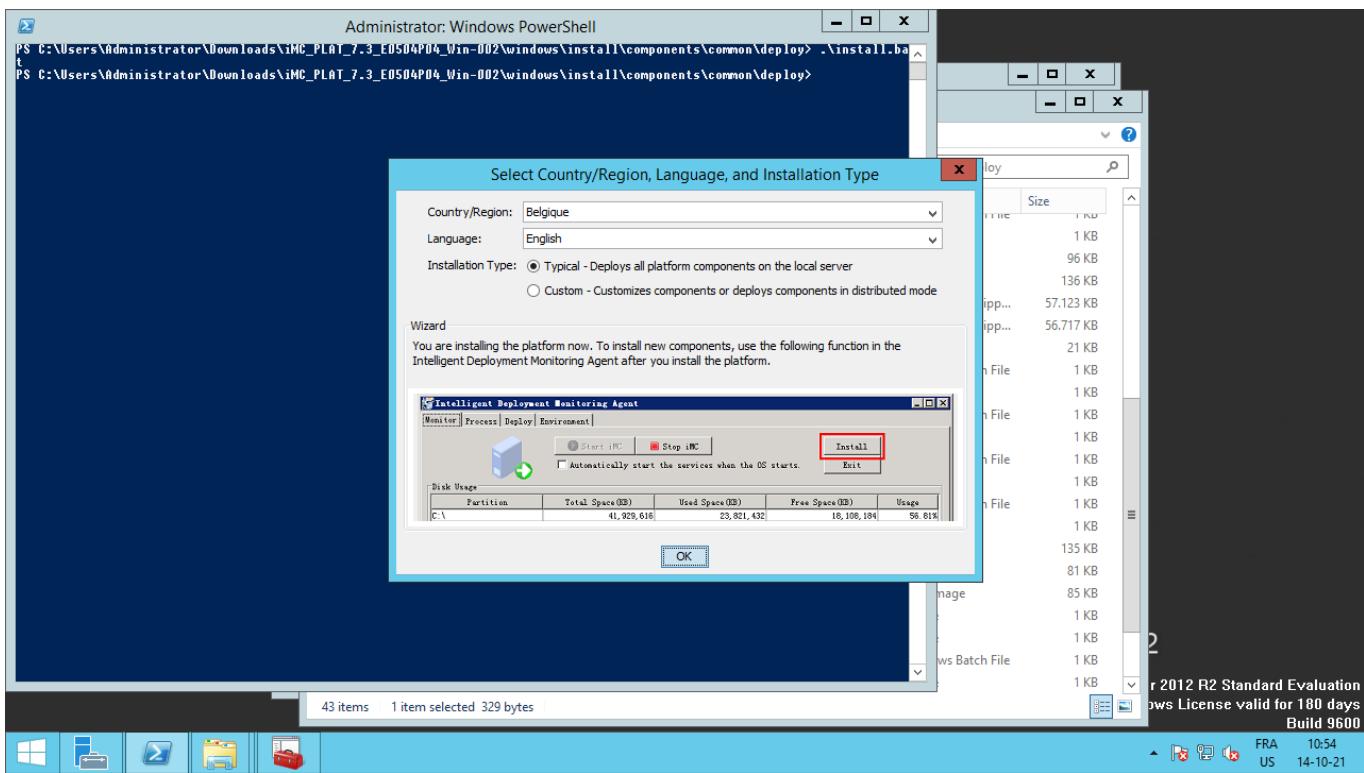
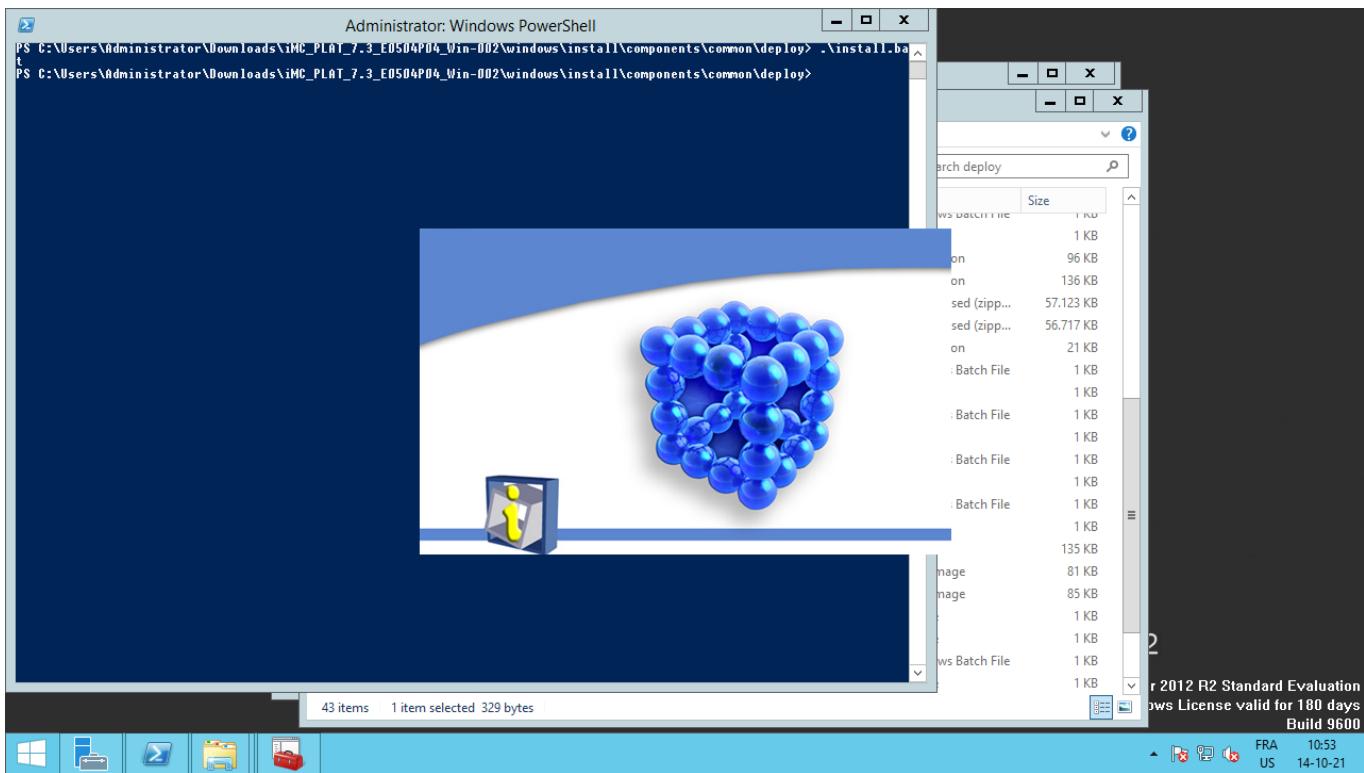
Installation steps:

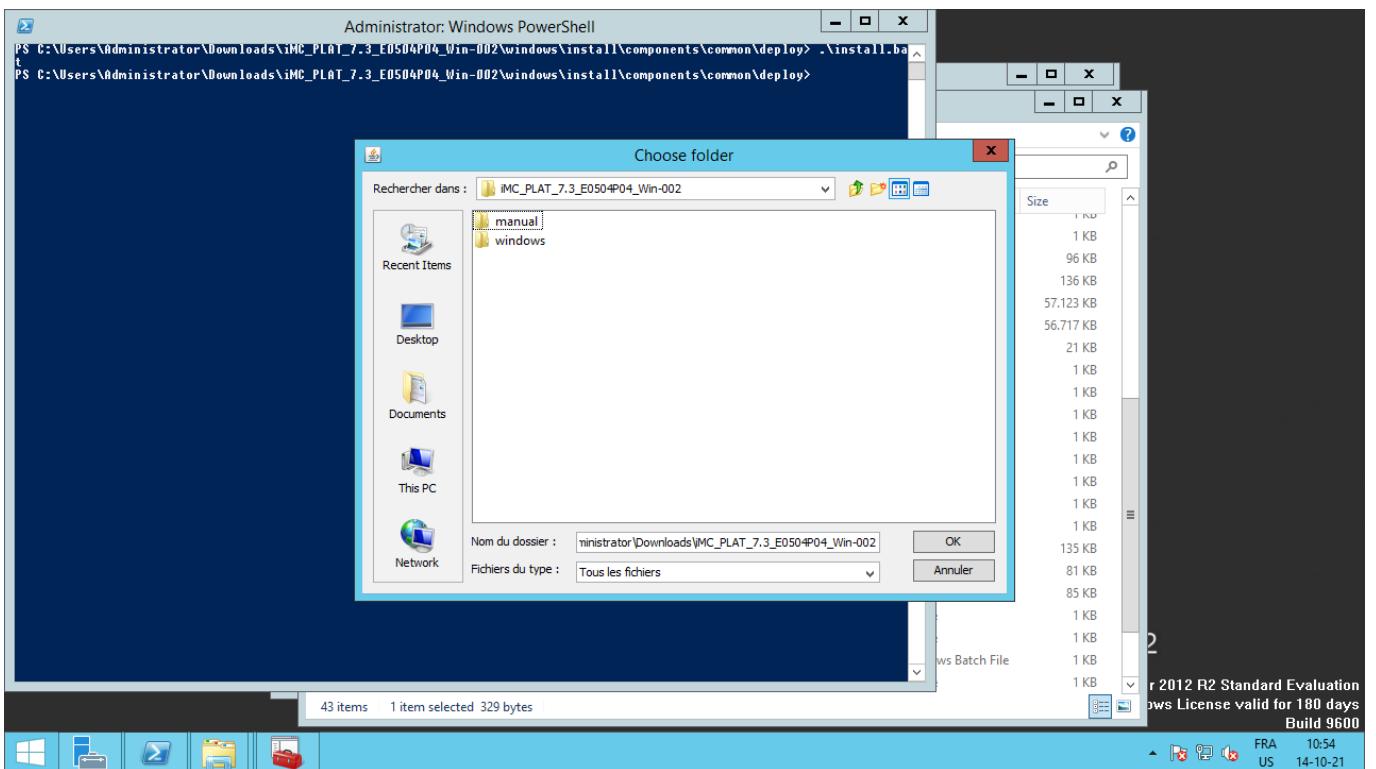
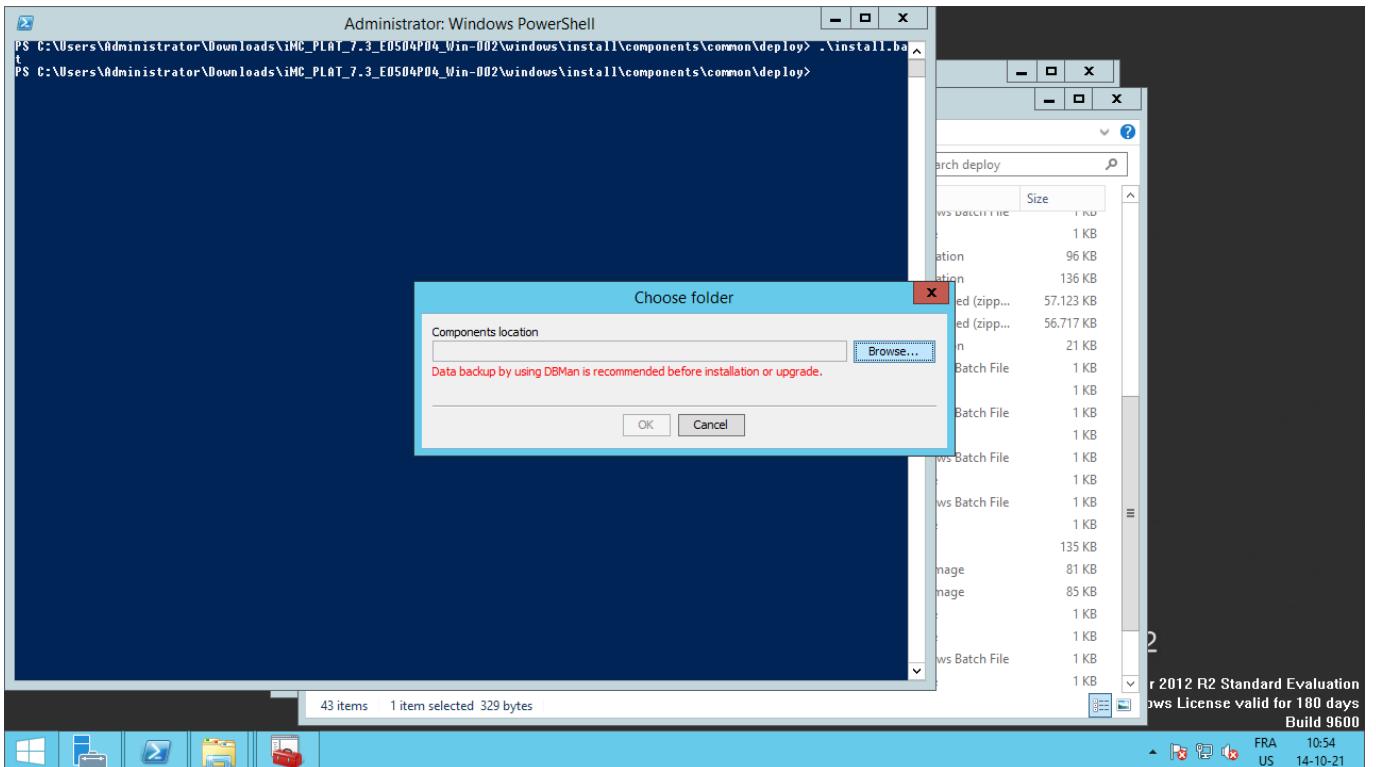
If the previous steps have been followed, there shouldn't be any issues with this point.
The only requirement is to unzip and place the JDK:

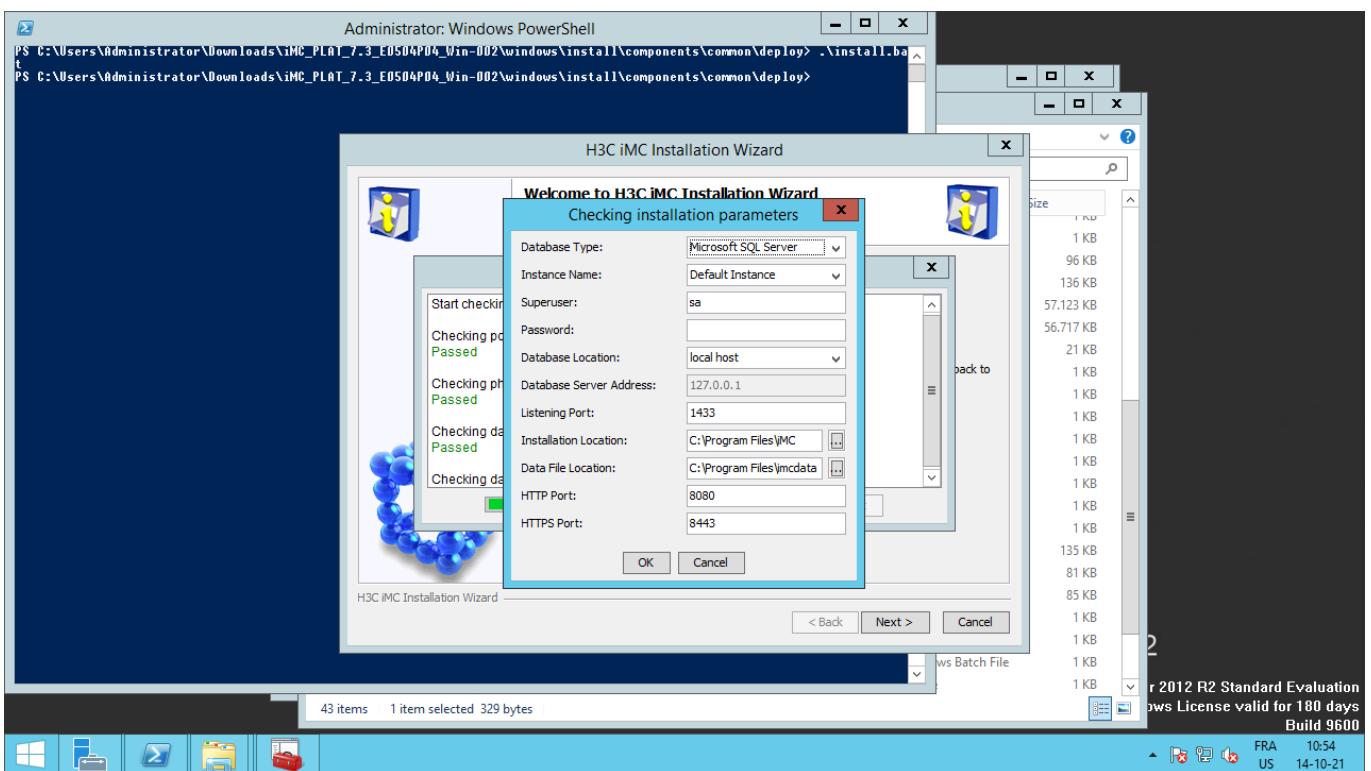
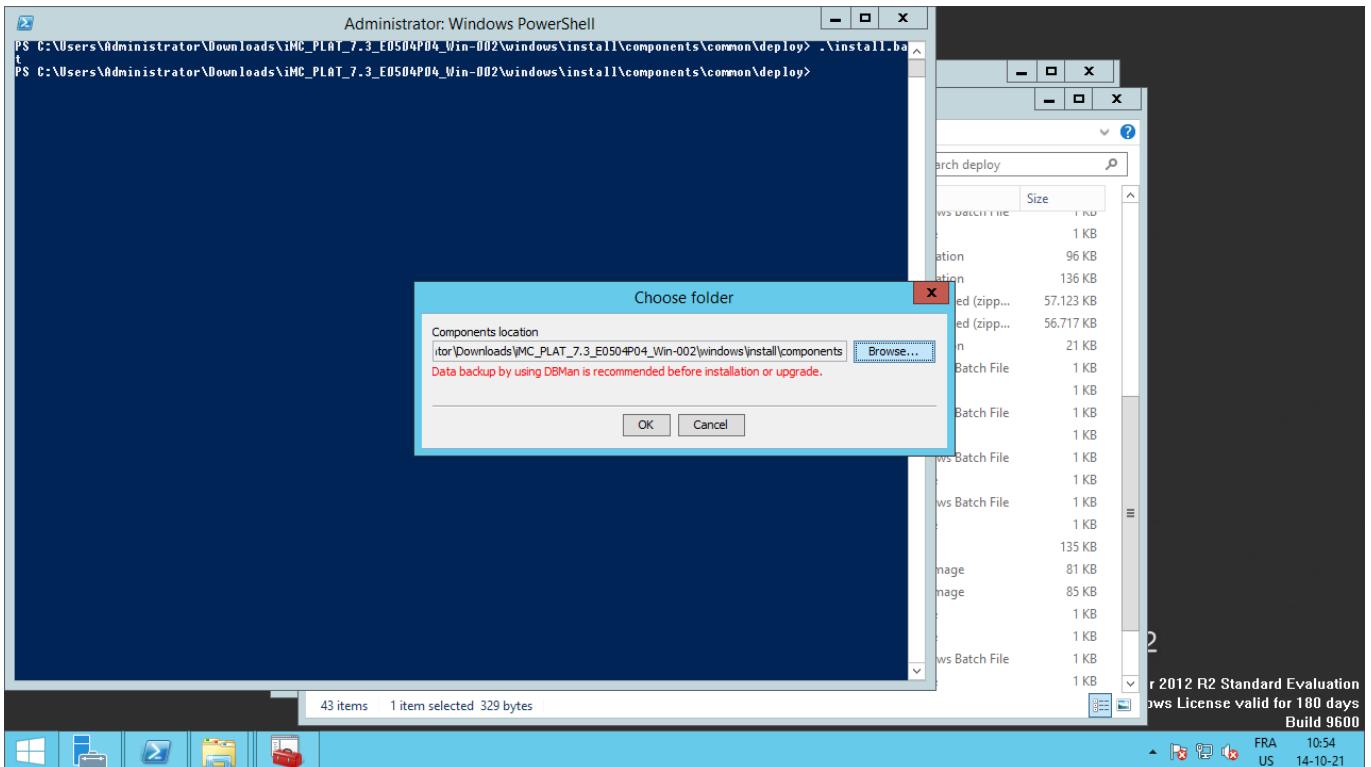
```
C:\Users\Administrator\Downloads\iMC_PLAT_7.3_E0504P04_Win-002\windows\install\components\common\deploy
```



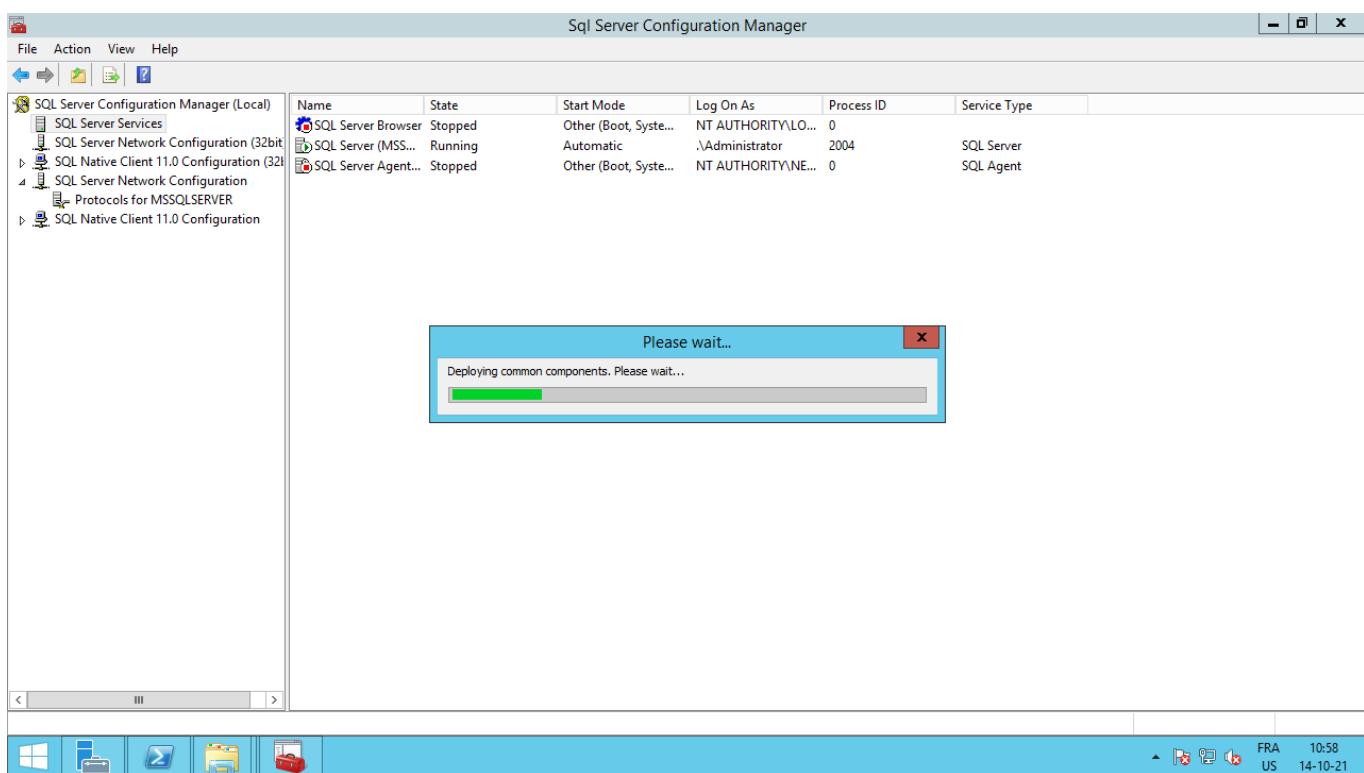
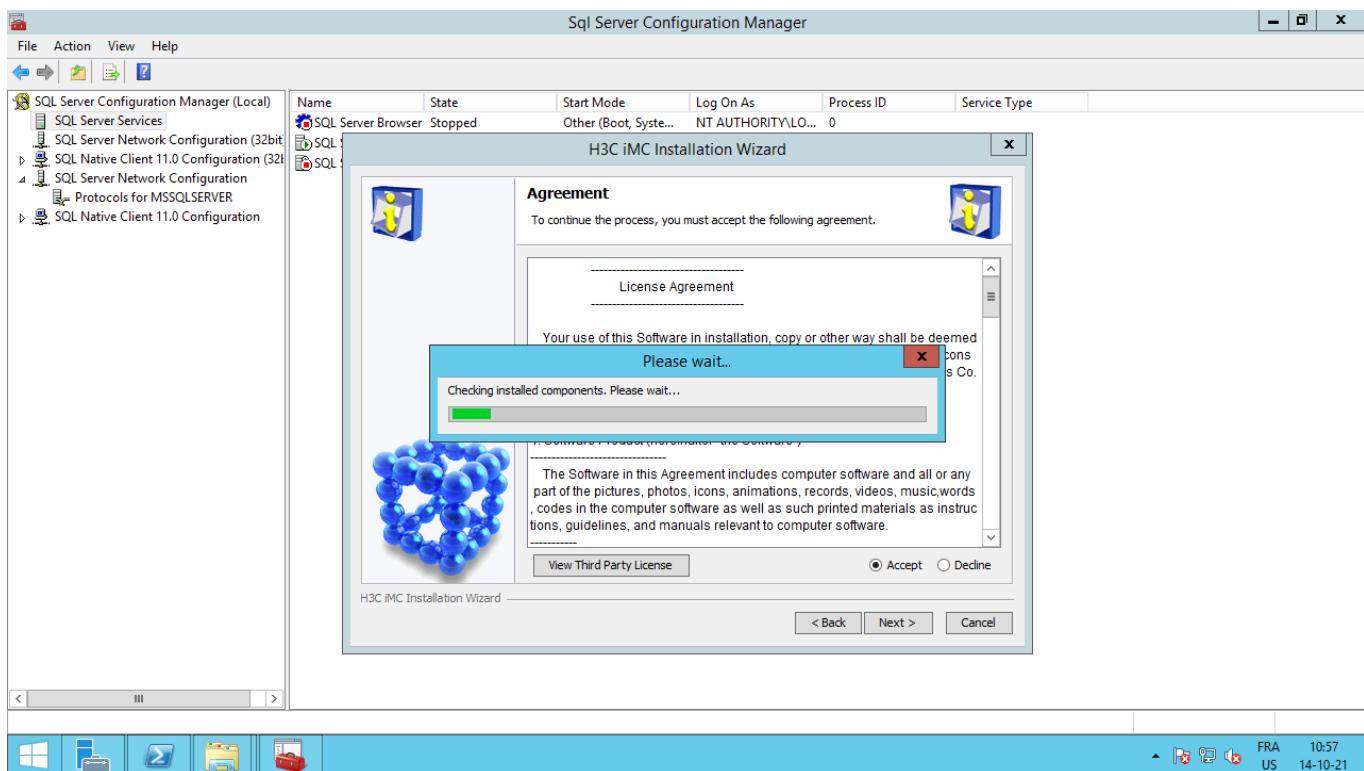




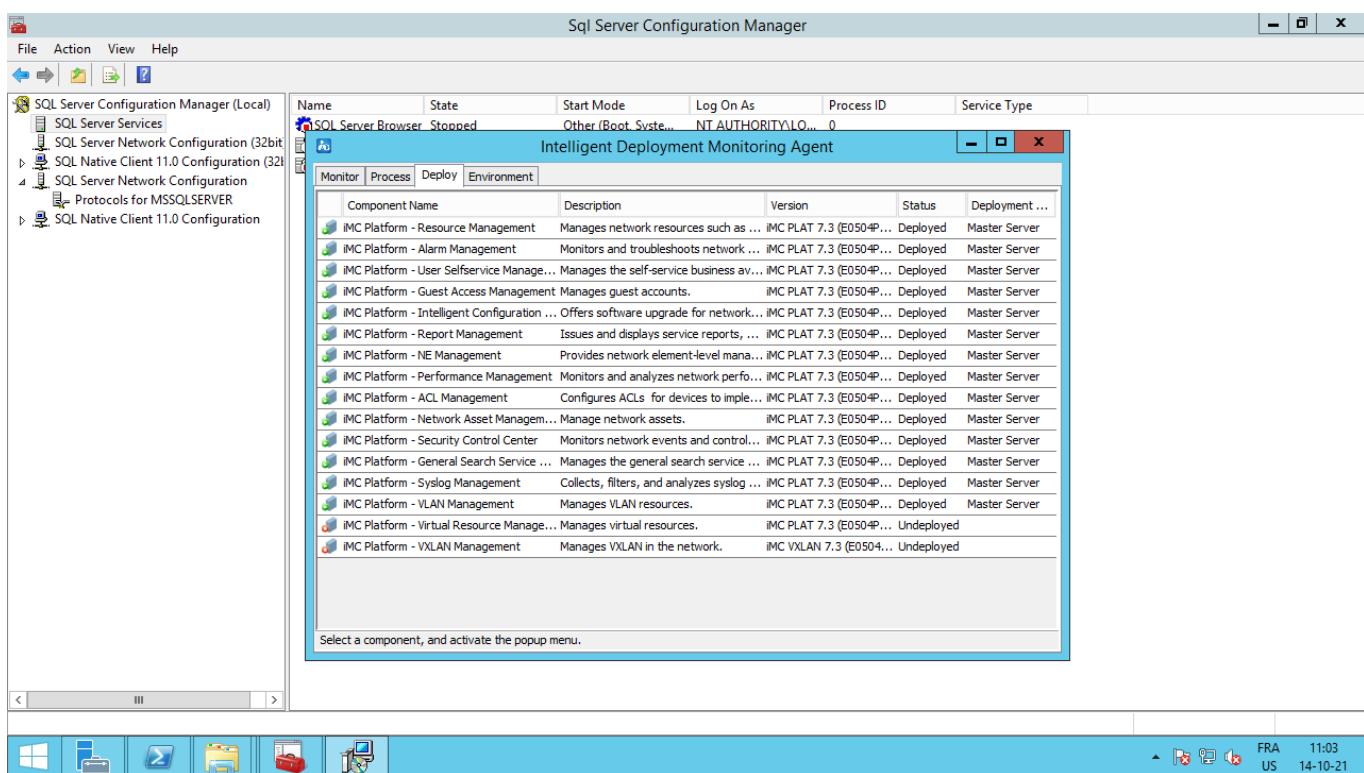
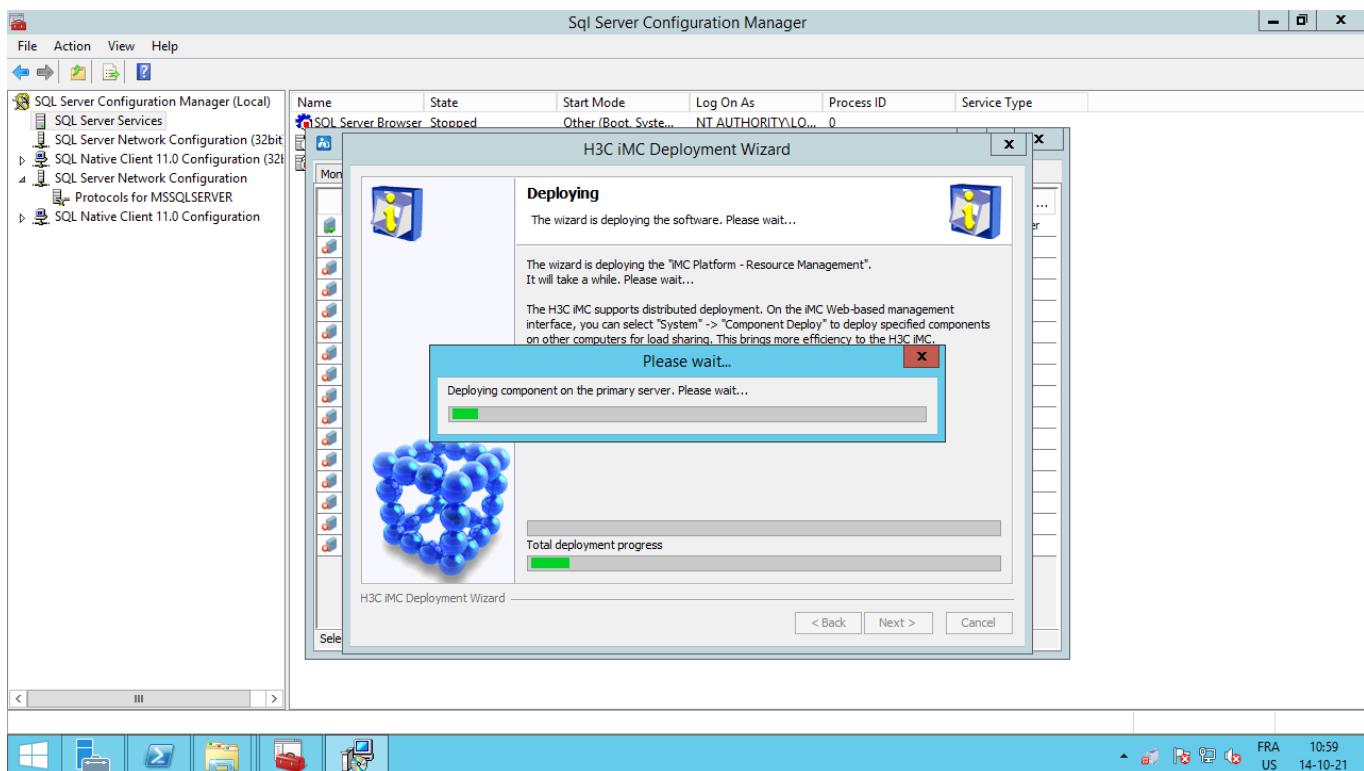


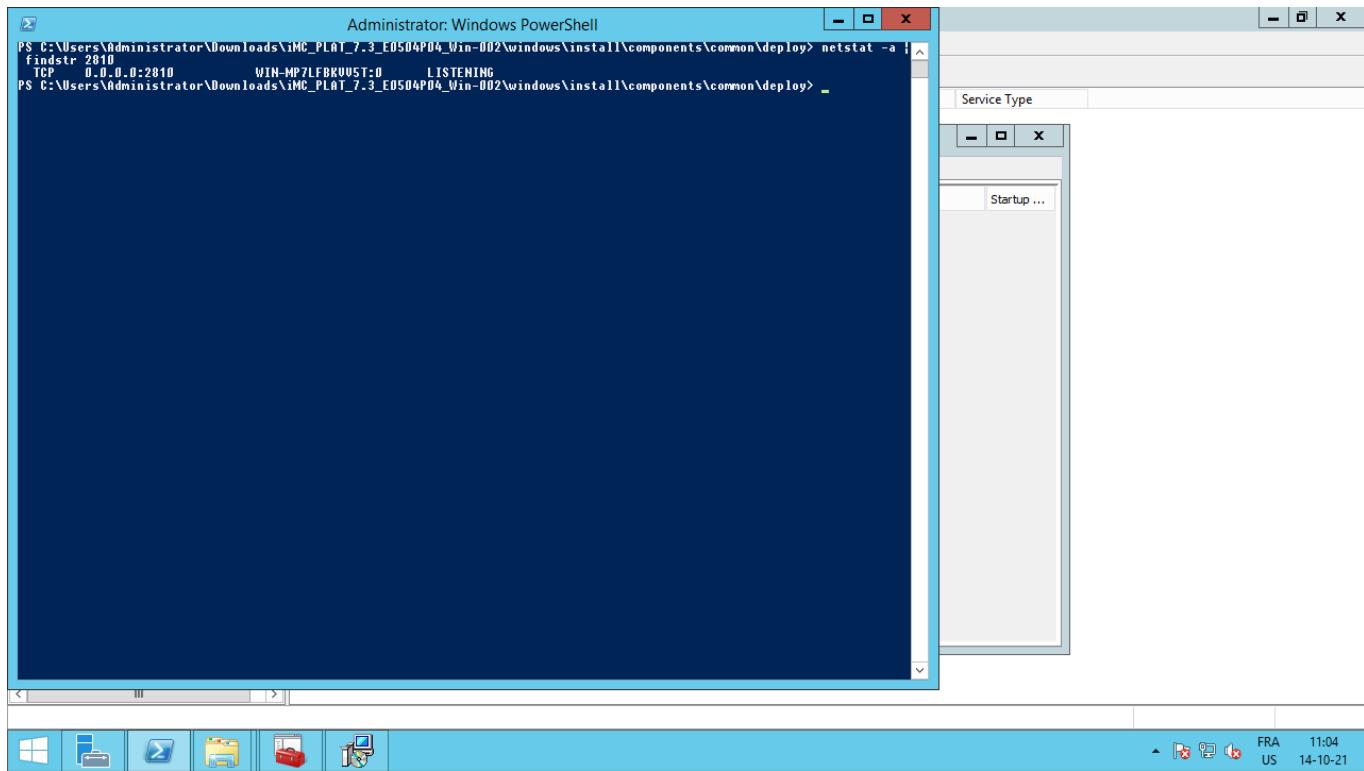
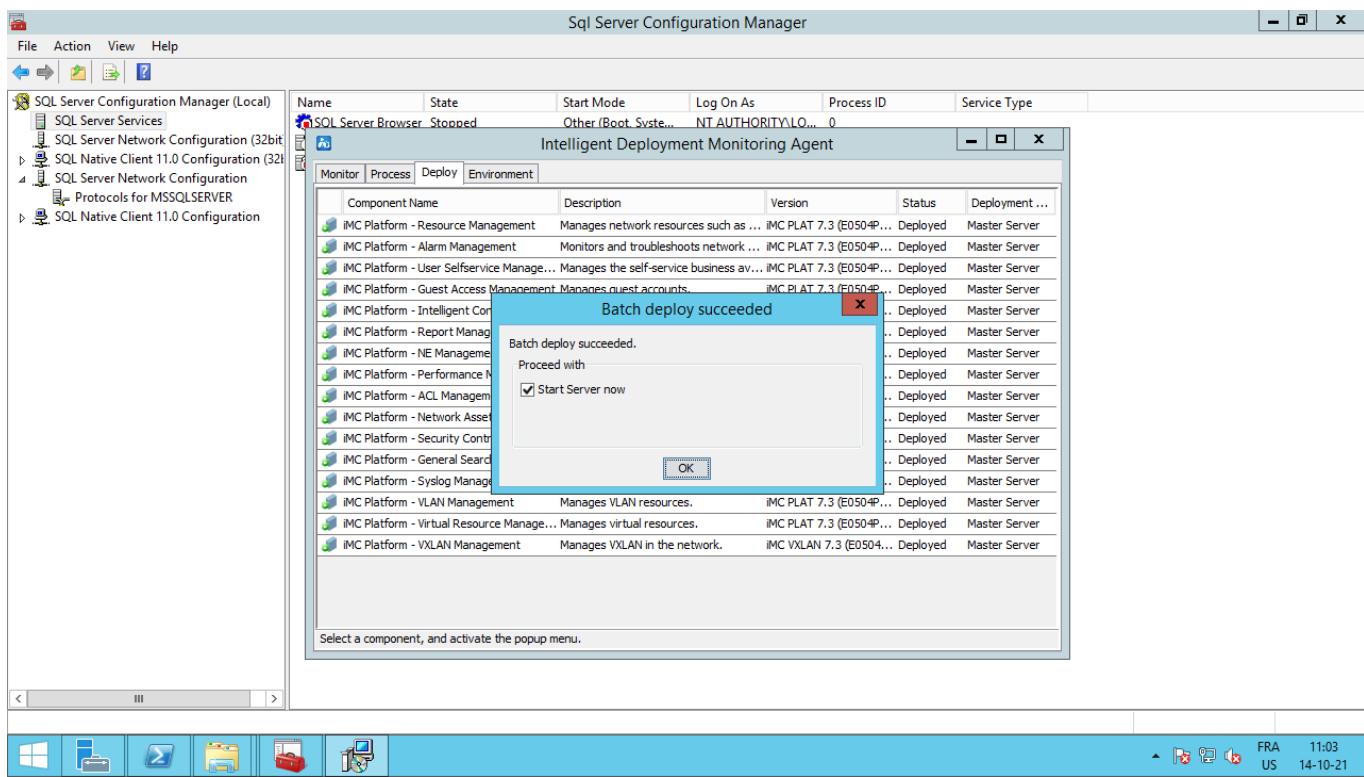


Fill in the password set up at the install of the SQLServer.



Note: the fact that the SQL Server Agent service doesn't run seems to be a normal behaviour because of the db versions is an "express".





We can see "dbman".

Analysis Setup

In order to analyze ZDI-17-836 / CVE-12561 :

<https://www.zerodayinitiative.com/advisories/ZDI-17-836/>

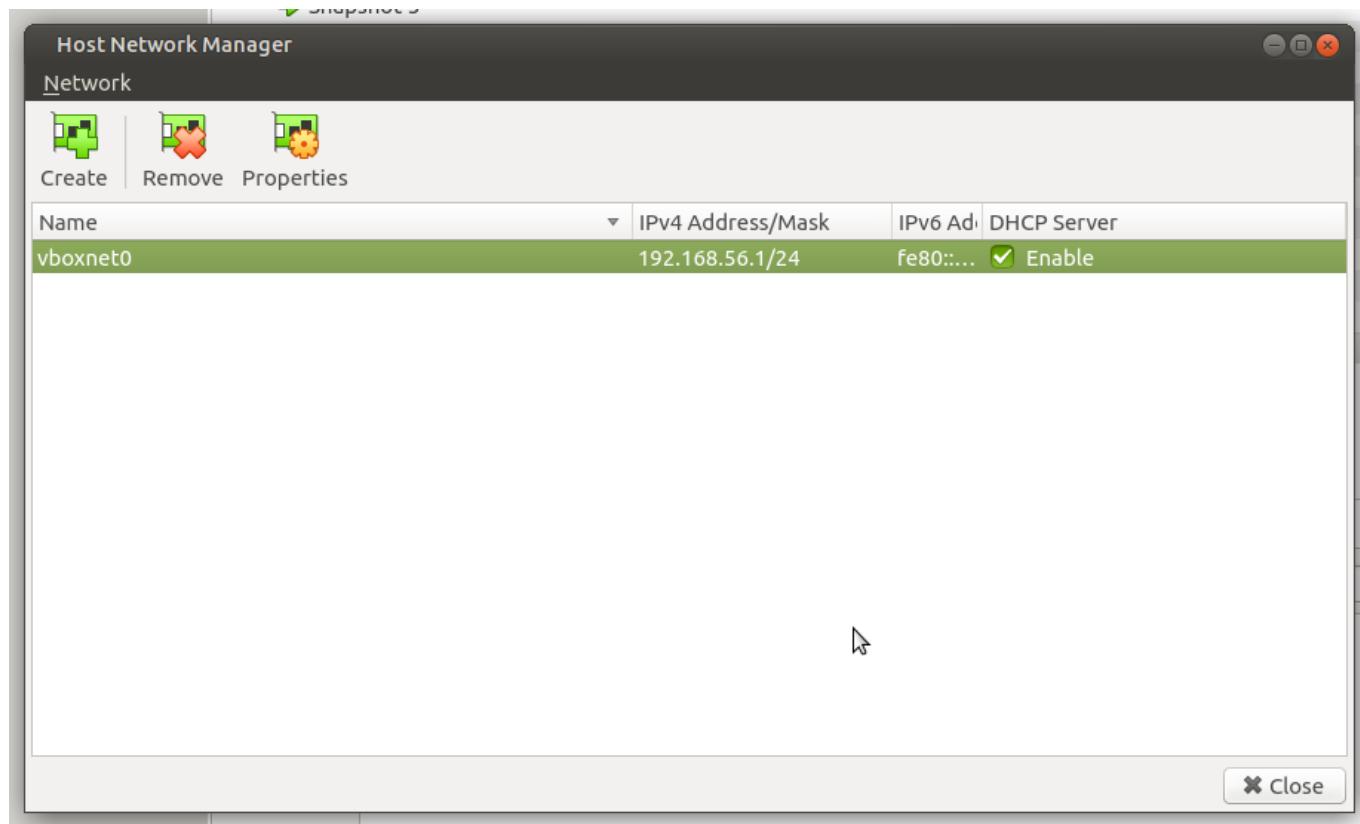
the use of the following software/tools is of great assistance.

Preliminary Virtual Network Configuration

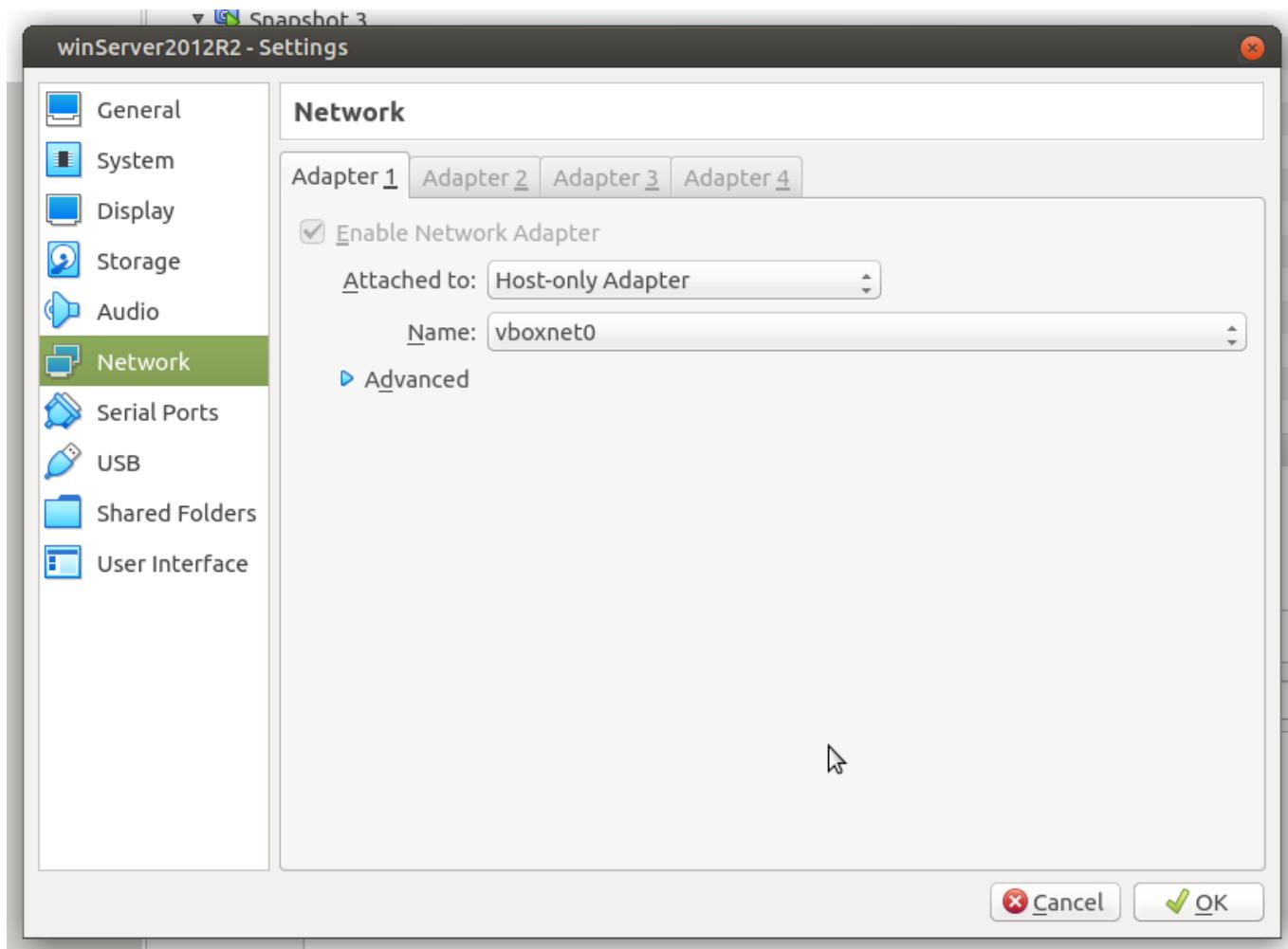
I am going with the following network setup:

To somehow isolate the network of the WinServer2012R2 and potential other instances, I am adding and using "Host-only adapter" for the VMs.

In Oracle VM VirtualBox Manager, under *File, Host Network Manager...*, create an adapter:



Using the created adapter in the concerned VMs:



WindowsServer2012R2

On the iMC hosting machine(WindowsServer2012R2):

I am using the following software/setup:

- WinDbg 10.0.22000.194 + msecdbg
- Ghidra-SRE [ghidra_10.0.4_PUBLIC_20210928.zip](https://ghidra-sre.readthedocs.io/en/latest/)
- xdbg [snapshot_2021-07-01_23-17.zip](https://github.com/0x00000000/xdbg/releases)
- BinDiff7 <https://zynamics.com/software.html>
- ProcessHacker <https://processhacker.sourceforge.io/>
- Wireshark(win64-3.4.9) <https://www.wireshark.org/>
- ...

WinDbg Install

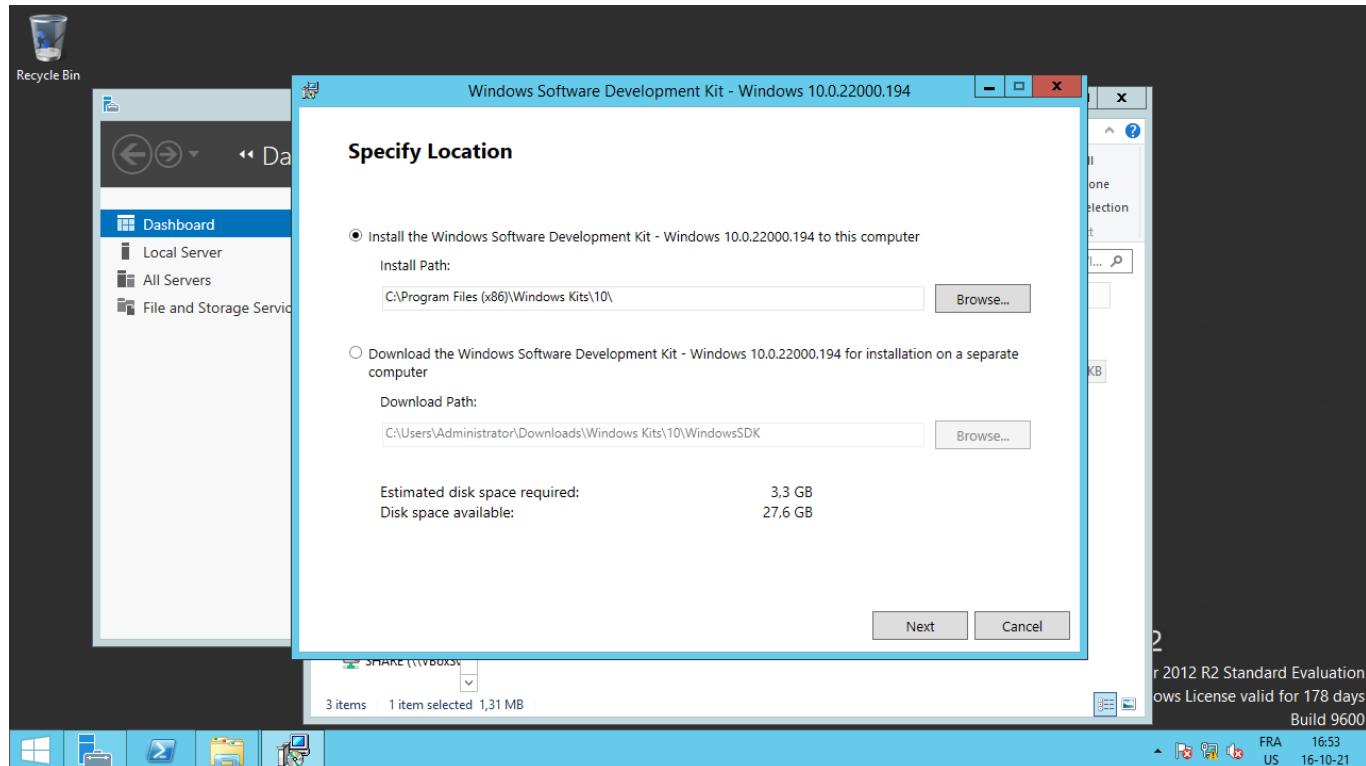
I am going for the iso version installer:

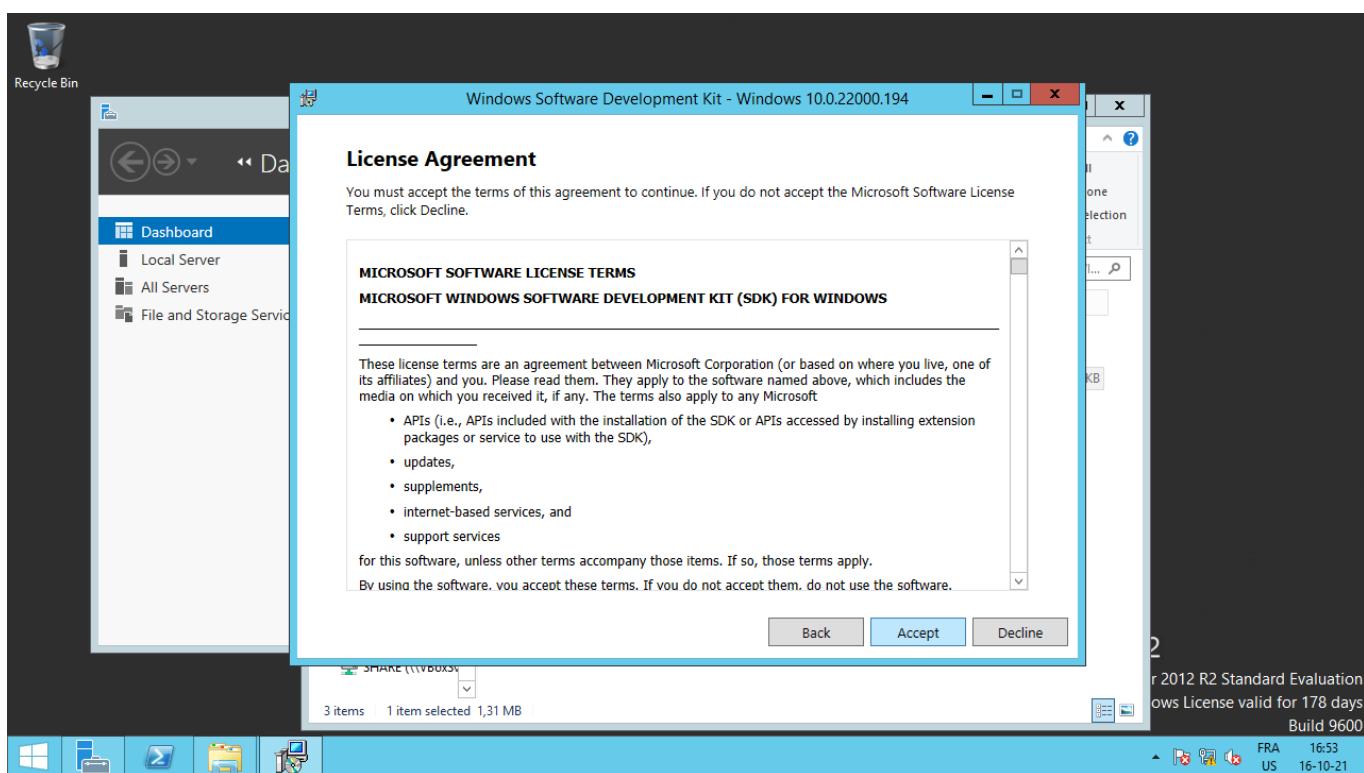
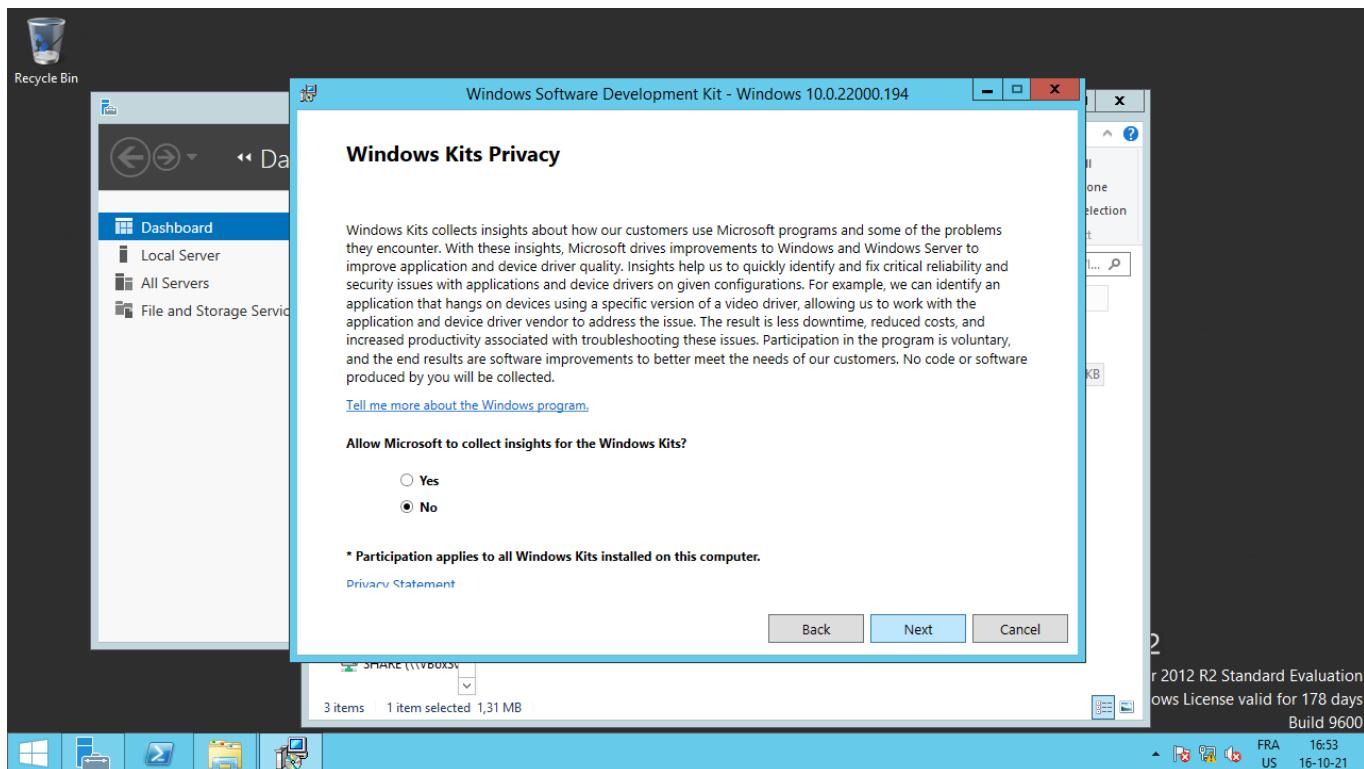
<https://developer.microsoft.com/en-us/windows/downloads/windows-sdk/>

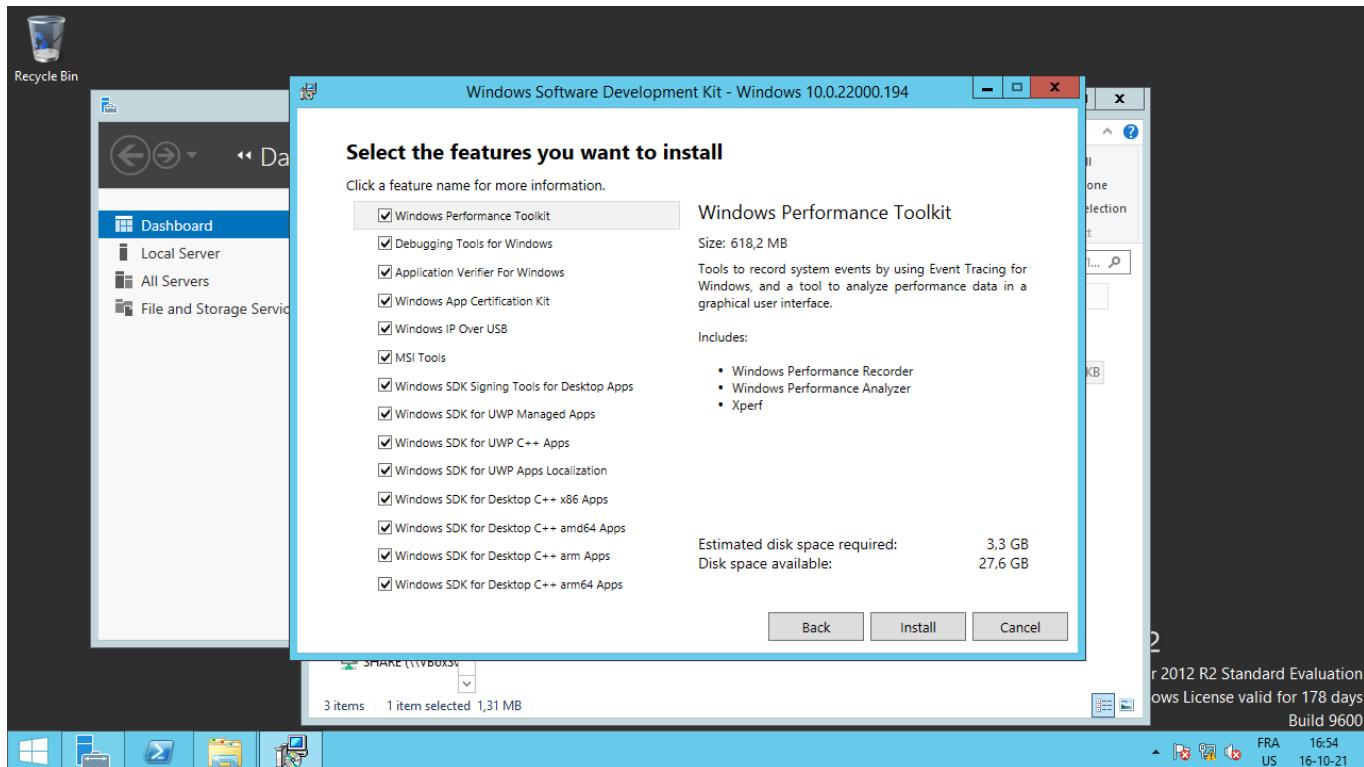
```
$ shasum 22000.194.210911-1543.co_release_svc_prod1_WindowsSDK.iso  
b875d4a935b08493a0e35f523010854abf2884c3 22000.194.210911-  
1543.co_release_svc_prod1_WindowsSDK.iso
```

To install through the iso:

- right click on the iso, then select *Mount*
- access the disk through the Windows Explorer and launch the installation







Msecdbg extensions

The **msecdbg** winDbg extensions helps one in assessing if an observed behaviour in the debugger(generally a crash) is exploitable or not.

The following is the official link:

<https://archive.codeplex.com/?p=msecdbg>

I am going for a compiled version, here:

<https://blog.didierstevens.com/2018/07/17/exploitable-crash-analyzer-statically-linked-crt/>

<https://didierstevens.com/files/software/MSECWinDbgExtensions.zip>

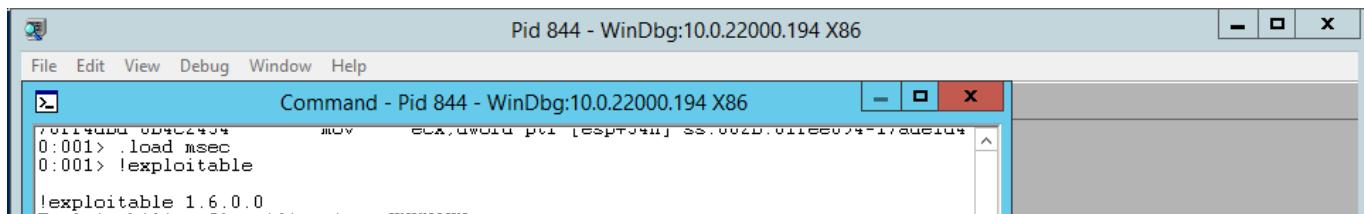
```
$ shasum MSECWinDbgExtensions.zip
63741f82e4da2b6762776b7bcf717c4497c05cca MSECWinDbgExtensions.zip
```

Installation:

The installation consists in placing the *MSEC.dll*(selecting the one corresponding to your ISA) in the following directory:

```
C:\Program Files (x86)\Windows Kits\10\Debuggers\x86\winext
```

The msecdbg extension can be loaded as follow:



Wireshark

The installation is pretty much straightforward, just need to go for the default configuration.

<https://www.wireshark.org/download.html>

```
$ sha1sum Wireshark-win64-3.4.9.exe
fea985c482130a2f92b04e2e3f8bbeb8815c3f5f  Wireshark-win64-3.4.9.exe
```

Kali 2021.3

I am using a Kali 2021.3 VM for a quick access to additional tools.

Kali version:

```
└─(kali㉿kali)-[~]
└─$ lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2021.3
Codename:      kali-rolling
```

Ghidra-SRE

Ghidra 10.0.4:

```
$ sha256sum ghidra_10.0.4_PUBLIC_20210928.zip
1ce9bdf2d7f6bdfe5dccd06da828af31bc74acfd800f71ade021d5211e820d5e
ghidra_10.0.4_PUBLIC_20210928.zip
```

Written mostly in Java, the install comes down to decompressing the archive and make the JDK visible to it.

[Installation Guide](#)

BinDiff 7

```
$ sha1sum bindiff7.msi  
95550a1539eabc8f91066210c815ac756eaa288f bindiff7.msi
```

<https://zynamics.com/software.html>

For nice graph visualization, using BinExport for Ghidra plugin, we can export the files(.BinExport) that can be fed into BinDiff.

<https://github.com/google/binexport>

Main reference for this setup:

<https://ihack4falafel.github.io/Patch-Diffing-with-Ghidra/>

Root Cause Analysis(RCA)

Dbman.exe

From <https://www.zerodayinitiative.com/advisories/ZDI-17-836/> :

This vulnerability allows remote attackers to execute arbitrary code on vulnerable installations of Hewlett Packard Enterprise Intelligent Management Center. Authentication is not required to exploit this vulnerability.

The specific flaw exists within dbman service, which listens on TCP port 2810 by default. A crafted opcode 10012 message can cause a pointer to be reused after it has been freed. An attacker can leverage this vulnerability to execute code under the context of SYSTEM.

We are interested in *dbman.exe*, which is a utility that deals with database related operations.

After the iMC's installation *dbman.exe* is located at:

The screenshot shows a Windows Server 2012 R2 desktop environment. In the foreground, a PowerShell window titled "Administrator: Windows PowerShell" is open, displaying the command "PS C:\Program Files> gci -Recurse -Filter "dbman*" -ErrorAction SilentlyContinue -Path "C:\Program Files"" and its output. The output shows the file "dbman.exe" located in the directory "C:\Program Files\iMC\dbman\bin". The taskbar at the bottom shows several icons, including the Start button, File Explorer, Task View, and Task Manager. The system tray indicates the location is FRA, the user is US, the date is 15-10-21, and the time is 15:43.

```

Administrator: Windows PowerShell
PS C:\Program Files> gci -Recurse -Filter "dbman*" -ErrorAction SilentlyContinue -Path "C:\Program Files"
Directory: C:\Program Files\iMC
Mode LastWriteTime Length Name
---- -- -- -- -- 
d--- 14-10-21 10:58 dbman

Directory: C:\Program Files\iMC\dbman\bin
Mode LastWriteTime Length Name
-- -- -- -- -- 
-a-- 14-10-21 10:58 857600 dbman.exe

Directory: C:\Program Files\iMC\dbman\etc
Mode LastWriteTime Length Name
-- -- -- -- -- 
-a-- 14-10-21 10:58 1638 dbman.conf

Directory: C:\Program Files\iMC\dbman\log
Mode LastWriteTime Length Name
-- -- -- -- -- 
-a-- 14-10-21 10:59 0 dbman.log
-a-- 14-10-21 18:00 53521 dbman_debug.log

PS C:\Program Files>

```

In this setting, *dbman.exe* binary lies in:

```
C:\Program Files\iMC\dbman\bin
```

dbman.exe versions hashes:

```

ixidor@pr0c:~/SHARE/dbman-iMC_PLAT_7.3_E0504_Ent_Win-003/dbman/bin$ md5sum dbman.exe
c0a5cd15339a8eda718886510a347ce8 dbman.exe
ixidor@pr0c:~/SHARE/dbman-iMC_PLAT_7.3_E0504_Ent_Win-003/dbman/bin$ cd ../../dbman-iMC_PLAT_7.3_E0504P04_Win-002/
ixidor@pr0c:~/SHARE/dbman-iMC_PLAT_7.3_E0504P04_Win-002$ md5sum dbman/bin/dbman.exe
3cd632d4245d08e76014e7240e5f5f82 dbman/bin/dbman.exe
ixidor@pr0c:~/SHARE/dbman-iMC_PLAT_7.3_E0504P04_Win-002$ 

```

PE File Security Mitigations:

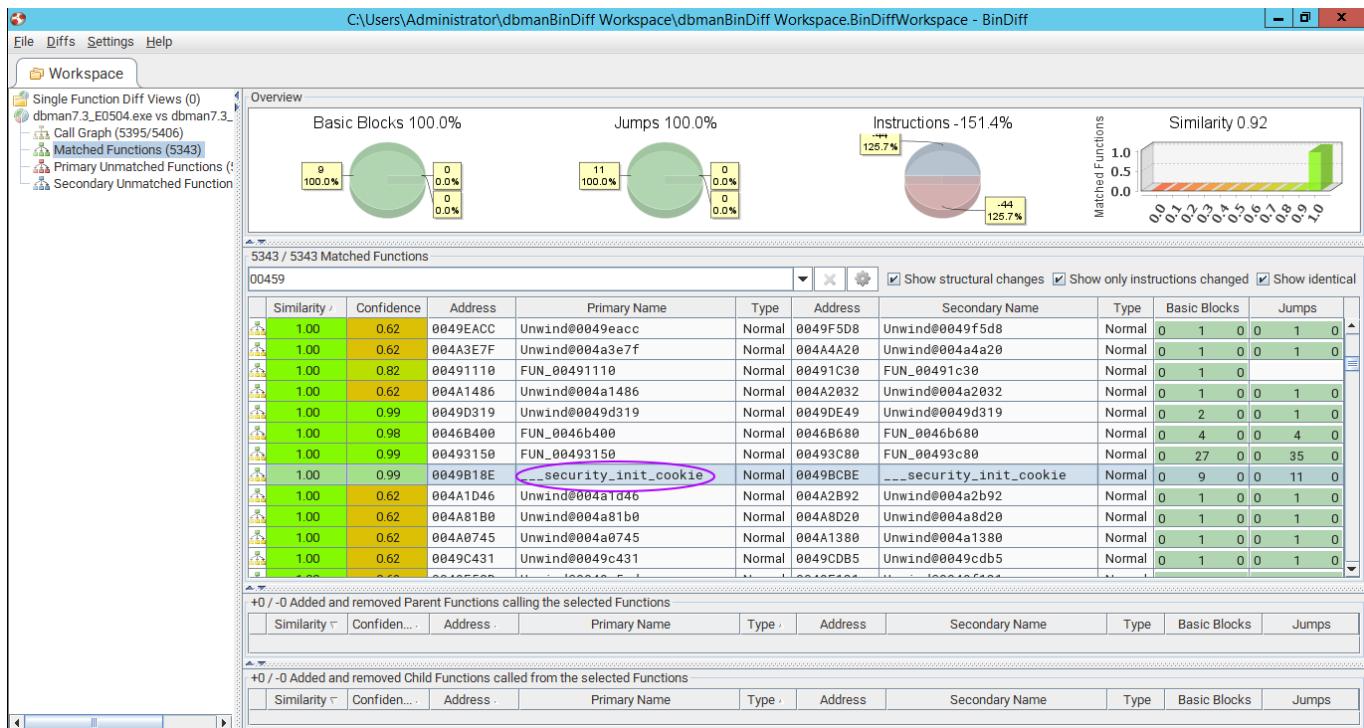
It's always interesting to have a first idea of the PE file(*dbman.exe*) security mitigations that are applied or not:

```
ixidor@pr0c:~/SHARE/build$ ./winchecksec ../dbman-iMC_PLAT_7.3_E0504_Ent_Win-003/dbman/bin/dbman.exe
Results for: ../dbman-iMC_PLAT_7.3_E0504_Ent_Win-003/dbman/bin/dbman.exe
Dynamic Base : "NotPresent"
ASLR : "NotPresent"
High Entropy VA : "NotPresent"
Force Integrity : "NotPresent"
Isolation : "Present"
NX : "NotPresent"
SEH : "Present"
CFG : "NotPresent"
RFG : "NotPresent"
SafeSEH : "NotPresent"
GS : "Present"
Authenticode : "NotPresent"
.NET : "NotPresent"

ixidor@pr0c:~/SHARE/build$ ./winchecksec ../dbman-iMC_PLAT_7.3_E0504P04_Win-002/dbman/bin/dbman.exe
Results for: ../dbman-iMC_PLAT_7.3_E0504P04_Win-002/dbman/bin/dbman.exe
Dynamic Base : "NotPresent"
ASLR : "NotPresent"
High Entropy VA : "NotPresent"
Force Integrity : "NotPresent"
Isolation : "Present"
NX : "NotPresent"
SEH : "Present"
CFG : "NotPresent"
RFG : "NotPresent"
SafeSEH : "NotPresent"
GS : "Present"
Authenticode : "NotPresent"
.NET : "NotPresent"
```

At this point, I cannot say if the output of `winchecksec` is quite reliable and/or accurate.

It seems that the binary has been compiled with the `/GS` flag:



<https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/exploit-protection-reference>

Confirming the bug(s)

(ZDI-17-836 / CVE-12561)

Approaches

In the context where we have two(or more) pieces of software, where one version contains a weakness/vulnerability and the other some sort of a patch to mitigate this flaw. Then, probably one of the most straightforward approach to id it is a process called **binary diffing**. We're looking for similarities/discrepancies in the binaries.

More often in large code bases, bringing a patch can also introduce new bugs.

Another possible approach could be to use **taint analysis** to be able to pinpoint ; or at least, to narrow down the space search related to the weakness. In this particular case, it would be interesting to see if such an approach could yield the function of interest. Given the architecture of iMC(Java and C++ code), I don't know if this approach could actually work.

There is the whole space of **dynamic binary instrumentation**(DBI), symbolic/concolic execution which has commonalities with the previous point.

Finally, whatever the approach chosen one can go with a blackbox or whitebox/greybox. In the context/mindset of exploit development, one is often interested in subverting/controlling designed functionalities. RE is a powerful but time consuming approach. The "exploitation" approach is way much more fun because what you're actually trying to do is to minimize the effort for a maximum gain(in theory) ; which in a way makes you some sort of an "optimized" programmer.

[ZDI-17-836 / CVE-12561]

To confirm the vulnerability or at least a bug, I am using the following python script(fuzz.py) from:

<https://www.github.com/pwnslinger/exploit-repo>

Often, it is easier to work backwards.

Given the information provided in the previous link, we should be able to identify two vulnerabilities:

- UAF(Use-After-Free)
- Stack-based buffer overflow

In both settings, I am using one Kali VM as the tester and a WindowsServer2012R2 VM(iMC software) as a testee.

UAF

To id UAF between the two versions of *dbman.exe*, we are going with a patch diffing process using Ghidra's *Version Tracking* to reduce and identify the number of potential function candidates ; using Bindiff to get a better visualization.

Given the nature of this class of bug, we are looking for something that roughly adds some check at some point ; something that frees and zeroes out a pointer or not, it depends from which perspective we're looking.

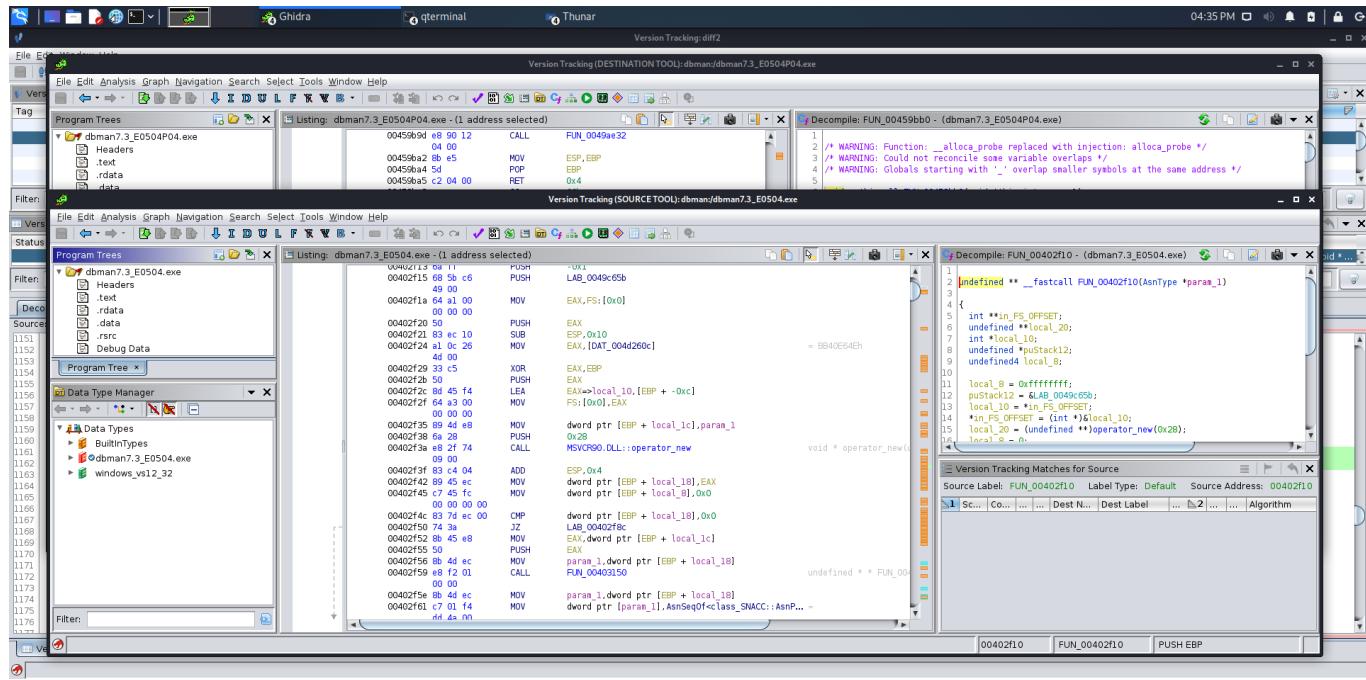
In this particular case, spotting the bug from a static only perspective isn't that easy because there is a call that is resolved dynamically i.e., when the binary is running. Combining static with a dynamic analysis/debugger helps.

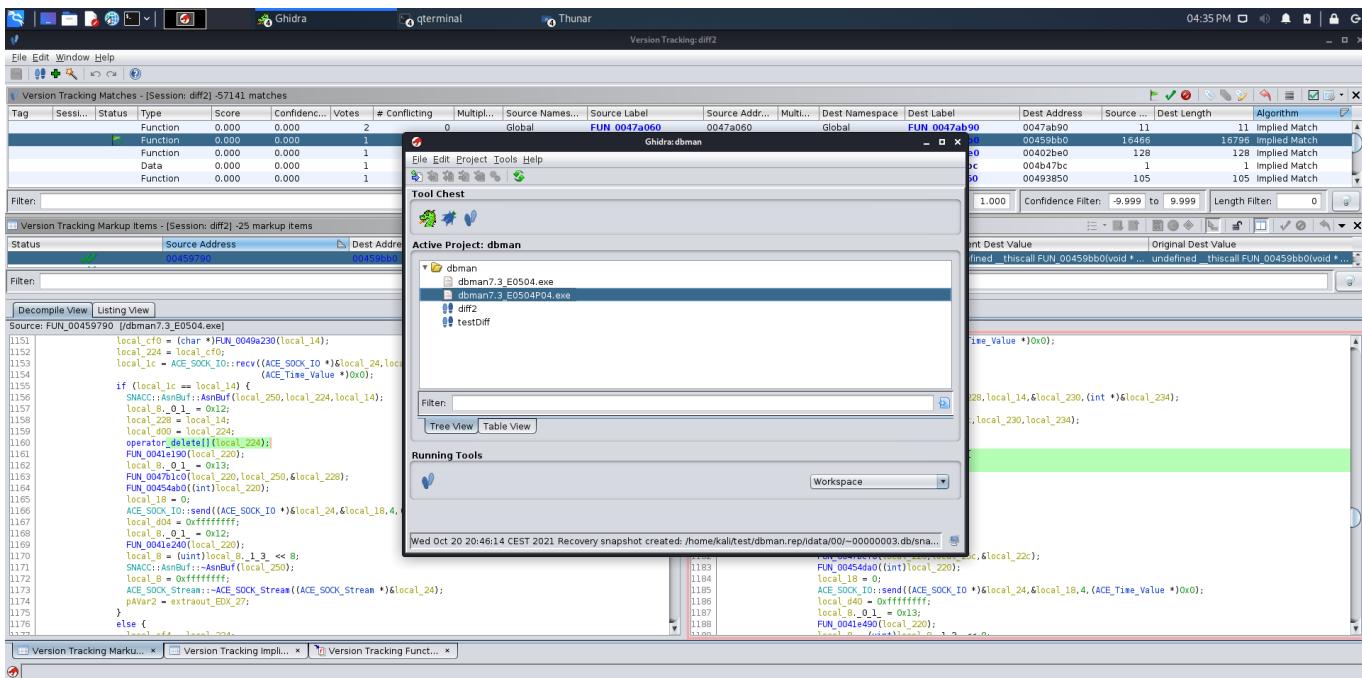
Software:

Each *dbman.exe* from the installation of:

iMC_PLAT_7.3_E0504_Ent_Win-003 and iMC_PLAT_7.3_E0504P04_Win-002

Ingesting binaries into Ghidra before using Version Tracking:





A first attempt was to look for references of the opcode command:

```
>>> hex(10012)
'0x271c'
```

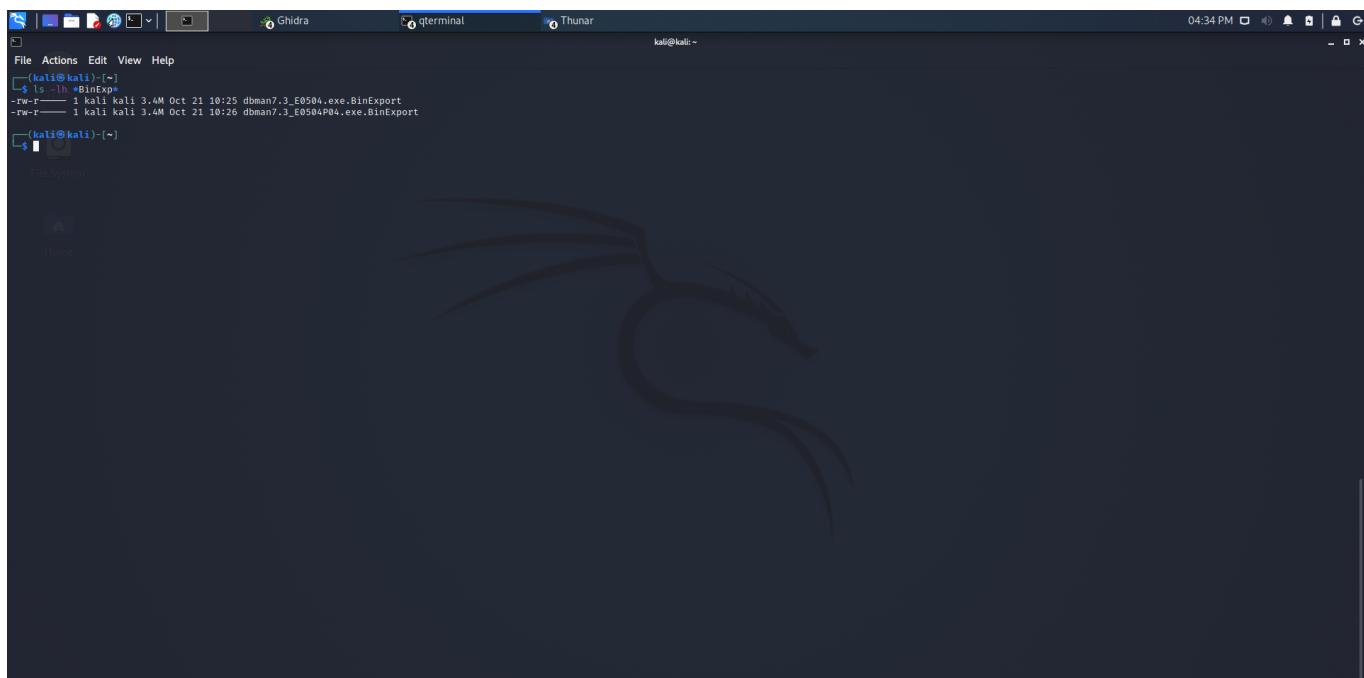
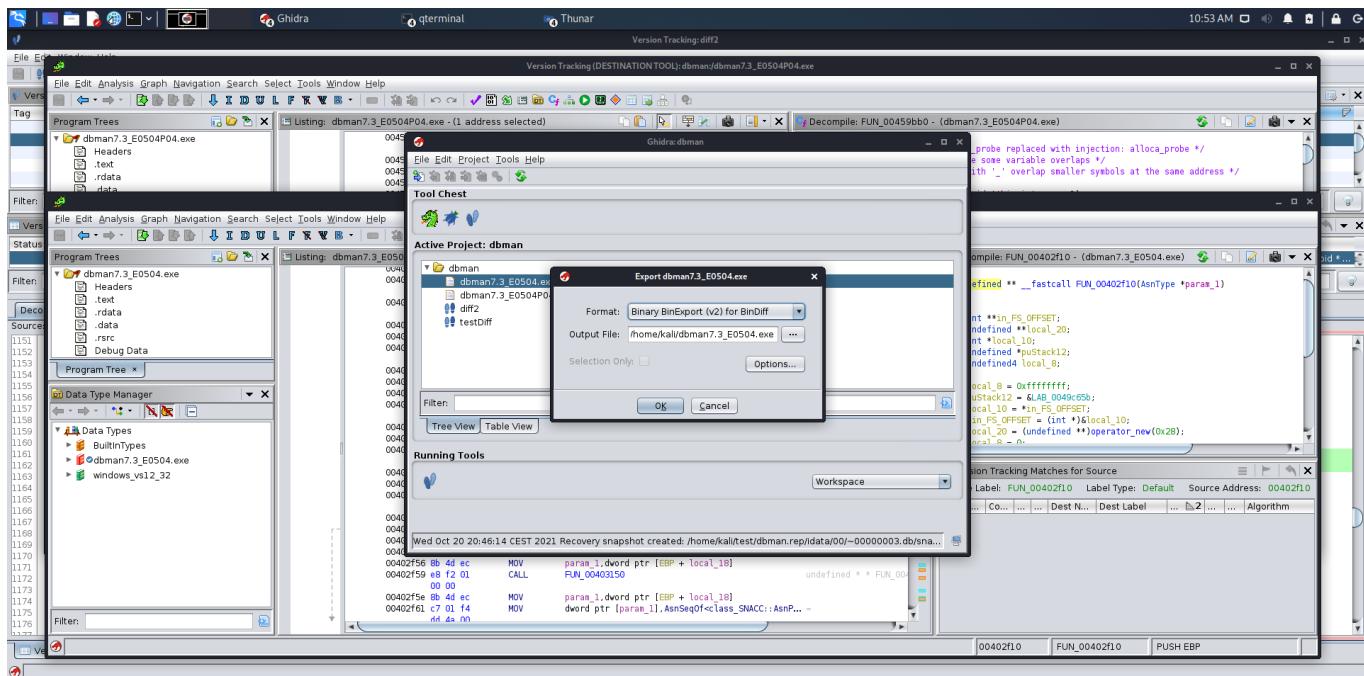
Didn't manage to get something useful by searching for this constant.

Speaking of opcode, it is likely that there is a typo in the *fuzz.py* script:

```
32 class AsnRemoteReservedFileRemove(object):
33     def __init__(self, reservedFilePath, backupPath, backFileExt, time):
34         super(AsnRemoteReservedFileRemove, self).__init__()
35         self.reservedFilePath = reservedFilePath
36         self.backupPath = backupPath
37         self.backFileExt = backFileExt
38         self.time = time
39         self.opode = 10012
40
41     def encode(self):
```

This should most likely be `self.opcode`.

Ghidra - BinExport for BinDiff visualization:



Back to Version Tracking, using and sorting by the *Implied Match* algorithm(maybe some sort of fuzzy matching):

| Tag | Sessi... | Status | Type | Score | Confidenc... | Votes | # Conflicting | Multipl... | Source Names... | Source Label | Source Addr... | Multi... | Dest Namespace | Dest Label | Dest Address | Source... | Dest Length | Algorithm |
|-----|----------|--------|----------|-------|--------------|-------|---------------|------------|-----------------|--------------|----------------|--------------|----------------|------------|--------------|-----------|---------------|-----------|
| | | | Function | 0.000 | 0.000 | 2 | 0 | Global | FUN_0047a060 | 0047a060 | Global | FUN_0047ab90 | 0047ab90 | 11 | 11 | 11 | Implied Match | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00459790 | 00459790 | Global | FUN_00459bb0 | 00459bb0 | 16466 | 16796 | 16796 | Implied Match | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00402be0 | 00402be0 | Global | FUN_00402be0 | 00402be0 | 128 | 128 | 128 | Implied Match | |
| | | | Data | 0.000 | 0.000 | 1 | 0 | Global | DAT_004b37b4 | 004b37b4 | Global | DAT_004b47bc | 004b47bc | 1 | 1 | 1 | Implied Match | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00492d20 | 00492d20 | Global | FUN_00493850 | 00493850 | 105 | 105 | 105 | Implied Match | |

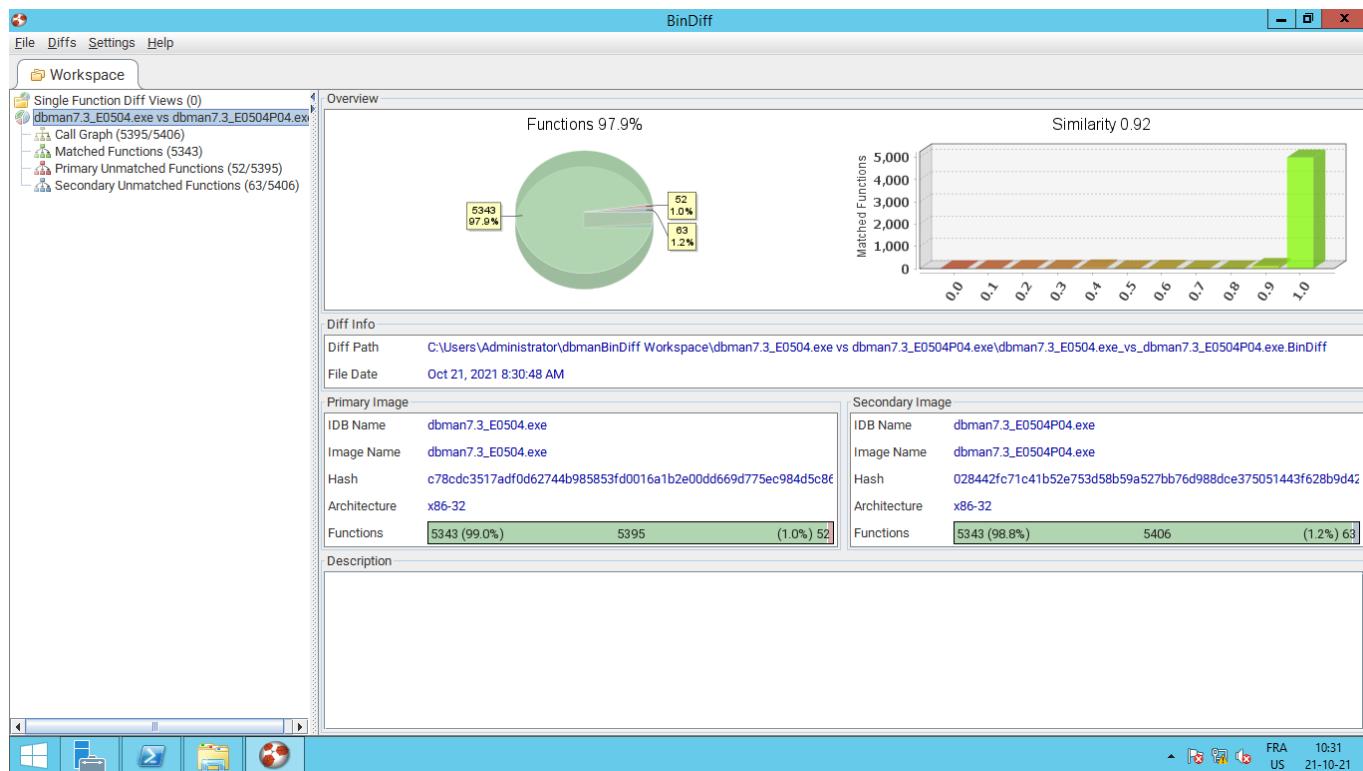
We can see:

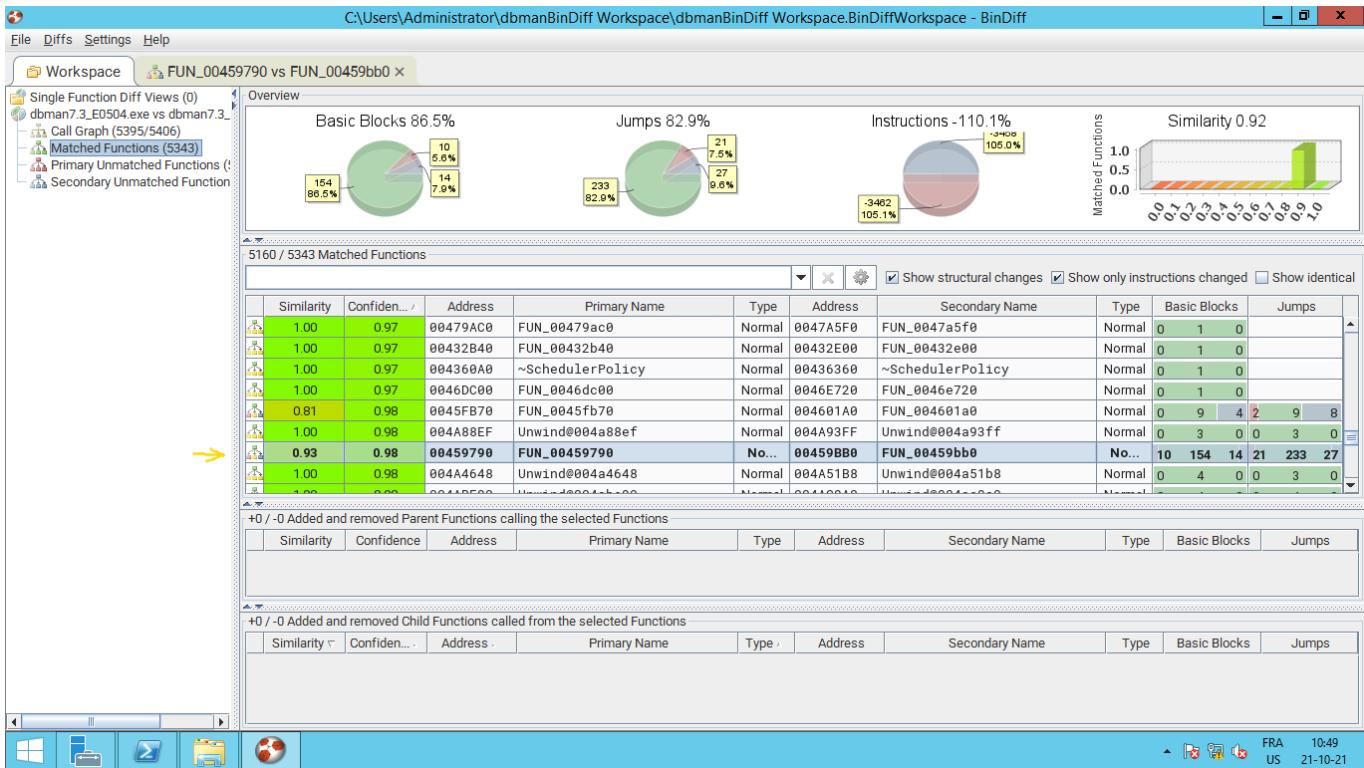
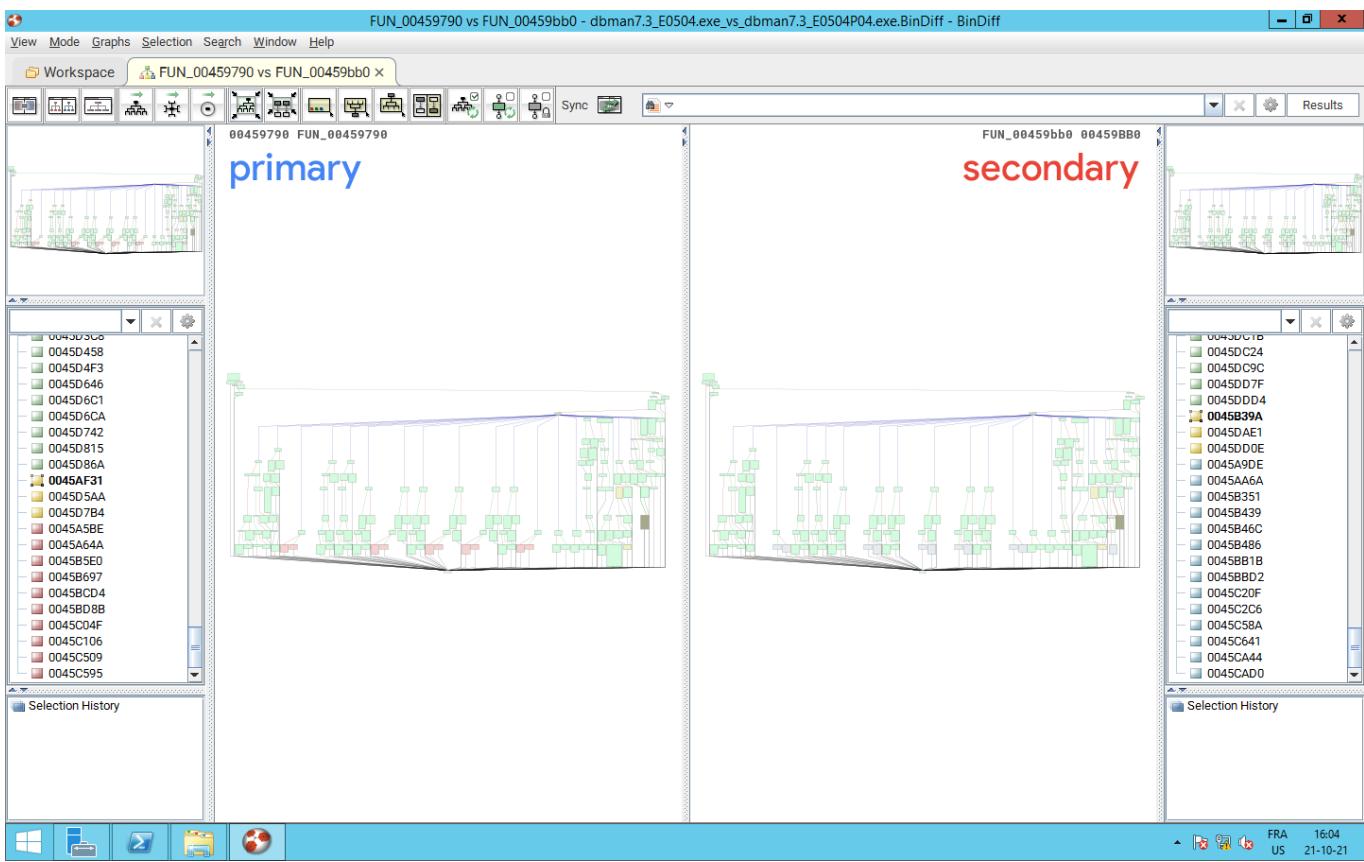
FUN_00459790 (7.3_E0504) and its corresponding **FUN_00459bb0** (7.3_E0504P04):

The screenshot shows the Ghidra interface with two windows open. The top window is titled "Version Tracking Matches - [Session: diff2].57141 matches" and displays a table of function mappings between two versions. The bottom window is titled "Version Tracking Markup Items - [Session: diff2]-25 markup items" and shows a list of specific items marked for review. Below these windows is a large assembly code editor comparing the assembly of **FUN_00459790** and **FUN_00459bb0**. The assembly code is identical, showing a series of recv, if, and operator delete instructions.

| Tag | Sess... | Status | Type | Score | Confiden... | Votes | # Conflicting | Multipl... | Source Names... | Source Label | Source Addr... | Multi... | Dest Namespace | Dest Label | Dest Address | Source... | Dest Length | Algorithm |
|-----|---------|--------|----------|-------|-------------|-------|---------------|------------|---------------------|--------------|----------------|---------------------|----------------|------------|--------------|-----------|---------------|-----------|
| | | | Function | 0.000 | 0.000 | 2 | 0 | Global | FUN_0047a060 | 0047a060 | Global | FUN_0047ab90 | 0047ab90 | 11 | 11 | 11 | Implied Match | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00402be0 | 00402be0 | Global | FUN_00402be0 | 00402be0 | 128 | 128 | 128 | Implied Match | |
| | | | Data | 0.000 | 0.000 | 1 | 0 | Global | DAT_004b37b4 | 004b37b4 | Global | DAT_004b47bc | 004b47bc | 1 | 1 | 1 | Implied Match | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00492d20 | 00492d20 | Global | FUN_00493850 | 00493850 | 105 | 105 | 105 | Implied Match | |

Looking at this in **Bindiff7**:





Back to Version Tracking:

Switch case for opcode 10012(**0x271c**):

Ghidra qterminal Thunar Version Tracking: diff2 05:05 PM

Version Tracking Matches - [Session: diff2] 57141 matches

| Tag | Sess... | Status | Type | Score | Confidenc... | Votes | # Conflicting | Multipl... | Source Names... | Source Label | Source Addr... | Multi... | Dest Namespace | Dest Label | Dest Address | Source... | Dest Length | Algorithm |
|-----|---------|--------|----------|-------|--------------|-------|---------------|------------|-----------------|--------------|----------------|--------------|----------------|------------|--------------|---------------|-------------|-----------|
| | | | Function | 0.000 | 0.000 | 2 | 0 | Global | FUN_0047a060 | 0047a060 | Global | FUN_0047ab90 | 0047ab90 | 11 | 11 | Implied Match | | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | global | FUN_00459790 | 00459790 | Global | FUN_00459bb0 | 00459bb0 | 16466 | 16796 | Implied Match | | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00402be0 | 00402be0 | Global | FUN_00402be0 | 00402be0 | 128 | 128 | Implied Match | | |
| | | | Data | 0.000 | 0.000 | 1 | 0 | Global | DAT_000b37b4 | 000b37b4 | Global | DAT_004047bc | 004047bc | 1 | 1 | Implied Match | | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00492d20 | 00492d20 | Global | FUN_00493850 | 00493850 | 105 | 105 | Implied Match | | |

Filter: Score Filter: 0.000 to 1.000 Confidence Filter: -9.999 to 9.999 Length Filter: 0

Version Tracking Markup Items - [Session: diff2] 25 markup items

| Status | Source Address | Dest Address | Markup Type | Source Value | Current Dest Value | Original Dest Value |
|--------|----------------|--------------|--------------------|--|--|--|
| W | 00459790 | 00459bb0 | Function Signature | undefined __thiscall FUN_00459790(void *...) | undefined __thiscall FUN_00459bb0(void *...) | undefined __thiscall FUN_00459bb0(void *...) |
| W | local_d00 | local_d00 | AAAI | zunrAkn | zunrAkn | zunrAkn |

Filter:

Decompile View Listing View

Source: FUN_00459790 (/dbman7.3_E0504.exe)

```

1144     case 0x271c:
1145         local_lc = ACE_SOCK_IO::recv((ACE_SOCK_IO *)local_24,&local_14,4,
1146                                     (ACE_Time_Value *)0x0);
1147         if (local_lc == 4) {
1148             local_14 <<= ((local_14 & 0xffff) << 8 | (int)(local_14 & 0xff00) >> 8) << 0x10 |
1149             ((int)local_14 & 0x10) & 0xffff) << 8 |
1150             ((int)(int)local_14 >> 0x10 & 0xffff) >> 8;
1151         local_cf0 = _towlcstr(FUN_00492d20(local_14));
1152         local_224 = local_cf0;
1153         local_lc = ACE_SOCK_IO::recv(ACE_SOCK_IO *)&local_24,local_cf0,local_14,
1154                                     (ACE_Time_Value *)0x0);
1155         if (local_lc == local_14) {
1156             SNACC::AnBuf::AnBuf(local_250,local_224,local_14);
1157             local_B_0_1_ = 0x2;
1158             local_228 = local_14;
1159             local_d00 = local_224;
1160             operator_delete[](local_224);
1161             FUN_00459790();
1162             local_B_0_1_ = 0x3;
1163             FUN_0047b1c0(local_220,local_250,local_228);
1164             FUN_00454a00(int)local_220;
1165             local_18 = 0;
1166             ACE_SOCK_Stream((ACE_SOCK_IO *)&local_24,&local_18,4,(ACE_Time_Value *)0x0);
1167             local_d04 = 0xffffffff;
1168             local_B_0_1_ = 0x2;
1169             FUN_0041e240(local_220);

```

Destination: FUN_00459bb0 (/dbman7.3_E0504P04.exe)

```

1153     case 0x271c:
1154         local_lc = ACE_SOCK_IO::recv((ACE_SOCK_IO *)local_24,&local_14,4,
1155                                     (ACE_Time_Value *)0x0);
1156         if (local_lc == 4) {
1157             local_14 = ((local_14 & 0xffff) << 8 | (int)(local_14 & 0xff00) >> 8) << 0x10 |
1158             ((int)local_14 >> 0x10) & 0xffff) << 8 |
1159             ((int)(int)local_14 >> 0x10 & 0xffff) >> 8;
1160         local_d24 = _towlcstr(FUN_0049ad00(local_14));
1161         local_228 = local_d24;
1162         local_lc = ACE_SOCK_IO::recv((ACE_SOCK_IO *)local_24,local_d24,local_14,
1163                                     (ACE_Time_Value *)0x0);
1164         if (local_lc == local_14) {
1165             local_224 = local_228;
1166             local_230 = (char *)0x0;
1167             local_234 = 0;
1168             iVar9 = FUN_0046cd00(int)local_228,local_14,&local_230,(int)&local_234;
1169             iVar9->operator_delete[](local_234);
1170             local_B_0_1_ = 0x3;
1171             local_22c = local_234;
1172             if (local_230 != (char *)0x0) {
1173                 free(local_230);
1174             }
1175             local_230 = (char *)0x0;
1176             local_d3c = local_228;
1177             operator_delete[](local_228);
1178

```

Bindiff:

FUN_00459790 (7.3_E0504) and its corresponding **FUN_00459bb0 (7.3_E0504P04):**

FUN_00459790 vs FUN_00459bb0 - dbman7.3_E0504.exe_vs_dbman7.3_E0504P04.exe.BindDiff - BinDiff

View Mode Graphs Selection Search Window Help

Workspace FUN_00459790 vs FUN_00459bb0 x

00459790 FUN_00459790

```

00459F31 MOV EDX, dword ptr [EBP + local_14]
00459F34 PUSH EDX
00459F35 MOV EAX, dword ptr [EBP + local_224]

0045AF3B PUSH EAX
0045AF3C LEA ECX, [EBP + 0xfffffd84]

0045AF42 CALL dword ptr [PTR_AnBuf_0044ad64]
0045AF48 MOV byte ptr [EBP + local_8], 0x12
0045AF4C MOV ECX, dword ptr [EBP + local_14]
0045AF50 MOV dword ptr [EBP + local_228], ECX
0045AF52 MOV EDX, dword ptr [EBP + local_124]
0045AF54 MOV ECX, dword ptr [EBP + local_000], EDX
0045AF56 MOV EAX, dword ptr [EBP + local_000]
0045AF58 PUSH ECX
0045AF59 LEA EDX, [EBP + 0xfffffd84]
0045AF5A LEA ECX, [EBP + 0xfffffdde4]
0045AF5B CALL FUN_0041e198
0045AF5C MOV byte ptr [EBP + local_8], 0x13
0045AF5D LEA ECX, [EBP + 0xfffffdde4]
0045AF5E PUSH ECX
0045AF5F LEA EDX, [EBP + 0xfffffd84]
0045AF60 PUSH EDX
0045AF61 LEA ECX, [EBP + 0xfffffd84]
0045AF62 PUSH ECX
0045AF63 CALL operator_delete[]
0045AF64 PUSH ECX
0045AF65 LEA ECX, [EBP + 0xfffffd84]
0045AF66 LEA ECX, [EBP + 0xfffffdde4]
0045AF67 PUSH ECX
0045AF68 CALL PTR_AnBuf_0044ad64
0045AF69 LEA ECX, [EBP + 0xfffffd84]
0045AF6A LEA ECX, [EBP + 0xfffffdde4]
0045AF6B CALL FUN_0041e198
0045AF6C MOV byte ptr [EBP + local_8], 0x12
0045AF6D LEA ECX, [EBP + 0xfffffd84]
0045AF6E LEA ECX, [EBP + 0xfffffdde4]
0045AF6F CALL PTR_send_004ad094
0045AF70 MOV dword ptr [EBP + local_d84], 0xffffffff
0045AF71 MOV byte ptr [EBP + local_8], 0x12
0045AF72 LEA ECX, [EBP + 0xfffffd84]
0045AF73 CALL PTR_AnBuf_0044ad64
0045AF74 LEA ECX, [EBP + 0xfffffd84]
0045AF75 LEA ECX, [EBP + 0xfffffdde4]
0045AF76 CALL PTR_AnBuf_0044ad64
0045AF77 LEA ECX, [EBP + 0xfffffd84]
0045AF78 LEA ECX, [EBP + 0xfffffdde4]
0045AF79 LEA ECX, [EBP + 0xfffffd84]
0045AF7A LEA ECX, [EBP + 0xfffffdde4]
0045AF7B LEA ECX, [EBP + 0xfffffd84]
0045AF7C LEA ECX, [EBP + 0xfffffdde4]
0045AF7D LEA ECX, [EBP + 0xfffffd84]
0045AF7E LEA ECX, [EBP + 0xfffffdde4]
0045AF7F LEA ECX, [EBP + 0xfffffd84]
0045AF80 LEA ECX, [EBP + 0xfffffdde4]
0045AF81 LEA ECX, [EBP + 0xfffffd84]
0045AF82 LEA ECX, [EBP + 0xfffffdde4]
0045AF83 LEA ECX, [EBP + 0xfffffd84]
0045AF84 LEA ECX, [EBP + 0xfffffdde4]
0045AF85 LEA ECX, [EBP + 0xfffffd84]
0045AF86 LEA ECX, [EBP + 0xfffffdde4]
0045AF87 LEA ECX, [EBP + 0xfffffd84]
0045AF88 LEA ECX, [EBP + 0xfffffdde4]
0045AF89 LEA ECX, [EBP + 0xfffffd84]
0045AF8A LEA ECX, [EBP + 0xfffffdde4]
0045AF8B LEA ECX, [EBP + 0xfffffd84]
0045AF8C LEA ECX, [EBP + 0xfffffdde4]
0045AF8D LEA ECX, [EBP + 0xfffffd84]
0045AF8E LEA ECX, [EBP + 0xfffffdde4]
0045AF8F LEA ECX, [EBP + 0xfffffd84]
0045AF90 LEA ECX, [EBP + 0xfffffdde4]
0045AF91 LEA ECX, [EBP + 0xfffffd84]
0045AF92 LEA ECX, [EBP + 0xfffffdde4]
0045AF93 CALL FUN_0047ab90
0045AF94 LEA ECX, [EBP + 0xfffffd84]
0045AF95 PUSH ECX
0045AF96 LEA ECX, [EBP + 0xfffffdde4]
0045AF97 LEA ECX, [EBP + 0xfffffd84]
0045AF98 LEA ECX, [EBP + 0xfffffdde4]
0045AF99 LEA ECX, [EBP + 0xfffffd84]
0045AF9A LEA ECX, [EBP + 0xfffffdde4]
0045AF9B LEA ECX, [EBP + 0xfffffd84]
0045AF9C LEA ECX, [EBP + 0xfffffdde4]
0045AF9D LEA ECX, [EBP + 0xfffffd84]
0045AF9E LEA ECX, [EBP + 0xfffffdde4]
0045AF9F LEA ECX, [EBP + 0xfffffd84]
0045AF9A0 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A1 LEA ECX, [EBP + 0xfffffd84]
0045AF9A2 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A3 LEA ECX, [EBP + 0xfffffd84]
0045AF9A4 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A5 LEA ECX, [EBP + 0xfffffd84]
0045AF9A6 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A7 LEA ECX, [EBP + 0xfffffd84]
0045AF9A8 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9 LEA ECX, [EBP + 0xfffffd84]
0045AF9A00 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A01 LEA ECX, [EBP + 0xfffffd84]
0045AF9A02 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A03 LEA ECX, [EBP + 0xfffffd84]
0045AF9A04 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A05 LEA ECX, [EBP + 0xfffffd84]
0045AF9A06 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A07 LEA ECX, [EBP + 0xfffffd84]
0045AF9A08 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A09 LEA ECX, [EBP + 0xfffffd84]
0045AF9A0A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A0B LEA ECX, [EBP + 0xfffffd84]
0045AF9A0C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A0D LEA ECX, [EBP + 0xfffffd84]
0045AF9A0E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A0F LEA ECX, [EBP + 0xfffffd84]
0045AF9A10 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A11 LEA ECX, [EBP + 0xfffffd84]
0045AF9A12 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A13 LEA ECX, [EBP + 0xfffffd84]
0045AF9A14 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A15 LEA ECX, [EBP + 0xfffffd84]
0045AF9A16 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A17 LEA ECX, [EBP + 0xfffffd84]
0045AF9A18 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A19 LEA ECX, [EBP + 0xfffffd84]
0045AF9A1A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A1B LEA ECX, [EBP + 0xfffffd84]
0045AF9A1C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A1D LEA ECX, [EBP + 0xfffffd84]
0045AF9A1E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A1F LEA ECX, [EBP + 0xfffffd84]
0045AF9A20 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A21 LEA ECX, [EBP + 0xfffffd84]
0045AF9A22 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A23 LEA ECX, [EBP + 0xfffffd84]
0045AF9A24 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A25 LEA ECX, [EBP + 0xfffffd84]
0045AF9A26 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A27 LEA ECX, [EBP + 0xfffffd84]
0045AF9A28 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A29 LEA ECX, [EBP + 0xfffffd84]
0045AF9A2A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A2B LEA ECX, [EBP + 0xfffffd84]
0045AF9A2C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A2D LEA ECX, [EBP + 0xfffffd84]
0045AF9A2E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A2F LEA ECX, [EBP + 0xfffffd84]
0045AF9A30 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A31 LEA ECX, [EBP + 0xfffffd84]
0045AF9A32 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A33 LEA ECX, [EBP + 0xfffffd84]
0045AF9A34 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A35 LEA ECX, [EBP + 0xfffffd84]
0045AF9A36 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A37 LEA ECX, [EBP + 0xfffffd84]
0045AF9A38 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A39 LEA ECX, [EBP + 0xfffffd84]
0045AF9A3A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A3B LEA ECX, [EBP + 0xfffffd84]
0045AF9A3C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A3D LEA ECX, [EBP + 0xfffffd84]
0045AF9A3E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A3F LEA ECX, [EBP + 0xfffffd84]
0045AF9A40 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A41 LEA ECX, [EBP + 0xfffffd84]
0045AF9A42 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A43 LEA ECX, [EBP + 0xfffffd84]
0045AF9A44 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A45 LEA ECX, [EBP + 0xfffffd84]
0045AF9A46 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A47 LEA ECX, [EBP + 0xfffffd84]
0045AF9A48 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A49 LEA ECX, [EBP + 0xfffffd84]
0045AF9A4A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A4B LEA ECX, [EBP + 0xfffffd84]
0045AF9A4C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A4D LEA ECX, [EBP + 0xfffffd84]
0045AF9A4E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A4F LEA ECX, [EBP + 0xfffffd84]
0045AF9A50 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A51 LEA ECX, [EBP + 0xfffffd84]
0045AF9A52 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A53 LEA ECX, [EBP + 0xfffffd84]
0045AF9A54 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A55 LEA ECX, [EBP + 0xfffffd84]
0045AF9A56 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A57 LEA ECX, [EBP + 0xfffffd84]
0045AF9A58 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A59 LEA ECX, [EBP + 0xfffffd84]
0045AF9A5A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A5B LEA ECX, [EBP + 0xfffffd84]
0045AF9A5C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A5D LEA ECX, [EBP + 0xfffffd84]
0045AF9A5E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A5F LEA ECX, [EBP + 0xfffffd84]
0045AF9A60 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A61 LEA ECX, [EBP + 0xfffffd84]
0045AF9A62 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A63 LEA ECX, [EBP + 0xfffffd84]
0045AF9A64 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A65 LEA ECX, [EBP + 0xfffffd84]
0045AF9A66 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A67 LEA ECX, [EBP + 0xfffffd84]
0045AF9A68 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A69 LEA ECX, [EBP + 0xfffffd84]
0045AF9A6A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A6B LEA ECX, [EBP + 0xfffffd84]
0045AF9A6C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A6D LEA ECX, [EBP + 0xfffffd84]
0045AF9A6E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A6F LEA ECX, [EBP + 0xfffffd84]
0045AF9A70 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A71 LEA ECX, [EBP + 0xfffffd84]
0045AF9A72 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A73 LEA ECX, [EBP + 0xfffffd84]
0045AF9A74 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A75 LEA ECX, [EBP + 0xfffffd84]
0045AF9A76 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A77 LEA ECX, [EBP + 0xfffffd84]
0045AF9A78 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A79 LEA ECX, [EBP + 0xfffffd84]
0045AF9A7A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A7B LEA ECX, [EBP + 0xfffffd84]
0045AF9A7C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A7D LEA ECX, [EBP + 0xfffffd84]
0045AF9A7E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A7F LEA ECX, [EBP + 0xfffffd84]
0045AF9A80 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A81 LEA ECX, [EBP + 0xfffffd84]
0045AF9A82 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A83 LEA ECX, [EBP + 0xfffffd84]
0045AF9A84 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A85 LEA ECX, [EBP + 0xfffffd84]
0045AF9A86 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A87 LEA ECX, [EBP + 0xfffffd84]
0045AF9A88 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A89 LEA ECX, [EBP + 0xfffffd84]
0045AF9A8A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A8B LEA ECX, [EBP + 0xfffffd84]
0045AF9A8C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A8D LEA ECX, [EBP + 0xfffffd84]
0045AF9A8E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A8F LEA ECX, [EBP + 0xfffffd84]
0045AF9A90 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A91 LEA ECX, [EBP + 0xfffffd84]
0045AF9A92 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A93 LEA ECX, [EBP + 0xfffffd84]
0045AF9A94 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A95 LEA ECX, [EBP + 0xfffffd84]
0045AF9A96 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A97 LEA ECX, [EBP + 0xfffffd84]
0045AF9A98 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A99 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9B LEA ECX, [EBP + 0xfffffd84]
0045AF9A9C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9D LEA ECX, [EBP + 0xfffffd84]
0045AF9A9E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9F LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A1 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A2 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A3 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A4 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A5 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A6 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A7 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A8 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A9 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A01 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A02 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A03 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A04 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A05 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A06 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A07 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A08 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A09 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0B LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0D LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0F LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A001 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A002 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A003 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A004 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A005 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A006 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A007 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A008 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A009 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00B LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00D LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00F LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0000 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0001 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0002 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0003 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0004 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0005 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0006 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0007 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0008 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0009 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000B LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000D LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000F LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00000 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00001 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00002 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00003 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00004 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00005 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00006 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00007 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00008 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00009 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0000A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0000B LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0000C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0000D LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A0000E LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A0000F LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000000 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000001 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000002 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000003 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000004 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000005 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000006 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000007 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A000008 LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A000009 LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00000A LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00000B LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00000C LEA ECX, [EBP + 0xfffffdde4]
0045AF9A9A00000D LEA ECX, [EBP + 0xfffffd84]
0045AF9A9A00
```

FUN_00459bb0 00459bb0

```

00459BB0 FUN_00459bb0
0045B3 9A PUSH    -0x8
0045B3 9C SUB     ESP, 0x14
0045B3 9F MOV     ECX, ESP
0045B3 A1 MOV     dword ptr [EBP + local_d34], ESP
0045B3 A7 MOV     dword ptr [EBP + local_fbc], ECX
0045B3 AD PUSH    0x1

0045B3 AF MOV     ECX, dword ptr [EBP + local_fbc]
0045B3 B5 CALL    dword ptr [PTR_AsnEnum_004ae5ac]
0045B3 BB MOV     EDX, dword ptr [EBP + local_fbc]
0045B3 C1 MOV     dword ptr [EDX], vftable
0045B3 C7 MOV     EAX, dword ptr [EBP + local_fbc]
0045B3 CD MOV     dword ptr [EAX + 0x8], vftable
0045B3 D4 MOV     ECX, dword ptr [EBP + local_fbc]
0045B3 DA MOV     dword ptr [EBP + local_10b8], ECX

0045B3 E0 MOV     byte ptr [EBP + local_8], 0x12

0045B3 E4 MOV     ECX, dword ptr [EBP + local_20]
0045B3 E7 CALL   FUN_0045eb70

0045B3 EC PUSH   EAX
0045B3 ED MOV     byte ptr [EBP + local_8], 0x0
0045B3 F1 CALL   FUN_00464510
0045B3 F6 ADD    ESP, 0x1c

0045B3 F9 PUSH   s_dbman_decode_len()_failed!_004b7148
0045B3 FE MOV     EDX, dword ptr [EBP + local_28]
0045B4 01 PUSH   EDX

0045B4 02 PUSH   0x1

0045B4 04 CALL   FUN_00475470

0045B4 09 ADD    ESP, 0xc
0045B4 0C MOV     ECX, dword ptr [EBP + local_20]

0045B4 0F CALL   FUN_0045ec20
0045B4 14 MOV     dword ptr [EBP + local_d38], 0x0

0045B4 1E MOV     dword ptr [EBP + local_8], 0xffffffff
0045B4 25 LEA    ECX, [EBP + -0x20]
0045B4 28 CALL   dword ptr [PTR_~ACE_SOCK_Stream_004ae000]
0045B4 2E MOV     EAX, dword ptr [EBP + local_d38]
0045B4 34 JMP    LAB_0045ddd4

```

We are interested in:

- @0045b40f: FUN_0045ec20

This is because of the following flow:

From/inside FUN_00459bb0 (switch case for 0x271c):

Ghidra Version Tracking (DESTINATION TOOL): dbman/dbman7.3_E0504P04.exe 01:48 PM

Program Trees Listing: dbman7.3_E0504P04.exe - (8 addresses selected)

```

0000 LAB_0045b351 XREF[1]: 0045b29e(j)
    MOV EDX,dword ptr [EBP + local_228]
0045b351 8b 95 dc MOV
0045b357 89 95 e0 MOV dword ptr [EBP + local_224],EDX
0045b35d 8d ff ff MOV
0045b361 8b 95 d0 MOV dword ptr [EBP + local_230],0x0
0045b367 8d ff ff MOV
0045b371 8b 95 d0 MOV dword ptr [EBP + local_234],0x0
0045b377 50 LEA EAX=>local_234,[EBP + 0xfffffffdd0]
0045b378 8d a0 00 PUSH EAX
0045b37e 8d ff ff LEA this=>local_230,[EBP + 0xfffffffdd4]
0045b37f 8d ff ff PUSH this
0045b381 8d ff ff MOV EDX,dword ptr [EBP + local_14]
0045b382 8d ff ff PUSH EDX
0045b383 8b e5 e0 MOV EAX,dword ptr [EBP + local_224]
0045b389 8d ff ff PUSH EAX
0045b38a 8b 41 1c CALL FUN_0046cf0 undefined FUN_0046cf0
0045b38f 83 ec 10 ADD ESP,0x10
0045b392 85 c0 TEST EAX,EAX
0045b394 8b 94 9f JZ LAB_0045b439
0045b395 8d ff ff MOV
0045b39a 8b f8 PUSH -0x8
0045b39c 83 ec 14 SUB ESP,0x14
0045b39f 8b cc MOV this,ESP
0045b3a1 89 a5 d0 MOV dword ptr [EBP + local_d34],ESP
0045b3a2 8d ff ff MOV
0045b3a7 89 8d 40 MOV dword ptr [EBP + local_fbc],this
0045b3ad 8a 01 PUSH 0x1
0045b3af 8b 8d 40 MOV this,dword ptr [EBP + local_fbc]
0045b3b5 8d 15 ac CALL dword ptr [->CPPASN1.DLL::SNACC::AsnEnum::AsnE...
0045b3b8 8b 95 40 MOV EDX,dword ptr [EBP + local_fbc]
0045b3c1 8d ff ff MOV
0045b3c7 8b 8d 40 MOV dword ptr [EBX],SNACC::AsnBmanCmdCode::vtable -
0045b3c8 8b 8d 40 MOV EAX,dword ptr [EBP + local_fbc]

```

Decompiler: FUN_0045bb0 - (dbman7.3_E0504P04.exe)

```

case 0x71c2:
    local_1c = ACE_SOCK_IO::recv((ACE_SOCK_IO *)&local_24,&local_14,4,
                                (ACE_Time_Value *)0x0);
    if (local_1c == 4) {
        local_14 = ((int)local_14 & 0xff) <> 0x10 | (int)(local_14 & 0xffff) <> 8) <> 0x10 |
        (int)(int)local_14 >> 0x10 & 0xffff) <> 8;
        local_d24 = (void *)FUN_0049a609(local_14);
        local_228 = local_d24;
        local_1c = ACE_SOCK_IO::recv((ACE_SOCK_IO *)&local_24,local_d24,local_14,
                                    (ACE_Time_Value *)0x0);
        if (local_1c == local_14) {
            local_228 = local_228;
            local_230 = (char *)0x0;
            local_18 = 0;
            iVar8 = FUN_0046cf0((int)local_228,local_14,&local_230,(int *)local_234);
            if (iVar8 == 0) {
                SNACC::AsnBuf::AsnBuf(local_25c,local_230,local_234);
                local_228 = local_228 & 0x13;
                local_230 = local_230 & 0x24;
                if (local_230 != (char *)0x0) {
                    free(local_230);
                    local_230 = (char *)0x0;
                }
                local_d28 = local_228;
                operator_delete((local_228));
                local_228 = (void *)0x0;
                FUN_0041a3e0(local_220);
                local_B_0_1 = 0x14;
                local_18 = 0;
                local_18 = FUN_004544d4((int)local_220);
                local_18 = 0;
                ACE_SOCK_IO::send((ACE_SOCK_IO *)&local_24,&local_18,4,(ACE_Time_Value *)0x0);
                local_40 = 0xffffffff;
                local_18 = FUN_0041a409(local_220);
                local_B_0_1 = 0x13 << 8;
                SNACC::AsnBuf::AsnBuf((local_25c));
            }
        }
        local_d28 = local_228;
        operator_delete((local_228));
        local_228 = (void *)0x0;
        FUN_0041a3e0(local_220);
        local_B_0_1 = 0x14;
        local_18 = 0;
        local_18 = FUN_004544d4((int)local_220);
        local_18 = 0;
        ACE_SOCK_STREAM::ACE_SOCK_Stream((ACE_SOCK_Stream *)local_24);
        ppuvar2 = extrastuff_EDX_28;
    }
    else {
        local_d34 = &stack0xfffffe88;
        SNACC::AsnBuf::AsnBuf((AsnEnum *)&stack0xfffffe88,1);
        local_B_0_1 = 0x12;
        pAvard = (ACE_SOCK_Stream *)FUN_0045b70((int)local_20);
        local_18 = 0;
        local_B_0_1 = 0x13 << 8;
        FUN_00464510(local_20,"dbman_decode_len() failed!\n");
        FUN_00474701(local_28,"dbman_decode_len() failed!\n");
        FUN_0045c20((local_20));
        local_d38 = 0;
        local_d38 = 0xffffffff;
        ACE_SOCK_Stream::ACE_SOCK_Stream((ACE_SOCK_Stream *)local_24);
        ppuvar2 = extrastuff_EBX_27;
    }
}
else {
    local_d28 = local_228;
    operator_delete((local_228));
    local_d2c = &stack0xfffffe88;
    SNACC::AsnBuf::AsnBuf((AsnEnum *)&stack0xfffffe88,1);
    FUN_00464510((ACE_SOCK_Stream *)local_24);
    FUN_00475401((local_20));
    local_18 = 0;
    local_18 = "Receive AsnBmanCmdCode::vtable REMOVE_RESERVED_FILE_REQ data_len
                error expect %d bytes infact %d bytes"
    local_d30 = 0xffffffff;
    local_d30 = 0xffffffff;
    ACE_SOCK_Stream::ACE_SOCK_Stream((ACE_SOCK_Stream *)local_24);
    ppuvar2 = extrastuff_EDX_26;
}
else {
}
```

Version Tracking Matches for Destination

| Score | Confid... | Votes | # C... | Source Na... | Source Label | Sour... | Dest... | Algorithm | |
|-------|-----------|-------|--------|--------------|--------------|----------|---------|-----------|---------------|
| 0.000 | 0.000 | 1 | 0 | Global | FUN_0045b70 | 00459790 | 16466 | 16796 | Implied Match |

We need to fail this **if** condition to end up in the **else** clause:

Ghidra Version Tracking (DESTINATION TOOL): dbman/dbman7.3_E0504P04.exe 01:45 PM

Program Trees Listing: dbman7.3_E0504P04.exe - (8 addresses selected)

```

0045b3f1 83 1a 91 CALL FUN_00464510 undefined FUN_00464510
0045b3f6 83 c4 0c ADD ESP,0x1c
0045b3f9 68 48 71 PUSH s_dbman_decode_len().failed_004b7148 ~ "dbman_decode...
0045b3ff 8d 55 dc MOV EDX,dword ptr [EBP + local_28]
0045b401 8d ff ff MOV EDX=>CDataCommStringQueueT::deal_msg_004b7734 ~ "CDataConn...
0045b404 8d 67 a0 CALL FUN_00475470 undefined FUN_00475470
0045b409 83 c4 0c ADD ESP,0x1c
0045b40c 8d ff ff MOV this,dword ptr [EBP + local_20]
0045b40f 8d 00 36 CALL FUN_0046c520 undefined FUN_0046c520
0045b414 8d 75 cc MOV dword ptr [EBP + local_d38],0x0
12 ff ff
0045b416 8d ff ff MOV
0045b41e 8d 4d 00 MOV dword ptr [EBP + local_8],0xffffffff
0045b425 8d ff ff MOV
0045b428 8d 15 00 LEA this=>local_24,[EBP + -0x20]
0045b42b 8d ff ff CALL dword ptr [->CPPASN1.DLL::ACE_SOCK_Stream::~ACE...
0045b42e 8b 8d cc MOV EAX,dword ptr [EBP + local_d38]
0045b434 99 29 JMP LAB_0045ddd4
0045b439 8d ff ff MOV EAX,dword ptr [EBP + local_234] XREF[1]: 0045b294(j)
0045b43f 50 PUSH EAX
0045b440 8d ff ff MOV this=>local_230,dword ptr [EBP + 0xfffffffdd4]
0045b446 51 PUSH this
0045b447 8d 8d a8 LEA this=>local_25c,[EBP + 0xfffffd8]
0045b44d 15 04 CALL dword ptr [->CPPASN1.DLL::SNACC::AsnBuf::AsnBuf]
0045b453 8d ff ff MOV byte ptr [EBP + local_8],0x13
0045b457 8b 95 d0 MOV EDX,dword ptr [EBP + local_224],EDX
0045b45d 8d ff ff MOV
0045b463 8d ff ff MOV dword ptr [EBP + local_22k],EDX
0045b463 8d bd d4 CMP dword ptr [EBP + local_230],0x0
0045b46a 74 1a JZ LAB_0045b486
0045b46c 8b 8d 40 MOV EAX,dword ptr [EBP + local_230]

```

Decompiler: FUN_0045bb0 - (dbman7.3_E0504P04.exe)

```

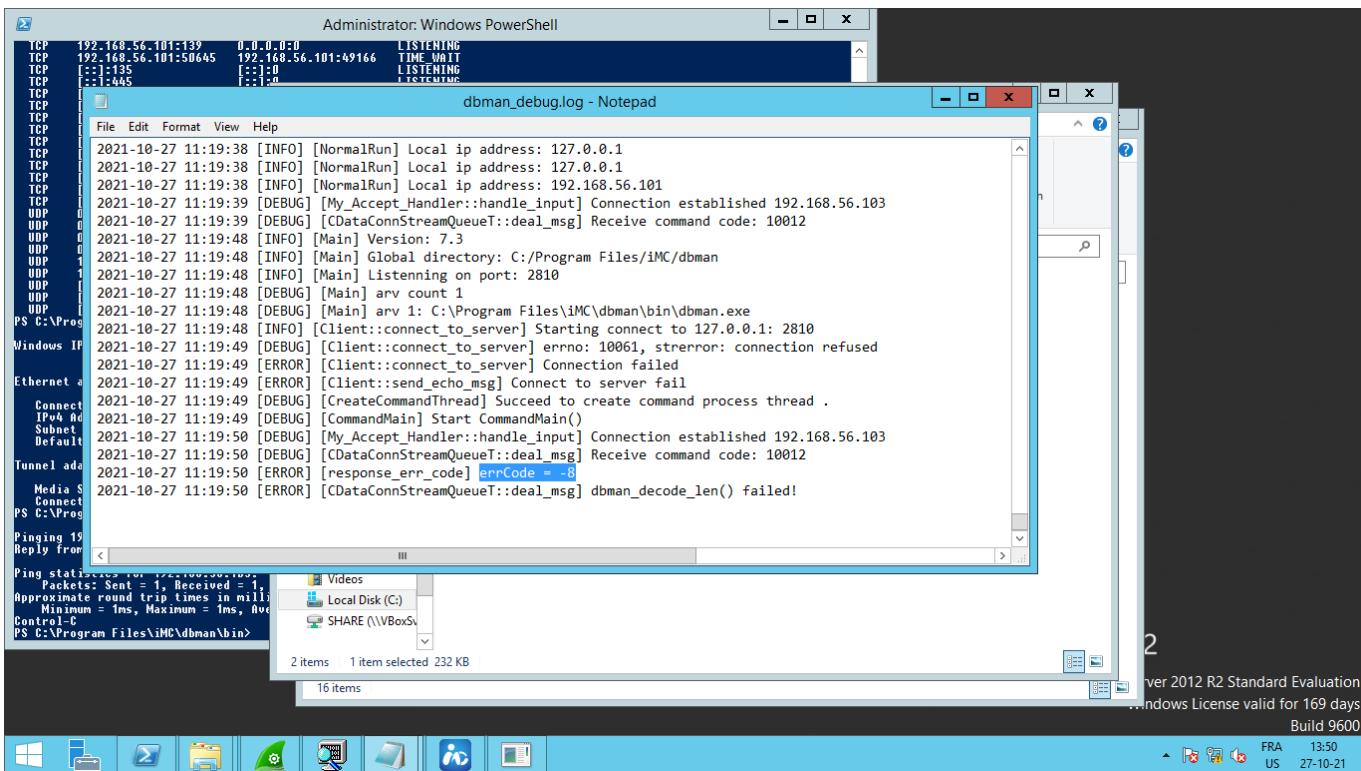
local_B_0_1 = &stack0xfffffe88;
SNACC::AsnBuf::AsnBuf((AsnEnum *)&stack0xfffffe88,1);
local_B_0_1 = 0x12;
pAvard = (ACE_SOCK_Stream *)FUN_0045b70((int)local_20);
local_18 = 0;
local_B_0_1 = 0x13 << 8;
FUN_00464510((local_20),"dbman_decode_len() failed!\n");
FUN_00474701((local_28),"dbman_decode_len() failed!\n");
local_d38 = 0;
local_d38 = 0xffffffff;
ACE_SOCK_Stream::ACE_SOCK_Stream((ACE_SOCK_Stream *)local_24);
ppuvar2 = extrastuff_EDX_28;
}
else {
    local_d28 = local_228;
    operator_delete((local_228));
    local_d2c = &stack0xfffffe88;
    SNACC::AsnBuf::AsnBuf((AsnEnum *)&stack0xfffffe88,1);
    FUN_00464510((ACE_SOCK_Stream *)local_24);
    FUN_00475401((local_20));
    local_18 = 0;
    local_18 = "Receive AsnBmanCmdCode::vtable REMOVE_RESERVED_FILE_REQ data_len
                error expect %d bytes infact %d bytes"
    local_d30 = 0xffffffff;
    local_d30 = 0xffffffff;
    ACE_SOCK_Stream::ACE_SOCK_Stream((ACE_SOCK_Stream *)local_24);
    ppuvar2 = extrastuff_EDX_26;
}
else {
}
```

Version Tracking Matches for Destination

| Score | Confid... | Votes | # C... | Source Na... | Source Label | Sour... | Dest... | Algorithm | |
|-------|-----------|-------|--------|--------------|--------------|----------|---------|-----------|---------------|
| 0.000 | 0.000 | 1 | 0 | Global | FUN_0045b70 | 00459790 | 16466 | 16796 | Implied Match |

First, we enter **FUN_00464510**:

which writes to the dbman log file `errCode = -8`:



The call to `FUN_00475470` :

Ghidra qterminal Thunar Version Tracking: diff2 10:09 AM

File Edit Window Help

Version Tracking Matches - [Session: diff2] 57141 matches

| Tag | Sess... | Status | Type | Score | Confidenc... | Votes | # Conflicting | Multipl... | Source Names... | Source Label | Source Addr... | Multi... | Dest Namespace | Dest Label | Dest Address | Source... | Dest Length | Algorithm |
|-----|---------|--------|----------|-------|--------------|-------|---------------|------------|-----------------|--------------|----------------|--------------|----------------|------------|--------------|---------------|-------------|-----------|
| | | | Function | 0.000 | 0.000 | 2 | 0 | Global | FUN_0047a060 | 0047a060 | Global | FUN_0047ab90 | 0047ab90 | 11 | 11 | Implied Match | | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00459790 | 00459790 | Global | FUN_00459bb0 | 00459bb0 | 16466 | 16796 | Implied Match | | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00402be0 | 00402be0 | Global | FUN_00402be0 | 00402be0 | 128 | 128 | Implied Match | | |
| | | | Data | 0.000 | 0.000 | 1 | 0 | Global | DAT_000b37b4 | 004b37b4 | Global | DAT_004047bc | 004b47bc | 1 | 1 | Implied Match | | |
| | | | Function | 0.000 | 0.000 | 1 | 0 | Global | FUN_00492d20 | 00492d20 | Global | FUN_00493850 | 00493850 | 105 | 105 | Implied Match | | |

Filter: Score Filter: 0.000 to 1.000 Confidence Filter: -9.999 to 9.999 Length Filter: 0

Version Tracking Markup Items - [Session: diff2]-25 markup items

| Status | Source Address | Dest Address | Markup Type | Source Value | Current Dest Value | Original Dest Value |
|--------|----------------|--------------|--------------------|---|---|---|
| OK | 00459790 | 00459bb0 | Function Signature | undefined _thiscall FUN_00459790(void *...) | undefined _thiscall FUN_00459bb0(void *...) | undefined _thiscall FUN_00459bb0(void *...) |

Filter:

Decompile View Listing View

Source: FUN_00459790 [/dbman7.3_E0504.exe]

```

1142
1143     break;
1144 case 0x271c:
1145     local_1c = ACE_SOCK_IO::recv((ACE_SOCK_IO *)local_24,&local_14,4,
1146     (ACE_Time_Value *)0x0);
1147     if (local_1c == 4) {
1148         local_1d = ((local_14 & 0xff) << 8 | (int)(local_14 & 0xffff00) >> 8) << 0x10 |
1149         ((int)local_14 >> 0x10 & 0xffff) << 8 | 0x1;
1150         (int)(int)local_14 >> 0x10 & 0xffff00) >> 8;
1151     local_c0 = (char *)FUN_0049a220(local_14);
1152     local_224 = local_c0;
1153     local_1c = ACE_SOCK_IO::recv((ACE_SOCK_IO *)local_24,&local_c0,local_14,
1154     (ACE_Time_Value *)0x0);
1155     if (local_1c == local_1d) {
1156         SNACC::AsnBuf::asnBuf(local_250,local_224,local_14);
1157         local_228 = local_1c;
1158         local_229 = local_228;
1159         operator_delete[](local_224);
1160         FUN_0041e190(local_220);
1161         local_8_0_1 = 0x13;
1162         FUN_0047c10(local_220,local_250,&local_228);
1163         FUN_0045ec20(int)local_220;
1164         local_18_0 = 0;
1165         ACE_SOCK_IO::send((ACE_SOCK_IO *)local_24,&local_18,4,(ACE_Time_Value *)0x0);
1166         local_d04 = 0xffffffff;
1167     }

```

Destination: FUN_00459bb0 [/dbman7.3_E0504P04.exe]

```

1194     }
1195     else {
1196         local_d34 = &stack0xfffffe8c;
1197         SNACC::AsnEnum::AsnEnum((AsnEnum *)&stack0xfffffe8c,1);
1198         local_8_0_1 = 0x12;
1199         pVar4 = (ACE_SOCK_Stream *)FUN_0045e70((int)local_20);
1200         local_1d = (uint)local_8_0_1_3 << 8;
1201         FUN_0045ec20(local_20);
1202         FUN_0047c10(local_28,"dbman_decode_len() failed!\n");
1203         FUN_0045ec20(local_20);
1204         local_d38 = 0;
1205         local_8_0_1_3 = 0xffffffff;
1206         ACE_SOCK_Stream::ACE_SOCK_Stream((ACE_SOCK_Stream *)local_24);
1207         ppVar2 = &extractEDK_27;
1208     }
1209 }
1210 else {
1211     local_d28 = local_228;
1212     operator_delete[](local_228);
1213     local_d2c = &stack0xfffffe8c;
1214     SNACC::AsnEnum::AsnEnum((AsnEnum *)&stack0xfffffe8c,1);
1215     FUN_0045ec20((ACE_SOCK_Stream *)local_24);
1216     FUN_0047c10(local_28,
1217         "Receive AsnHandleCedCode:1MSG_V001_REMOVE_RESERVED_FILE_REQ data_len
1218         error expect %d bytes infact %d bytes");
}

```

which writes `dbman_decode_len() failed` error to the log:

Administrator: Windows PowerShell

```

TCP 192.168.56.101:139 0.0.0.0.0 LISTENING
TCP 192.168.56.101:50645 192.168.56.101:49166 TIME_WAIT
TCP [-]:135 [-]:* LISTENING
TCP [-]:1445 [-]:* LISTENING
TCP TCP dbman_debug.log - Notepad
TCP TCP File Edit Format View Help
TCP TCP 2021-10-27 11:19:38 [INFO] [NormalRun] Local ip address: 127.0.0.1
TCP TCP 2021-10-27 11:19:38 [INFO] [NormalRun] Local ip address: 127.0.0.1
TCP TCP 2021-10-27 11:19:38 [INFO] [NormalRun] Local ip address: 192.168.56.101
TCP TCP 2021-10-27 11:19:39 [DEBUG] [My_Accept_Handler::handle_input] Connection established 192.168.56.103
TCP UDP 2021-10-27 11:19:39 [DEBUG] [CDataConnStreamQueueT::deal_msg] Receive command code: 10012
TCP UDP 2021-10-27 11:19:48 [INFO] [Main] Version: 7.3
TCP UDP 2021-10-27 11:19:48 [INFO] [Main] Global directory: C:/Program Files/iMC/dbman
TCP UDP 2021-10-27 11:19:48 [INFO] [Main] Listenning on port: 2810
TCP UDP 2021-10-27 11:19:48 [DEBUG] [Main] arv count 1
PS C:\Prog... 2021-10-27 11:19:48 [DEBUG] [Main] arv 1: C:\Program Files\iMC\dbman\bin\dbman.exe
Windows IP 2021-10-27 11:19:48 [INFO] [Client::connect_to_server] Starting connect to 127.0.0.1: 2810
2021-10-27 11:19:49 [DEBUG] [Client::connect_to_server] errno: 10061, strerror: connection refused
Ethernet a 2021-10-27 11:19:49 [ERROR] [Client::connect_to_server] Connection failed
Connect 2021-10-27 11:19:49 [ERROR] [Client::send_echo_msg] Connect to server fail
IPV4 Ad... 2021-10-27 11:19:49 [DEBUG] [CreateCommandThread] Succeed to create command process thread .
Subset 2021-10-27 11:19:49 [DEBUG] [CommandMain] Start CommandMain()
Default 2021-10-27 11:19:50 [DEBUG] [Client::connect_to_server] errno: 10061, strerror: connection refused
Tunnel ad... 2021-10-27 11:19:50 [DEBUG] [CDataConnStreamQueueT::deal_msg] Receive command code: 10012
Media S 2021-10-27 11:19:50 [ERROR] [response_err_code] errCode = -8
PS C:\Prog... 2021-10-27 11:19:50 [ERROR] [CDataConnStreamQueueT::deal_msg] dbman_decode_len() failed!
Pinging 19
Reply from < >:
Ping statistics for 192.168.56.103:
    Packets: Sent = 1, Received = 1,
    Approximate round trip times in milli:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
Control-C
PS C:\Program Files\iMC\dbman\bin>

```

2 over 2012 R2 Standard Evaluation
Windows License valid for 169 days
Build 9600

16 items

2 items 1 item selected 232 KB

Windows Taskbar icons: Videos, Local Disk (C), SHARE (\VBoxSv)

15:40 FRA US 27-10-21

We're still in the `else` clause, we'll end up hitting `FUN_0045ec20`:

```

0045ec20: FUN_0045ec20
    undefined4 __fastcall FUN_0045ec20(undefined **param_1)
    {
        undefined4 EAX:4           ->RETURN;
        undefined4 * ECX:4         param_1;
        undefined4 Stack[-0x8]:dword local_8;
        undefined4 Stack[-0x10]:dword local_10;
        undefined4 Stack[-0x10]:dword local_10;
        undefined4 Stack[-0x14]:dword local_14;

        PUSH EAX
        .ec20 PUSH EBP
        .ec21 MOV EBP,ESP
        .ec22 SUB ESP,0x10
        .ec23 MOV EBP,local_10
        .ec24 MOV EAX,dword ptr [EBP + local_10]
        .ec25 MOV EAX,dword ptr [EBP + local_10]
        .ec26 PUSH EAX
        .ec27 CALL FUN_0045eb0
        .ec28 MOV param_1,dword ptr [EBP + local_10]
        .ec29 MOV EAX,dword ptr [EBP + local_10]
        .ec30 MOV EAX,dword ptr [EBP + local_10]
        .ec31 MOV EAX,dword ptr [EBP + local_10]
        .ec32 MOV param_1,dword ptr [EBP + local_10]
        .ec33 CALL dword ptr [-ACE_V6_DLL::ACE_SOCK_Stream::close(ACE_SOCK_Stream *)]
        .ec34 CALL FUN_0045fb0
        .ec35 MOV EAX,dword ptr [EBP + local_10]
        .ec36 MOV EAX,dword ptr [EBP + local_10]
        .ec37 MOV EAX,dword ptr [EBP + local_10]
        .ec38 MOV EAX,dword ptr [EBP + local_10]
        .ec39 CALL dword ptr [-ACE_V6_DLL::ACE_SOCK_Stream::close(ACE_SOCK_Stream *)]
        .ec40 MOV EAX,dword ptr [EBP + local_10]
        .ec41 MOV EAX,dword ptr [EBP + local_10]
        .ec42 MOV EAX,dword ptr [EBP + local_10]
        .ec43 MOV EAX,dword ptr [EBP + local_10]
        .ec44 MOV EAX,dword ptr [EBP + local_10]
        .ec45 MOV EAX,dword ptr [EBP + local_10]
        .ec46 MOV EAX,dword ptr [EBP + local_10]
        .ec47 CMP EAX,local_10
        .ec48 JNE .ec71
        .ec49 POP EBP
        .ec50 RET
    }

```

The `CALL EAX` needs to be resolved dynamically:

```

0045ec20: FUN_0045ec20
    undefined4 __fastcall FUN_0045ec20(undefined **param_1)
    {
        undefined4 EAX:4           ->RETURN;
        undefined4 * ECX:4         param_1;
        undefined4 Stack[-0x8]:dword local_8;
        undefined4 Stack[-0x10]:dword local_10;
        undefined4 Stack[-0x10]:dword local_10;
        undefined4 Stack[-0x14]:dword local_14;

        PUSH EAX
        .ec20 PUSH EBP
        .ec21 MOV EBP,ESP
        .ec22 SUB ESP,0x10
        .ec23 MOV EBP,local_10
        .ec24 MOV EAX,dword ptr [EBP + local_10],param_1
        .ec25 MOV EAX,dword ptr [EBP + local_10]
        .ec26 PUSH EAX
        .ec27 CALL FUN_00440840
        .ec28 MOV param_1,EAX
        .ec29 CALL FUN_0045fb0
        .ec30 MOV param_1,dword ptr [EBP + local_10]
        .ec31 ADD param_1,local_10
        .ec32 MOV EAX,dword ptr [-ACE_V6_DLL::ACE_SOCK_Stream::close(ACE_SOCK_Stream *)]
        .ec33 ADD param_1,local_10
        .ec34 MOV EAX,dword ptr [EBP + local_10]
        .ec35 MOV EAX,dword ptr [EBP + local_10]
        .ec36 MOV EAX,dword ptr [EBP + local_10]
        .ec37 MOV EAX,dword ptr [EBP + local_10]
        .ec38 MOV EAX,dword ptr [EBP + local_10]
        .ec39 MOV EAX,dword ptr [EBP + local_10]
        .ec40 MOV EAX,dword ptr [EBP + local_10]
        .ec41 MOV EAX,dword ptr [EBP + local_10]
        .ec42 MOV EAX,dword ptr [EBP + local_10]
        .ec43 MOV EAX,dword ptr [EBP + local_10]
        .ec44 MOV EAX,dword ptr [EBP + local_10]
        .ec45 MOV EAX,dword ptr [EBP + local_10]
        .ec46 MOV EAX,dword ptr [EBP + local_10]
        .ec47 MOV EAX,dword ptr [EBP + local_10]
        .ec48 MOV EAX,dword ptr [EBP + local_10]
        .ec49 MOV EAX,dword ptr [EBP + local_10]
        .ec50 MOV EAX,dword ptr [EBP + local_10]
        .ec51 CMP EAX,local_10
        .ec52 JZ .ec71
        .ec53 MOV EAX,dword ptr [EBP + local_10]
        .ec54 ADD EAX,local_10
        .ec55 MOV EAX,dword ptr [EBP + local_10]
        .ec56 ADD EAX,local_10
        .ec57 PUSH EBP
        .ec58 MOV EAX,dword ptr [EBP + local_10]
        .ec59 ADD EAX,local_10
        .ec60 MOV EAX,dword ptr [EBP + local_10]
        .ec61 ADD EAX,local_10
        .ec62 MOV EAX,dword ptr [EBP + local_10]
        .ec63 ADD EAX,local_10
        .ec64 MOV EAX,dword ptr [EBP + local_10]
        .ec65 ADD EAX,local_10
        .ec66 MOV EAX,dword ptr [EBP + local_10]
        .ec67 ADD EAX,local_10
        .ec68 MOV EAX,dword ptr [EBP + local_10]
        .ec69 ADD EAX,local_10
        .ec70 MOV EAX,dword ptr [EBP + local_10]
        .ec71 OR EAX,0xffffffff
        .ec72 MOV EBP,ESP
        .ec73 POP EBP
        .ec74 RET
    }

0045ec60: FUN_0045ec60
    undefined4 __fastcall FUN_0045ec60(undefined **param_1)
    {
        undefined4 EAX:4           ->FUN_0045ec20;
        undefined4 * ECX:4         param_1;
        undefined4 Stack[-0x8]:dword local_8;
        undefined4 Stack[-0x10]:dword local_10;
        undefined4 Stack[-0x10]:dword local_10;
        undefined4 Stack[-0x14]:dword local_14;

        ACE_Cleanup + FUN_0045ec20
        .ec20 PUSH EBP
        .ec21 MOV EBP,ESP
        .ec22 SUB ESP,0x10
        .ec23 MOV EBP,local_10
        .ec24 MOV EAX,dword ptr [EBP + local_10],param_1
        .ec25 MOV EAX,dword ptr [EBP + local_10]
        .ec26 PUSH EAX
        .ec27 CALL FUN_00440840
        .ec28 MOV param_1,EAX
        .ec29 CALL FUN_0045fb0
        .ec30 MOV param_1,dword ptr [EBP + local_10]
        .ec31 ADD param_1,local_10
        .ec32 MOV EAX,dword ptr [-ACE_V6_DLL::ACE_SOCK_Stream::close(ACE_SOCK_Stream *)]
        .ec33 ADD param_1,local_10
        .ec34 MOV EAX,dword ptr [EBP + local_10]
        .ec35 MOV EAX,dword ptr [EBP + local_10]
        .ec36 MOV EAX,dword ptr [EBP + local_10]
        .ec37 MOV EAX,dword ptr [EBP + local_10]
        .ec38 MOV EAX,dword ptr [EBP + local_10]
        .ec39 MOV EAX,dword ptr [EBP + local_10]
        .ec40 MOV EAX,dword ptr [EBP + local_10]
        .ec41 MOV EAX,dword ptr [EBP + local_10]
        .ec42 MOV EAX,dword ptr [EBP + local_10]
        .ec43 MOV EAX,dword ptr [EBP + local_10]
        .ec44 MOV EAX,dword ptr [EBP + local_10]
        .ec45 MOV EAX,dword ptr [EBP + local_10]
        .ec46 MOV EAX,dword ptr [EBP + local_10]
        .ec47 MOV EAX,dword ptr [EBP + local_10]
        .ec48 MOV EAX,dword ptr [EBP + local_10]
        .ec49 MOV EAX,dword ptr [EBP + local_10]
        .ec50 MOV EAX,dword ptr [EBP + local_10]
        .ec51 CMP EAX,local_10
        .ec52 JZ .ec71
        .ec53 MOV EAX,dword ptr [EBP + local_10]
        .ec54 ADD EAX,local_10
        .ec55 MOV EAX,dword ptr [EBP + local_10]
        .ec56 ADD EAX,local_10
        .ec57 PUSH EBP
        .ec58 MOV EAX,dword ptr [EBP + local_10]
        .ec59 ADD EAX,local_10
        .ec60 MOV EAX,dword ptr [EBP + local_10]
        .ec61 ADD EAX,local_10
        .ec62 MOV EAX,dword ptr [EBP + local_10]
        .ec63 ADD EAX,local_10
        .ec64 MOV EAX,dword ptr [EBP + local_10]
        .ec65 ADD EAX,local_10
        .ec66 MOV EAX,dword ptr [EBP + local_10]
        .ec67 ADD EAX,local_10
        .ec68 MOV EAX,dword ptr [EBP + local_10]
        .ec69 ADD EAX,local_10
        .ec70 MOV EAX,dword ptr [EBP + local_10]
        .ec71 OR EAX,0xffffffff
        .ec72 MOV EBP,ESP
        .ec73 POP EBP
        .ec74 RET
    }

```

The `CALL EAX` needs to be resolved dynamically:

Looking/breakpoint for `FUN_0045ec20` at `0045b40f`:

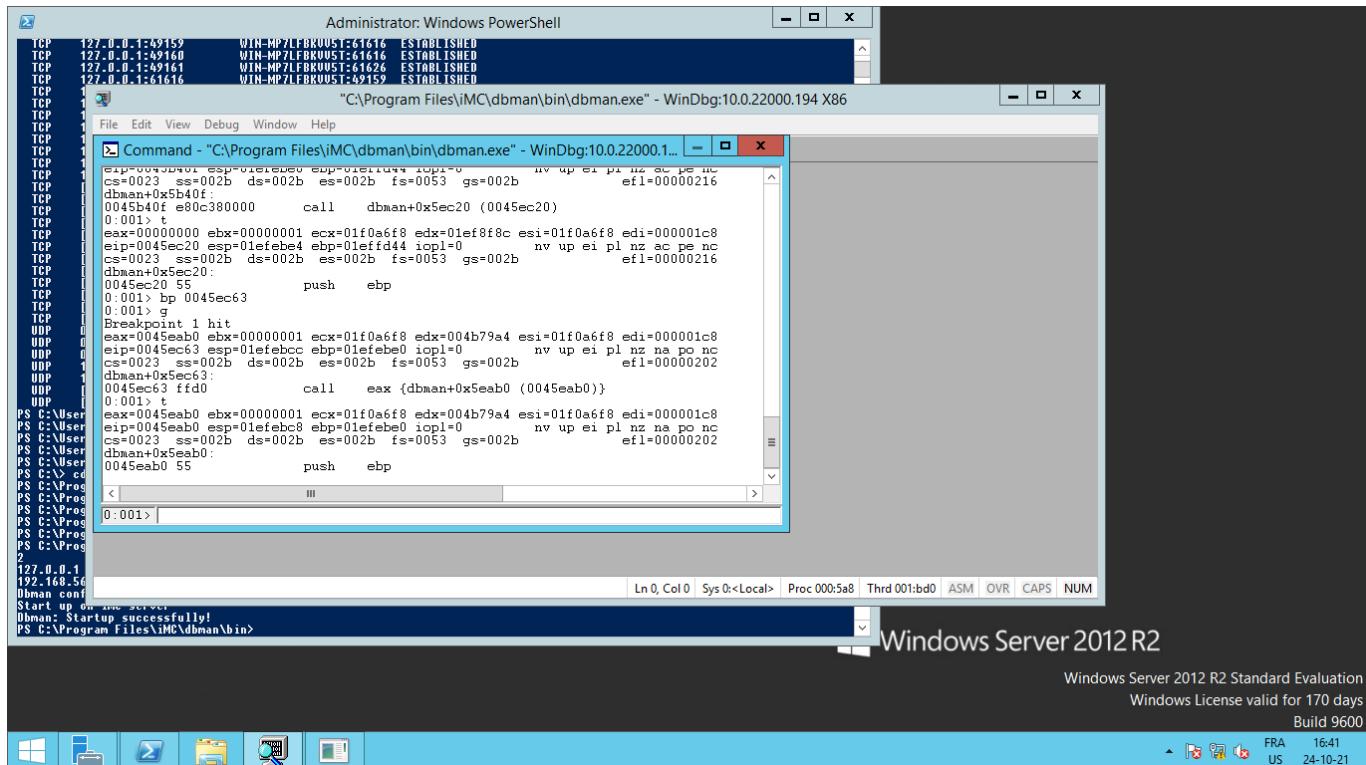
The screenshot shows a Windows Server 2012 R2 Standard Evaluation system. A WinDbg session is running under the Administrator: Windows PowerShell prompt. The command window displays assembly code and memory dump information. A Task Manager window is visible in the background, showing a list of running processes. The Task Manager window has a status bar indicating "Windows License valid for 170 days".

Next, we're looking for `CALL EAX` instruction which is located at: `0045ec63`.

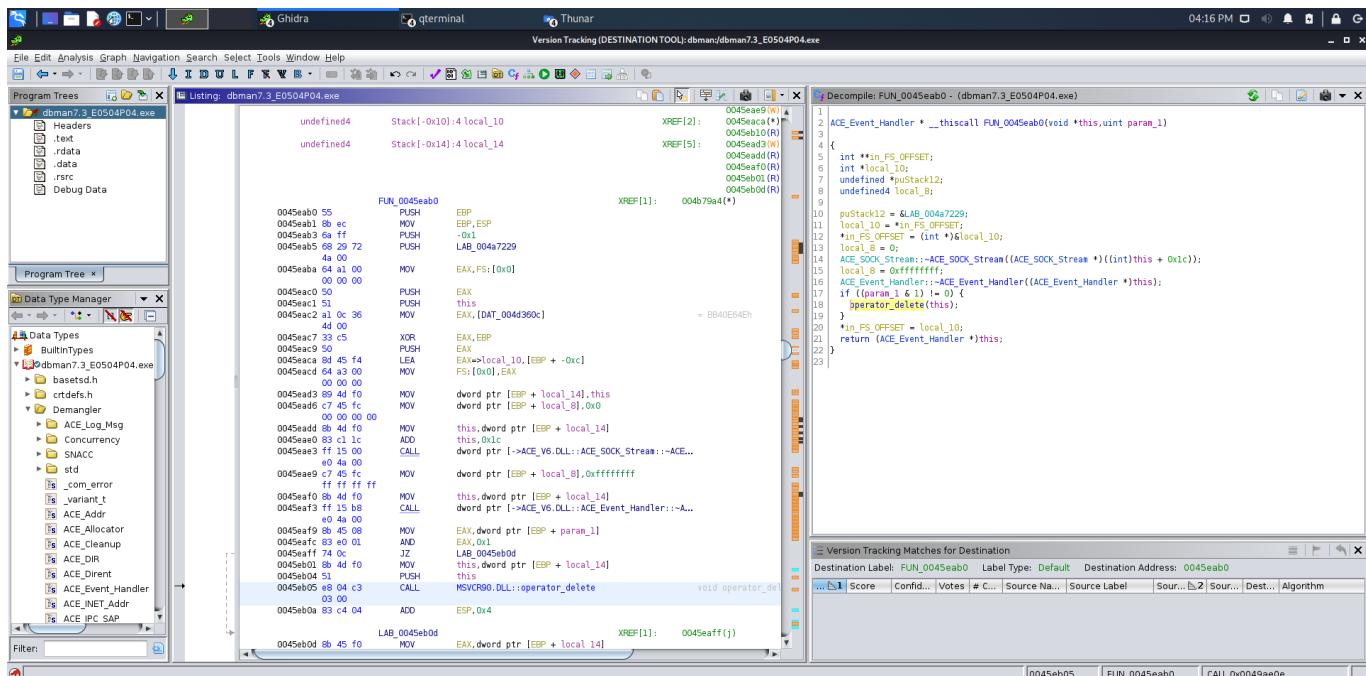
Looking at 0045ec63 from windbg:

The screenshot shows a Windows Server 2012 R2 desktop environment. The taskbar at the bottom includes icons for File Explorer, Task View, Start, and several pinned application icons. The main window is a PowerShell session titled "Administrator: Windows PowerShell". The command line shows the path to the dbman executable and its memory dump file. Below the command line, the WinDbg debugger interface is visible, displaying assembly code and registers. A status bar at the bottom of the debugger window provides information about the current assembly, processor, and keyboard settings. The desktop background features the standard Windows Server 2012 R2 wallpaper.

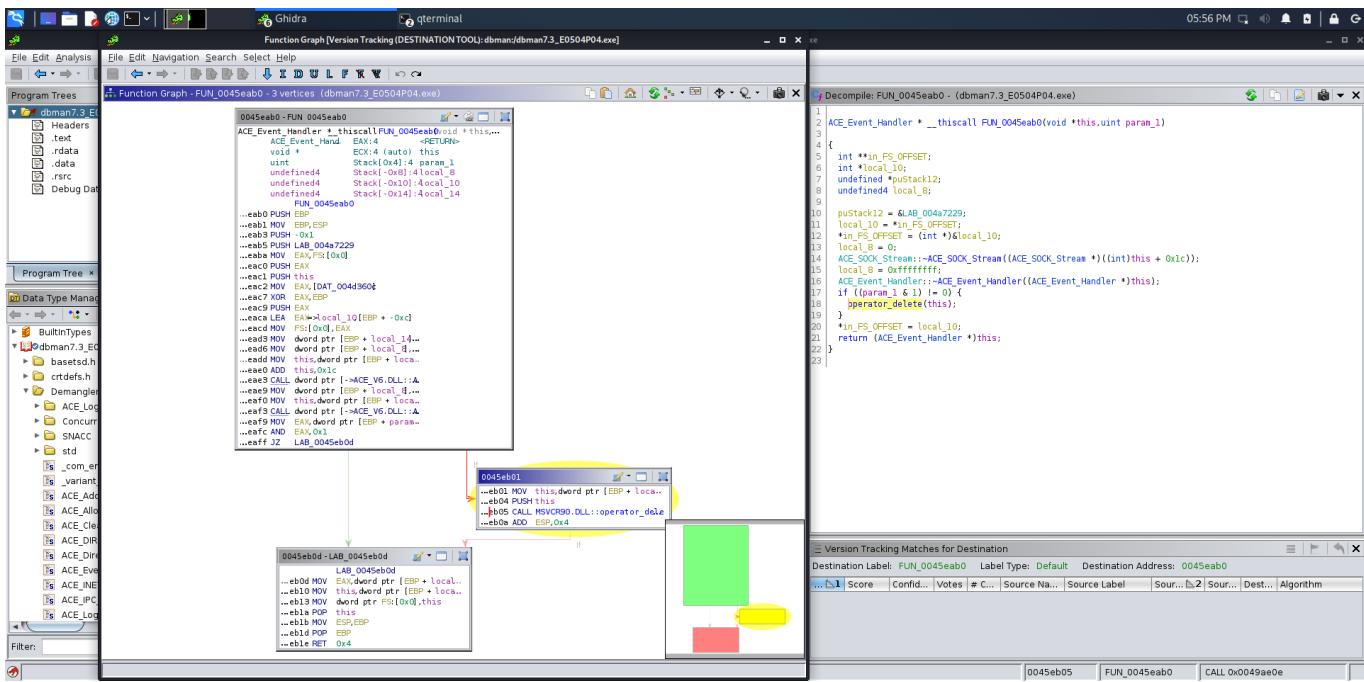
The next instruction after the `call` will resolve for `0045eab0`:



We see that it's calling `FUN_0045eb0()` (where `operator_delete` is @0045eb05), which in Ghidra decompiler gives:



Graph view of it:



Looking at it(`FUN_0045eb0`) in windbg:

The screenshot shows the WinDbg command window displaying the assembly dump of `FUN_0045eb0`. The assembly code is shown in columns, with labels and comments above the instructions. The code includes various pushes, moves, and calls to standard library functions like `operator new` and `operator delete`.

```

0:001> u 0045eb00
dbman+0x5eb00:
0045eb00  push    ebp
0045eb01 8bec   mov     esp,ebp
0045eb03 6aff   push    offset dbman!std::_Init_locks::operator=+0xc61f (004a7229)
0045eb05 6829724a00 push    offset dbman!std::_Init_locks::operator=+0xc61f (004a7229)
0045eb07 64a100000000 mov     eax,dword ptr fs:[00000000h]
0045eb09 50      push    eax
0045eb0c 51      push    ecx
0045eb0d a10c364d00 mov     eax,dword ptr [dbman!std::_Init_locks::operator=+0x38a02 (004d360c)]
0:001> u 0045eb00
dbman+0x5eb01:
0045eb01 55      push    ebp
0045eb02 8bec   mov     esp,ebp
0045eb04 6aff   push    offset dbman!std::_Init_locks::operator=+0xc61f (004a7229)
0045eb06 6829724a00 push    offset dbman!std::_Init_locks::operator=+0xc61f (004a7229)
0045eb08 64a100000000 mov     eax,dword ptr fs:[00000000h]
0045eb0a 50      push    eax
0045eb0c 51      push    ecx
0045eb0d a10c364d00 mov     eax,dword ptr [dbman!std::_Init_locks::operator=+0x38a02 (004d360c)]
0:001> u 0045eb00
dbman+0x5eb02:
0045eb02 55      push    ebp
0045eb03 8bec   mov     esp,ebp
0045eb05 6aff   push    offset dbman!std::_Init_locks::operator=+0xc61f (004a7229)
0045eb07 64a300000000 mov     eax,dword ptr fs:[00000000h]
0045eb09 84df00 mov     dword ptr [ebp-10h],ecx
0045eb0b c745fc00000000 mov     dword ptr [ebp-4],0
0045eb0d 854000 mov     ecx,dword ptr [ebp-10h]
0045eb0f 83c100 add    eax,1
0045eb11 f1150004a00 call    dword ptr [dbman!std::_Init_locks::operator=+0x133f6 (004ae000)]
0045eb13 c745fcfffff mov     dword ptr [ebp-4],0xffffffff
0045eb15 8b4df0 call    dword ptr [ebp-10h]
0045eb17 ff15b8e04a00 call    dword ptr [dbman!std::_Init_locks::operator=+0x134ae (004ae0b8)]
0045eb19 8b4508 mov     eax,dword ptr [ebp+8]
0045eb1b 83e001 and    eax,1
0045eb1d 740c je     dbman+0x5eb0d (0045eb0d) Branch
dbman+0x5eb01:
0045eb01 8b4df0 mov     ecx,dword ptr [ebp-10h]
0045eb04 51      push    ecx
0045eb05 e804c30300 call    dbman!std::_Init_locks::operator=+0x204 (0049ae0e)
0045eb07 83c404 add    esp,4
dbman+0x5eb0d:
0045eb0d 8b45f0 mov     eax,dword ptr [ebp-10h]
0045eb10 8b4df4 mov     ecx,dword ptr [ebp-0Ch]
0045eb13 64890d00000000 mov     dword ptr fs:[0].ecx
0045eb15 59      pop    ecx
0045eb1b 8be5 mov     esp,ebp
0045eb1d 5d      pop    ebp
0045eb1e c20400 ret    4

```

Windbg: breaking @0045eb05 (`operator_delete`):

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
ModLoad: 71420000 71432000 C:\Windows\SysWOW64\NETAPI32.dll
ModLoad: 75140000 7514e000 C:\Windows\SysWOW64\combase.dll
ModLoad: 75820000 758de000 C:\Windows\SysWOW64\asvcrt.dll
ModLoad: 77140000 771f1000 C:\Windows\SysWOW64\RPCRT4.dll
ModLoad: 758e0000 7591e000 C:\Windows\SysWOW64\sechost.dll
ModLoad: 74ef0000 74ff8000 C:\Windows\SysWOW64\GDI32.dll
ModLoad: 75d50000 75d53000 C:\Windows\SysWOW64\NSL.dll
ModLoad: 72490000 72493000 C:\Windows\SysWOW64\RPCRT4.dll
ModLoad: 74490000 74493000 C:\Windows\SysWOW64\netutils.dll
ModLoad: 71410000 7141b000 C:\Windows\SysWOW64\avrcv.dll
ModLoad: 713e0000 71400000 C:\Windows\SysWOW64\vksccli.dll
ModLoad: 74dd0000 74dd1000 C:\Windows\SysWOW64\Spicli.dll
ModLoad: 74cf0000 74cf9000 C:\Windows\SysWOW64\CRYPTBASE.dll
ModLoad: 74c90000 74c93000 C:\Windows\SysWOW64\bcryptPrimitives.dll
(4b0:28c): Break instruction exception - code 80000003 (first chance)
eax=7737da8 esp=0018faec ebp=0018fb18 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000246
ntdll!_drpDbgBreak+0xb:
7737da8 cc          int     3
0:000>bp 0045eb05
0:000>g
ModLoad: 75920000 75945000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75ab0000 75ba7000 C:\Windows\SysWOW64\MSCTF.dll
Breakpoint 0 hit
eax=00000001 ebx=00000000 ecx=1bb10000 edx=00000000 esi=7ffd000 edi=00000000
eip=0045eb05 esp=01dfebac ebp=01dfebc4 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
dbman!_Init_locks@0x5eb05:
0045eb05 e804c30300 call    dbman!std::_Init_locks::operator+0x204 (0049ae0e)
0:001 u
dbman!_Init_locks@0x5eb05:
0045eb05 e804c30300 call    dbman!std::_Init_locks::operator+0x204 (0049ae0e)
0045eb05 83c404 add    esp, 4
0045eb05 8b45f0 mov    eax, dword ptr [ebp-10h]
0045eb05 8b4df4 mov    ecx, dword ptr [ebp-0Ch]
0045eb13 64890d00000000 mov    dword ptr fs:[0].ecx
0045eb1a 59 pop    ecx
0045eb1b 8b55 mov    esp, ebp
0045eb1d 5d pop    ebp
0:001> u 0049ae0e
dbman!std::_Init_locks::operator+0x204:
0049ae0f ff252ce54a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x13922 (004ae52c)]
0049ae14 ff25d0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137c6 (004ae3d0)]
0049ae20 ff25d8e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137ca (004ae3d4)]
0049ae26 ff25dce34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d2 (004ae3dc)]
0049ae2c ff25e0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d6 (004ae3e0)]
0049ae32 3b0d0c364d00 cmp    ecx, dword ptr [dbman!std::_Init_locks::operator+0x38a02 (004d360c)]
0049ae38 7502 jne    dbman!std::_Init_locks::operator+0x232 (0049ae3c)
0:001> t
eax=00000001 ebx=00000001 ecx=01eae000 edx=004b79a4 esi=01eae00 edi=000000150
eip=0049ae0e esp=01dfeb08 ebp=01dfebc4 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
dbman!std::_Init_locks::operator+0x204:
0049ae0f ff252ce54a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x13922 (004ae52c)] ds:002b:004ae52c=MSVCR90!operator delete (6ff03f03)
0:001 u
dbman!std::_Init_locks::operator+0x204:
0049ae14 ff25d0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x13922 (004ae52c)]
0049ae1a ff25d8e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137c6 (004ae3d0)]
0049ae20 ff25dce34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137ca (004ae3d4)]
0049ae26 ff25e0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d2 (004ae3dc)]
0049ae2c ff25e0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d6 (004ae3e0)]
0049ae32 3b0d0c364d00 cmp    ecx, dword ptr [dbman!std::_Init_locks::operator+0x38a02 (004d360c)]
0049ae38 7502 jne    dbman!std::_Init_locks::operator+0x232 (0049ae3c)
0:001> t
eax=00000001 ebx=00000001 ecx=01eae000 edx=004b79a4 esi=01eae00 edi=000000150
0:001> 6ff03f03 esp=01dfeb08 ebp=01dfebc4 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
MSVCR90!operator delete:
6ff03f03 8bf1 mov    edi,edi
0:001 u
MSVCR90!operator delete:
6ff03f03 8bf1 mov    edi,edi
6ff03f05 55 push    ebp
6ff03f08 8bec mov    esp,ebp
6ff03f08 5d pop    ebp
6ff03f09 e940fcffff jmp    MSVCR90!free (6ff03b4e)
6ff03f0e cc        int     3
6ff03f0f cc        int     3
6ff03f10 cc        int     3

```

It looks like `operator delete` makes use of `free`:

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
0045eb05 e804c30300 call    dbman!std::_Init_locks::operator+0x204 (0049ae0e)
0045eb05 83c404 add    esp, 4
0045eb05 8b45f0 mov    eax, dword ptr [ebp-10h]
0045eb05 8b4df4 mov    ecx, dword ptr [ebp-0Ch]
0045eb13 64890d00000000 mov    dword ptr fs:[0].ecx
0045eb1a 59 pop    ecx
0045eb1b 8b55 mov    esp, ebp
0045eb1d 5d pop    ebp
0:001 u 0049ae0e
dbman!std::_Init_locks::operator+0x204:
0049ae0f ff252ce54a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x13922 (004ae52c)]
0049ae14 ff25d0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137c6 (004ae3d0)]
0049ae20 ff25d8e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137ca (004ae3d4)]
0049ae26 ff25dce34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d2 (004ae3dc)]
0049ae2c ff25e0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d6 (004ae3e0)]
0049ae32 3b0d0c364d00 cmp    ecx, dword ptr [dbman!std::_Init_locks::operator+0x38a02 (004d360c)]
0049ae38 7502 jne    dbman!std::_Init_locks::operator+0x232 (0049ae3c)
0:001> t
eax=00000001 ebx=00000001 ecx=01eae000 edx=004b79a4 esi=01eae00 edi=000000150
eip=0049ae0e esp=01dfeb08 ebp=01dfebc4 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
dbman!std::_Init_locks::operator+0x204:
0049ae0f ff252ce54a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x13922 (004ae52c)] ds:002b:004ae52c=MSVCR90!operator delete (6ff03f03)
0:001 u
dbman!std::_Init_locks::operator+0x204:
0049ae14 ff25d0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x13922 (004ae52c)]
0049ae1a ff25d8e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137c6 (004ae3d0)]
0049ae20 ff25dce34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137ca (004ae3d4)]
0049ae26 ff25e0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d2 (004ae3dc)]
0049ae2c ff25e0e34a00 jmp    dword ptr [dbman!std::_Init_locks::operator+0x137d6 (004ae3e0)]
0049ae32 3b0d0c364d00 cmp    ecx, dword ptr [dbman!std::_Init_locks::operator+0x38a02 (004d360c)]
0049ae38 7502 jne    dbman!std::_Init_locks::operator+0x232 (0049ae3c)
0:001> t
eax=00000001 ebx=00000001 ecx=01eae000 edx=004b79a4 esi=01eae00 edi=000000150
0:001> 6ff03f03 esp=01dfeb08 ebp=01dfebc4 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
MSVCR90!operator delete:
6ff03f03 8bf1 mov    edi,edi
6ff03f05 55 push    ebp
6ff03f08 8bec mov    esp,ebp
6ff03f08 5d pop    ebp
6ff03f09 e940fcffff jmp    MSVCR90!free (6ff03b4e)
6ff03f0e cc        int     3
6ff03f0f cc        int     3
6ff03f10 cc        int     3

```

Analyzing through gflags.exe from WindowsSDK:

At times, tracing/following the binary's flow can be cumbersome ; especially for this type of bug.

Using the *gflags.exe* utility can help by putting objects into a full page heap, which can be found at:

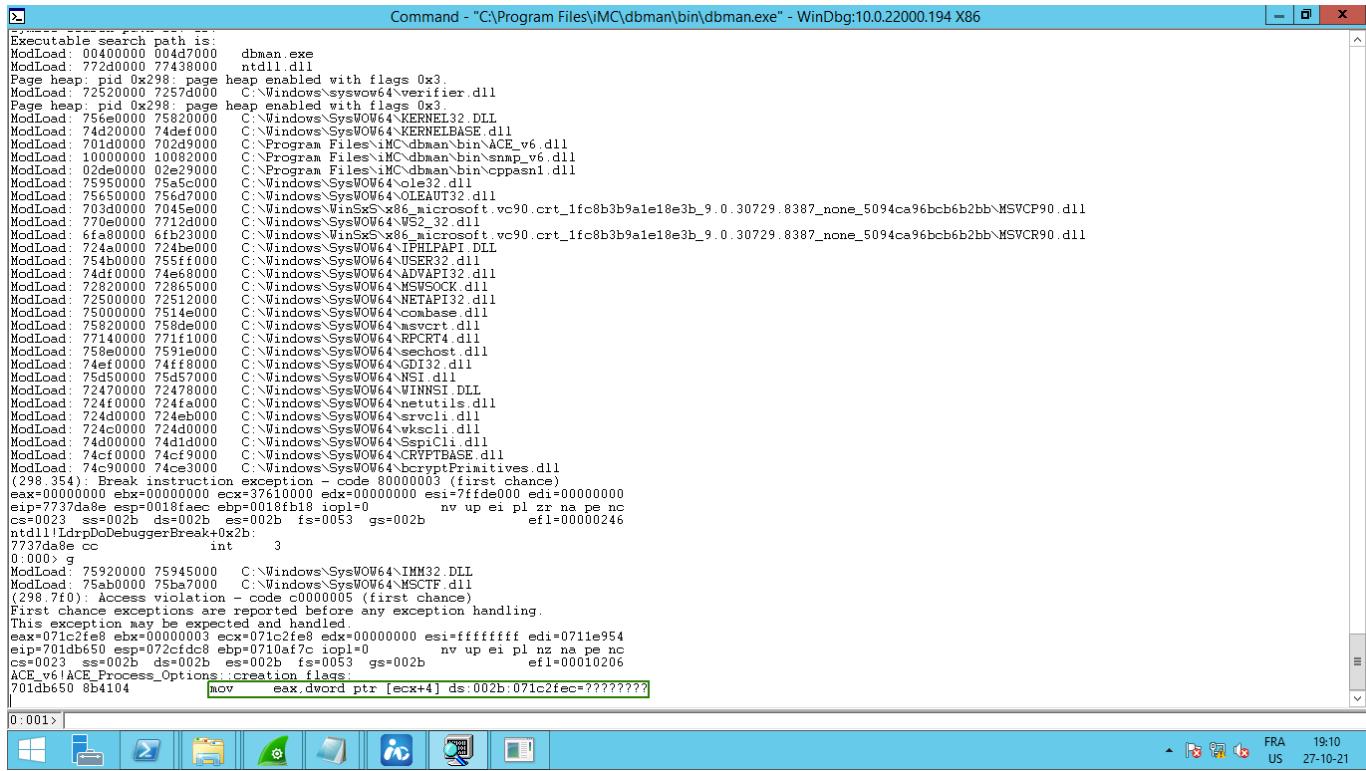
```
C:\Program Files (x86)\Windows Kits\10\Debuggers\x86\
```

Enabling *gflags* on *dbman*:

```
> .\gflags.exe /i "C:\Program Files\iMC\dbman\bin\dbman.exe" +hpa
```

A bit like when you're trying to workout an exploit "inside" a debugger, this *dbman* behaviour isn't identical to the "vanilla" one.

Observing the behaviour in windbg, *access violation*:



```
Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86

Executable search path is:
ModLoad: 00400000 004d7000 dbman.exe
ModLoad: 772d0000 77438000 ntdll.dll
Page heap: pid 0x298: page heap enabled with flags 0x3.
ModLoad: 72520000 7257d000 C:\Windows\syswow64\Verifier.dll
Page heap: pid 0x298: page heap enabled with flags 0x3.
ModLoad: 756e0000 75820000 C:\Windows\SysWOW64\KERNEL32.DLL
ModLoad: 74d40000 74def000 C:\Windows\SysWOW64\KERNELBASE.dll
ModLoad: 701d0000 70249000 C:\Program Files\iMC\dbman\bin\ACE_v6.dll
ModLoad: 75940000 75a00000 C:\Program Files\iMC\dbman\bin\smmp_v6.dll
ModLoad: 02de0000 02e29000 C:\Windows\Files\iMC\dbman\bin\cprasn1.dll
ModLoad: 75950000 75a5c000 C:\Windows\SysWOW64\ole32.dll
ModLoad: 75650000 756d7000 C:\Windows\SysWOW64\OLEAUT32.dll
ModLoad: 703d0000 7045e000 C:\Windows\WinSxS\x86_microsoft.vc90.crt_1fc8b3b9ale18e3b_9.0.30729.8387_none_5094ca96bcb6b2bb\MSVCP90.dll
ModLoad: 770e0000 7712d000 C:\Windows\SysWOW64\MS2_32.dll
ModLoad: 6fa80000 6fb23000 C:\Windows\WinSxS\x86_microsoft.vc90.crt_1fc8b3b9ale18e3b_9.0.30729.8387_none_5094ca96bcb6b2bb\MSVCR90.dll
ModLoad: 724a0000 724be000 C:\Windows\SysWOW64\IPHLPAPI.DLL
ModLoad: 754b0000 755ff000 C:\Windows\SysWOW64\USER32.dll
ModLoad: 74d40000 74e68000 C:\Windows\SysWOW64\ADVAPI32.dll
ModLoad: 72820000 72865000 C:\Windows\SysWOW64\WSOCK.dll
ModLoad: 72500000 72512000 C:\Windows\SysWOW64\NETAPI32.dll
ModLoad: 75930000 75a4e000 C:\Windows\SysWOW64\combase.dll
ModLoad: 75820000 75840000 C:\Windows\SysWOW64\RPCRT4.dll
ModLoad: 77140000 771f1000 C:\Windows\SysWOW64\olecrn.dll
ModLoad: 758e0000 7591e000 C:\Windows\SysWOW64\sechost.dll
ModLoad: 74ef0000 74ff8000 C:\Windows\SysWOW64\GDI32.dll
ModLoad: 75d50000 75d57000 C:\Windows\SysWOW64\NSI.dll
ModLoad: 72470000 72478000 C:\Windows\SysWOW64\WINNSI.DLL
ModLoad: 724f0000 724fa000 C:\Windows\SysWOW64\netutils.dll
ModLoad: 724d0000 724eb000 C:\Windows\SysWOW64\srvccli.dll
ModLoad: 724c0000 724d0000 C:\Windows\SysWOW64\wkscli.dll
ModLoad: 74d00000 74d1d000 C:\Windows\SysWOW64\spicli.dll
ModLoad: 74cf0000 74cf9000 C:\Windows\SysWOW64\CRYPTBASE.dll
ModLoad: 74e30000 74e33000 C:\Windows\SysWOW64\bcryptPrimitives.dll
(298.254) Break instruction exception - code: 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=07610000 edx=00000000 esi=7f1de000 edi=00000000
eip=7737da9e esp=0018fa00 ebp=0018fb18 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b
efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2b:
7737da89 cc int 3
0:000> g
ModLoad: 75920000 75945000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75ab0000 75ba7000 C:\Windows\SysWOW64\MSCTF.dll
(298.7f0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=7f1cfe1f ebx=00000003 ecx=071c2fe8 edx=00000000 esi=ffffffffff edi=0711e954
eip=701db650 esp=072cfdc8 ebp=072cfad0 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b
efl=00001026
ACE_v6!ACE_Process_Options::creation_flags
701db650 8b4104 mov eax,dword ptr [ecx+4] ds:002b:071c2fec=????????
```

It breaks when it's trying to put the value pointed by `[ecx+4]` into `eax` register.

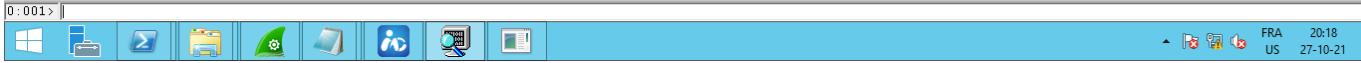
Looking at the heap:

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86

```

Executable search path is:
ModLoad: 00400000 004d7000 dbman.exe
ModLoad: 772d0000 77438000 ntdll.dll
Page heap: pid 0x96C page heap enabled with flags 0x3.
ModLoad: 72520000 7257d000 C:\Windows\SysWOW64\Verifier.dll
Page heap: pid 0x96C page heap enabled with flags 0x3.
ModLoad: 756e0000 75820000 C:\Windows\SysWOW64\KERNEL32.DLL
ModLoad: 74500000 7451f000 C:\Windows\SysWOW64\KERNELBASE.dll
ModLoad: 70140000 70249000 C:\Program Files\iMC\dbman\bin\ACE_v6.dll
ModLoad: 00230000 100932000 C:\Windows\SysWOW64\iMC_dbman\bin\smmp_v6.dll
ModLoad: 75950000 75a5c000 C:\Windows\SysWOW64\ole32.dll
ModLoad: 75650000 756d7000 C:\Windows\SysWOW64\OLEAUT32.dll
ModLoad: 703d0000 7045e000 C:\Windows\WinSxS\x86_microsoft.vc90.crt_1fc8b3b9ale18e3b_9.0.30729.8387_none_5094ca96bcb6b2bb\MSVCP90.dll
ModLoad: 61fa8000 61b23000 C:\Windows\WinSxS\x86_microsoft.vc90.crt_1fc8b3b9ale18e3b_9.0.30729.8387_none_5094ca96bcb6b2bb\MSVCR90.dll
ModLoad: 724a0000 724be000 C:\Windows\SysWOW64\IPHLPAPI.DLL
ModLoad: 754b0000 755ff000 C:\Windows\SysWOW64\USER32.dll
ModLoad: 74ad0000 74e58000 C:\Windows\SysWOW64\ADVAPI32.dll
ModLoad: 72820000 72865000 C:\Windows\SysWOW64\WS2_32.dll
ModLoad: 72510000 72545000 C:\Windows\SysWOW64\NETAPI32.dll
ModLoad: 75820000 758de000 C:\Windows\SysWOW64\RPCBASE.dll
ModLoad: 77140000 771f1000 C:\Windows\SysWOW64\RPCRT4.dll
ModLoad: 758e0000 7591e000 C:\Windows\SysWOW64\sechost.dll
ModLoad: 74def000 74ff8000 C:\Windows\SysWOW64\GDI32.dll
ModLoad: 75d50000 75d57000 C:\Windows\SysWOW64\NS.dll
ModLoad: 72470000 72478000 C:\Windows\SysWOW64\WINNSI.DLL
ModLoad: 724f0000 724fa000 C:\Windows\SysWOW64\util.dll
ModLoad: 724d0000 724eb000 C:\Windows\SysWOW64\ervcv1.dll
ModLoad: 724c0000 724d0000 C:\Windows\SysWOW64\wkscl1.dll
ModLoad: 74d00000 74d1d000 C:\Windows\SysWOW64\SpicCI.dll
ModLoad: 74cf0000 74cf9000 C:\Windows\SysWOW64\CRYPTBASE.dll
ModLoad: 75920000 75945000 C:\Windows\SysWOW64\NcryptPrimitives.dll
(96c:80) Break instruction received. Code: 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=00000000 esi=00000000 edi=00000000
esp=7737da8 esp=0018fec ebp=0018fb18 icpl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b ef1=00000246
ntdll!LdrpDoDebuggerBreak+0x2b:
7737da8 cc int 3
0:000> g
ModLoad: 75920000 75945000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75ab0000 75ba7000 C:\Windows\SysWOW64\MSCTF.dll
(96c:8ec): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected or handled.
eax=07222fe8 ebx=00000003 ecx=07222fec edx=00000000 esi=f0fffff edi=0717e954
esp=701db650 ebp=0732fid9 icpl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b ef1=000010206
ACE_v6!ACE_Process_Options::creation_flags:
701db650 8b4104 mov eax,dword ptr [ecx+4] ds:002b 07222fec=?????????
0:001> !heap -p -a ecx+4

```



Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86

```

*** For some commands to work properly, your symbol path ***  

*** must point to .pdb files that have full type information. ***  

***  

*** Certain .pdb files (such as the public OS symbols) do not ***  

*** contain the required information. Contact the group that ***  

*** provided you with these symbols if you need this command to ***  

*** work. ***  

***  

*** Type referenced: verifier!_DPH_BLOCK_INFORMATION ***  

***  

*****  

***** Either you specified an unqualified symbol, or your debugger ***  

*** doesn't have full symbol information. Unqualified symbol ***  

*** resolution is turned off by default. Please either specify a ***  

*** fully qualified symbol name or enable symbol resolution ***  

*** of unqualified symbols by typing ".synapt- 100". Note that ***  

*** enabling unqualified symbol resolution with network symbol ***  

*** server shares in the symbol path may cause the debugger to ***  

*** appear to hang for long periods of time when an incorrect ***  

*** symbol name is typed or the network symbol server is down. ***  

***  

*** For some commands to work properly, your symbol path ***  

*** must point to .pdb files that have full type information. ***  

***  

*** Certain .pdb files (such as the public OS symbols) do not ***  

*** contain the required information. Contact the group that ***  

*** provided you with these symbols if you need this command to ***  

*** work. ***  

***  

*** Type referenced: verifier!_DPH_BLOCK_INFORMATION ***  

***  

*****  

address 07222fec found in  

_DPH_HEAP_ROOT @ 6bd1000  

in freed-ed allocation ( _DPH_HEAP_BLOCK: VirtAddr 7222000 VirtSize 2000  

72528fc2 verifier!VerifierDisableFaultInjectionExcludeRange+0x00003232  

773b05ee nt!RtlDebugFreeHeap+0x00000032  

77368033 nt!RtlFreeHeap+0x0000580e9  

77368031 nt!RtlFreeHeap+0x0000580e9  

6fae3c1b MSVCR90!free+0x000000cd  

0045eb09 dbman+0x0005eb09  

0045ec65 dbman+0x0005ec65  

0045b414 dbman+0x0005b414  

0045bb55 dbman+0x0005bb55  

7024c3c9 ACE_v6!ACE_WFM0_Reactor::upcall+0x00000099

```





[ecx+4] register address belongs to the heap block that has been freed. By trying to "move"/dereference that value into eax, it triggers an *access violation*(AV).

AV code in windbg:

Access violation - code c0000005 with DEP enabled:

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86

```
ModLoad: 75410000 7545d000 C:\Windows\SysWOW64\WS2_32.dll
ModLoad: 70030000 700d3000 C:\Windows\Win32s\x86_microsoft_vc90.crt_1fc8b3b9a1e18e3b_9.0.30729.8387_none_5094ca96bc6b2bb\MSVCR90.dll
ModLoad: 74ed0000 74ee0000 C:\Windows\SysWOW64\IPHLPAPI.DLL
ModLoad: 75610000 7575f000 C:\Windows\SysWOW64\USER32.dll
ModLoad: 76470000 764e8000 C:\Windows\SysWOW64\ADVAPI32.dll
ModLoad: 74e80000 74ec5000 C:\Windows\SysWOW64\MSVSOCK.dll
ModLoad: 74f30000 74f42000 C:\Windows\SysWOW64\NETAPI32.dll
ModLoad: 76320000 7646e000 C:\Windows\SysWOW64\cibase.dll
ModLoad: 752b0000 752be000 C:\Windows\SysWOW64\msvcr7.dll
ModLoad: 75b10000 75c51000 C:\Windows\SysWOW64\CRYPTBASE.dll
ModLoad: 74f10000 752e0000 C:\Windows\SysWOW64\GDI32.dll
ModLoad: 75770000 75878000 C:\Windows\SysWOW64\RPCRT4.dll
ModLoad: 75760000 75767000 C:\Windows\SysWOW64\NSI.dll
ModLoad: 74e50000 74e58000 C:\Windows\SysWOW64\WINNSI.dll
ModLoad: 74f20000 74f2a000 C:\Windows\SysWOW64\netutils.dll
ModLoad: 74f00000 74f1b000 C:\Windows\SysWOW64\svclib.dll
ModLoad: 74ef0000 74f00000 C:\Windows\SysWOW64\wkscli.dll
ModLoad: 75b1b000 75c1d000 C:\Windows\SysWOW64\SspCli.dll
ModLoad: 751a0000 751a9000 C:\Windows\SysWOW64\CRYPTBASE.dll
(704.199) Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=f3040000 edi=00000000 esi=7ffd0000 edi=00000000
esp=7782d8e8 esp=0019faec ebp=0018fb19 iopl=0 nv up ei pl zr na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000246
ntdll!LdrInitShimEngineDynamic+0x75a:
7782d8e8 cc          int     3
0:000: bp 0045b40f
0:000: g
ModLoad: 759d0000 759f5000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75310000 75407000 C:\Windows\SysWOW64\MSCTF.dll
Breakpoint 0 hit
eax=00000000 ebx=00000001 ecx=01e5a6f8 edx=01db8f8c esi=01e5a6f8 edi=000001c8
esp=0045b40c esp=001dbcb08 ebp=01dbcb44 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000216
dbman+0x5b40f e80c380000 call dbman+0x5ec20 (0045ec20)
0:001: g
(704.bf0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=000001ff ebx=00000000 ecx=01e5a6f8 edx=feeefeee esi=01e5a6f8 edi=01e4d614
esp=feeefeee esp=01dbfdc0 ebp=01e4befc iopl=0 nv up ei ng nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00010286
feeefeee ?? ??
0:001: load msec
0:001: !exploitable

!exploitable 1.6.0.0
Exploitability Classification: EXPLOITABLE
Recommended Bug Title: Exploitable - Data Execution Prevention Violation starting at Unknown Symbol @ 0xfffffffffffffeffff called from ACE_v6!ACE_WFMO_Reactor_Handler_Report
```

Access violation - code **c0000005** without DEP:

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
ModLoad: 74a40000 74a5d000 C:\Windows\SysWOW64\SSpiCli.dll
ModLoad: 74a30000 74a39000 C:\Windows\SysWOW64\CRYPTBASE.dll
ModLoad: 749d0000 74a23000 C:\Windows\SysWOW64\bcryptPrimitives.dll
(900 :880): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=f0970000 edx=00000000 esi=f1fe000 edi=00000000
eip=770bda8e esp=0018faec ebp=0018fb18 iopl=0 nv up ei pl nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2b:
770bda8e cc          int     3
0:000> g
ModLoad: 768b0000 768d5000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75090000 75187000 C:\Windows\SysWOW64\MSCTF.dll
Breakpoint 0 hit
eax=00000000 ebx=00000001 ecx=01e6a7c8 edx=005d8f8c esi=01e6a7c8 edi=000001c4
eip=0045b40f esp=005debe8 ebp=005fdfd4 iopl=0 nv up ei pl nz ac pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000216
dbman+0xb40f:
0045b40f e80c380000 call    dbman+0xsec20 (0045ec20)
0:001> bp 0045ec63
0:000> g
Breakpoint 1 hit
eax=0045eb0 ebx=00000001 ecx=01e6a7c8 edx=004b79a4 esi=01e6a7c8 edi=000001c4
eip=0045eb3 esp=005debc0 ebp=005dfbe0 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
dbman+0xsec63:
0045ec63 ffd0        call    eax (dbman+0x5eb0 (0045eb0))
0:001> t
eax=0045eb0 ebx=00000001 ecx=01e6a7c8 edx=004b79a4 esi=01e6a7c8 edi=000001c4
eip=0045eb0 esp=005debc8 ebp=005dfbe0 iopl=0 nv up ei pl nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000202
dbman+0xebab0:
0045ebab0 55         push    ebp
0:000> push    ebp
(900 :ba0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=000001ff ebx=00000000 ecx=01e6a7c8 edx=f0fffffeeee esi=01e6a7c8 edi=0035d6d4
eip=f0fffffeeee esp=005dfdc0 ebp=0035bfbc iopl=0 nv up ei ng nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00010286
f0fffffeeee ???
0:001> !load msec
0:001> !exploitable
!exploitable 1.6.0.0
Exploitability Classification: EXPLOITABLE
Recommended Bug Title: Exploitable - Read Access Violation at the Instruction Pointer starting at Unknown Symbol @ 0xfffffffffffffeeee called from ACE_v6!ACE_WFMO_Reactor
Access violations at the instruction pointer are exploitable if not near NULL.

0:001> 

```

Looking at register/heap just after AV:

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
7737da8e cc          int     3
0:000> g
ModLoad: 75920000 75945000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75ab0000 75ba7000 C:\Windows\SysWOW64\MSCTF.dll
(6a4 :60): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=000001ff ebx=00000000 ecx=01e6a6f8 edx=f0fffffeeee esi=01e6a6f8 edi=001fd614
eip=f0fffffeeee esp=001fd000 ebp=001fbefc iopl=0 nv up ei ng nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00010286
f0fffffeeee ???
0:001> kF
# !killDEBP RetAddr
WARNING: Frame IP not in any known module. Following frames may be wrong.
00 01e5fdb0 7005b2f2 _HEAP@f0fffffeeee
01 01e5fdec 7001fe8 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x17f
02 01e5fdff 7005d419 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes+0x8
03 01e5f020 7001fid8 ACE_v6!ACE_WFMO_Reactor::update_state+0x119
04 00000000 00000000 ACE_v6!ACE_WFMO_Reactor::safe_dispatch+0x8
0:001> !heap -p -a ecx
address 01e6a6f8 found in
 _HEAP @ f10000
 HEAP_ENTRY Size Prev Flags      UserPtr UserSize - state
 01e6a6f0 0009 0000 [00] 01e6a6f8 00040 - (free)

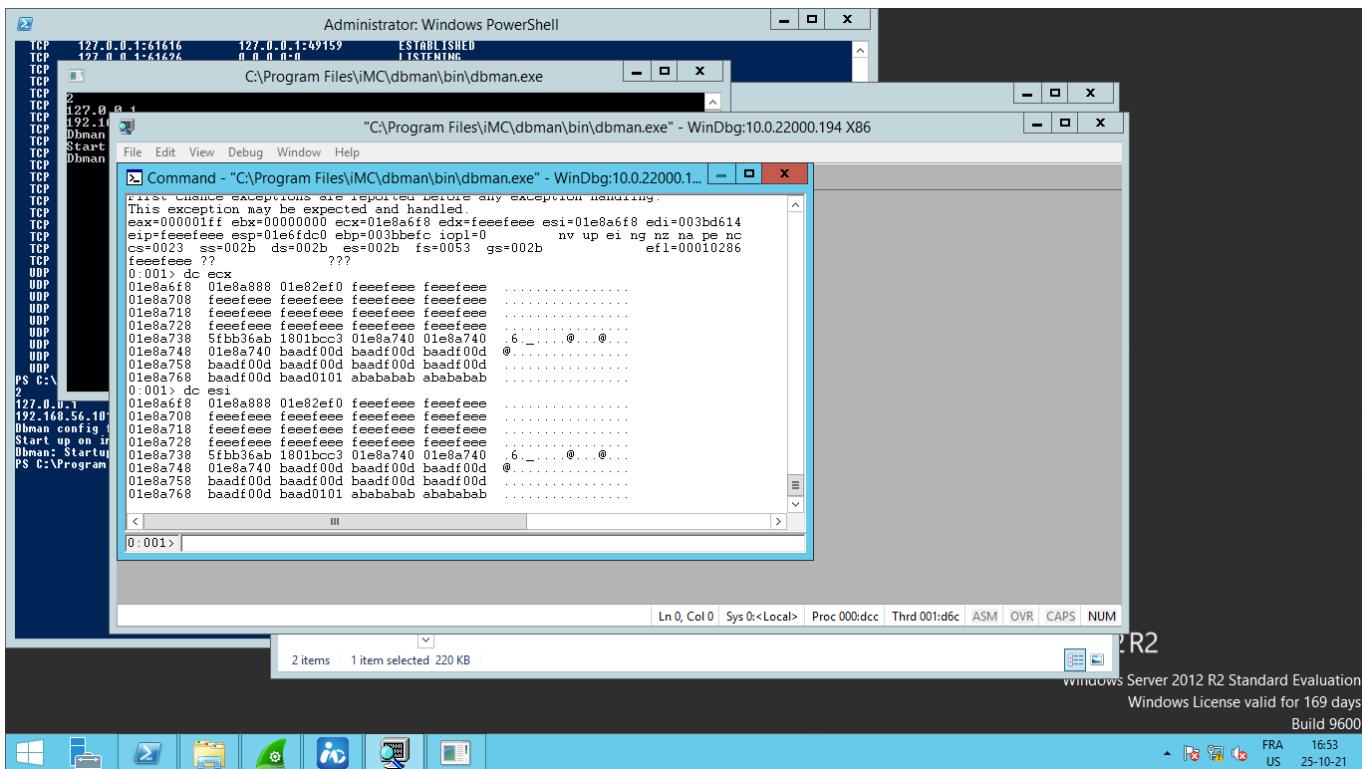
0:001> !heap -p -a edi
address 001fd614 found in
 _HEAP @ f10000
 HEAP_ENTRY Size Prev Flags      UserPtr UserSize - state
 001fd5b8 00e3 0000 [00] 001fd5c0 00700 - (busy)

0:001> !heap -p -a ebp
address 001fbefc found in
 _HEAP @ f10000
 HEAP_ENTRY Size Prev Flags      UserPtr UserSize - state
 001fbe48 0029 0000 [00] 001fbe50 00130 - (busy)
 ACE_v6!ACE_WFMO_Reactor::vitable

0:001> u eip
f0fffffeeee ???
^ Memory access error in 'u eip'

0:001> 

```



From: [https://en.wikipedia.org/wiki/Magic_number_\(programming\)](https://en.wikipedia.org/wiki/Magic_number_(programming))

BAADF00D:

| | |
|----------|---|
| BAADF00D | "Bad food", Used by Microsoft's debug HeapAlloc() to mark uninitialized allocated heap memory |
|----------|---|

FEEEFEEE:

| | |
|----------|--|
| FEEEFEEE | "Fee fee", Used by Microsoft's debug HeapFree() to mark freed heap memory. Some nearby internal bookkeeping values may have the high word set to FEEE as well. |
|----------|--|

These, confirm that there is some heap magic going on.

Refining analysis:

Loading **dbman.exe** from windbg:

```

ModLoad: 75520000 75529000  C:\Windows\SysWOW64\CRYPTBASE.dll
ModLoad: 754c0000 75513000  C:\Windows\SysWOW64\bcryptPrimitives.dll
(558.718): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=859b0000 edx=00000000 esi=7ffdde000 edi=00000000
eip=77bada8e esp=0018faec ebp=0018fb18 iopl=0          nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b          efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2b:
77bada8e cc          int     3

```

```

0:000> g
ModLoad: 76540000 76565000  C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 76760000 76857000  C:\Windows\SysWOW64\MSCTF.dll
(558.3d4): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.

eax=07294fe8 ebx=00000003 ecx=07294fe8 edx=00000000 esi=fffffff edi=071f0954
eip=6ff7b650 esp=0739fdc8 ebp=071dcf7c iopl=0          nv up ei pl nz na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b          efl=00010206
ACE_v6!ACE_Process_Options::creation_flags:
6ff7b650 8b4104          mov     eax,dword ptr [ecx+4] ds:002b:07294fec=?????????
```

ACE_v6!ACE_Process_Options::creation_flags

According to:

<http://www.dre.vanderbilt.edu/Doxygen/5.7.6/html/ace/a00507.html>

Set the creation flags to affect how a new process is spawned.

How has **ecx** been affected earlier, looking backwards:

```

0:001> ub eip . L 16
ACE_v6!ACE_INET_Addr::is_ipv4_compatible_ipv6+0x1d:
6ff7b62d cc          int     3
6ff7b62e cc          int     3
6ff7b62f cc          int     3
ACE_v6!ACE_Asynch_Result::ACE_Asynch_Result:
6ff7b630 8bc1        mov     eax,ecx
6ff7b632 8b4c2404    mov     ecx,dword ptr [esp+4]
6ff7b636 c70044e4ff6f  mov     dword ptr [eax],offset
ACE_v6!ACE_Asynch_Result::`vftable' (6ffe444)
6ff7b63c 894804    mov     dword ptr [eax+4],ecx
6ff7b63f c20400    ret     4
6ff7b642 cc          int     3
6ff7b643 cc          int     3
6ff7b644 cc          int     3
6ff7b645 cc          int     3
6ff7b646 cc          int     3
6ff7b647 cc          int     3
6ff7b648 cc          int     3
6ff7b649 cc          int     3
```

```
6ff7b64a cc          int     3
6ff7b64b cc          int     3
6ff7b64c cc          int     3
6ff7b64d cc          int     3
6ff7b64e cc          int     3
6ff7b64f cc          int     3
```

it takes the value at `[esp+4]`

Looking at the call stack and first three parameters:

```
0:001> kb
# ChildEBP RetAddr      Args to Child
WARNING: Stack unwind information not available. Following frames may be wrong.
00 0739fdc4 6ffeb813      071dcf7c 071dcf7c 071dcf50
ACE_v6!ACE_Process_Options::creation_flags
01 0739fdec 6ffaef8       071dcf7c 071dcf7c 071dcf50
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x163
02 0739fdf4 6ffed419       53e7b4d5 00000000 071dcf7c
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes+0x8
03 0739fe20 6ffaf1d8       6ffaf1b2 53e7b489 00000000
ACE_v6!ACE_WFMO_Reactor::update_state+0x119
04 00000000 00000000       00000000 00000000 00000000
ACE_v6!ACE_WFMO_Reactor::safe_dispatch+0x88
```

For line 00, we see two times `071dcf7c` passed as argument to `Process Options...`

Looking at the heap:

```
0:001> !heap -p -a 071dcf7c
address 071dcf7c found in
_DPH_HEAP_ROOT @ 6b41000
in busy allocation ( DPH_HEAP_BLOCK:             UserAddr           UserSize -
VirtAddr           VirtSize)
                           73a0e38:           71dcf7c           130 -
71dc000           2000
                  ACE_v6!ACE_WFMO_Reactor::`vftable'
70088d9c verifier!AVrfDebugPageHeapAllocate+0x0000023c
77bdfb79 ntdll!RtlDebugAllocateHeap+0x00000032
77b79a73 ntdll!RtlpAllocateHeap+0x0003914a
77b40b43 ntdll!RtlAllocateHeap+0x0000014c
```

```

6fe93db8 MSVCR90!malloc+0x00000079
[f:\dd\vctools\crt_bld\self_x86\crt\src\malloc.c @ 163]
    6fe93eb8 MSVCR90!operator new+0x0000001f
[f:\dd\vctools\crt_bld\self_x86\crt\src\new.cpp @ 59]
    6fff4853 ACE_v6!std::_Init_locks::operator=+0x0000029d
    6ffcb0b7 ACE_v6!ACE_Reactor::ACE_Reactor+0x00000057
    6ffc1004 ACE_v6!ACE_OS_Thread_Adapter::invoke+0x00000074
    6fe53433 MSVCR90!_callthreadstartex+0x0000001b
[f:\dd\vctools\crt_bld\self_x86\crt\src\threadex.c @ 348]
    6fe534c7 MSVCR90!_threadstartex+0x00000069
[f:\dd\vctools\crt_bld\self_x86\crt\src\threadex.c @ 326]
    7575919f KERNEL32!BaseThreadInitThunk+0x0000000e
    77b4a8cb ntdll!__RtlUserThreadStart+0x00000020
    77b4a8a1 ntdll!_RtlUserThreadStart+0x0000001b

```

Looking at the return address of line 00:

```

0:001> ub 6ffeb813
ACE_v6!ACE_WFM0_Reactor_Handler_Repository::make_changes_in_current_infos+0x149:
6ffeb7f9 893481      mov     dword ptr [ecx+eax*4],esi
6ffeb7fc 017514      add     dword ptr [ebp+14h],esi
6ffeb7ff 8b4c2414      mov     ecx,dword ptr [esp+14h]
6ffeb803 3bca      cmp     ecx,edx
6ffeb805 7439      je
ACE_v6!ACE_WFM0_Reactor_Handler_Repository::make_changes_in_current_infos+0x190
(6ffeb840)
6ffeb807 e8b4dafaff      call
ACE_v6!ACE_Event_Handler::reference_counting_policy (6ff992c0)
6ffeb80c 8bc8      mov     ecx,eax
6ffeb80e e83dfef8ff      call     ACE_v6!ACE_Process_Options::creation_flags
(6ff7b650)

```

Still looking backwards, restart by setting `bp 6ffeb7fc`:

```

(ba4.470): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=2ee70000 edx=00000000 esi=7ffdde000 edi=00000000
eip=77bada8e esp=0018faec ebp=0018fb18 iopl=0          nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b          efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2b:
77bada8e cc          int     3

```

```

0:000> bp 6ffeb7fc
0:000> g
ModLoad: 76540000 76565000 C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 76760000 76857000 C:\Windows\SysWOW64\MSCTF.dll
Breakpoint 0 hit

eax=00000003 ebx=00000003 ecx=0710ef00 edx=00000000 esi=ffffffff edi=07110954
eip=6ffeb7fc esp=072bfdcc ebp=070fcf7c iopl=0 nv up ei pl zr na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000246
ACE_v6!ACE_WFM0_Reactor_Handler_Repository::make_changes_in_current_infos+0x14c:
6ffeb7fc 017514 add dword ptr [ebp+14h],esi
ss:002b:070fcf90=00000004

0:001> u
ACE_v6!ACE_WFM0_Reactor_Handler_Repository::make_changes_in_current_infos+0x14c:
6ffeb7fc 017514 add dword ptr [ebp+14h],esi
6ffeb7ff 8b4c2414 mov ecx,dword ptr [esp+14h]
6ffeb803 3bca cmp ecx,edx
6ffeb805 7439 je
ACE_v6!ACE_WFM0_Reactor_Handler_Repository::make_changes_in_current_infos+0x190
(6ffeb840)
6ffeb807 e8b4dafaff call
ACE_v6!ACE_Event_Handler::reference_counting_policy (6ff992c0)
6ffeb80c 8bc8 mov ecx, eax
6ffeb80e e83dfef8ff call ACE_v6!ACE_Process_Options::creation_flags
(6ff7b650)
6ffeb813 8b742414 mov esi,dword ptr [esp+14h]

0:001> dc esi
ffffffff ????????? ?????????? ?????????? ?????????? ??????????????????
0000000f ?????????? ?????????? ?????????? ?????????? ??????????????????
0000001f ?????????? ?????????? ?????????? ?????????? ??????????????????
0000002f ?????????? ?????????? ?????????? ?????????? ??????????????????
0000003f ?????????? ?????????? ?????????? ?????????? ??????????????????
0000004f ?????????? ?????????? ?????????? ?????????? ??????????????????
0000005f ?????????? ?????????? ?????????? ?????????? ??????????????????
0000006f ?????????? ?????????? ?????????? ?????????? ??????????????????

```

Keep going backwards:

From **kb** output:

```
02 0739fdf4 6ffed419      53e7b4d5 00000000 071dced0
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes+0x8
```

Restart using `bp 6ffed419`:

```
(86c.224): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=fadaf0000 edx=00000000 esi=7ffdde000 edi=00000000
eip=77bada8e esp=0018faec ebp=0018fb18 iopl=0          nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b          efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2b:
77bada8e cc          int     3
0:000> bp 6ffed419
0:000> g
ModLoad: 76540000 76565000  C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 76760000 76857000  C:\Windows\SysWOW64\MSCTF.dll
Breakpoint 0 hit
eax=00000000 ebx=0732cf50 ecx=ffffffff edx=c0c00001 esi=0732ced0 edi=0732cf7c
eip=6ffed419 esp=074efdfc ebp=00000000 iopl=0          nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b          efl=00000246
ACE_v6!ACE_WFMO_Reactor::update_state+0x119:
6ffed419 8b07          mov     eax,dword ptr [edi]  ds:002b:0732cf7c=
{ACE_v6!ACE_WFMO_Reactor_Handler_Repository::`vtable' (70001ba8)}

0:001> u
ACE_v6!ACE_WFMO_Reactor::update_state+0x119:
6ffed419 8b07          mov     eax,dword ptr [edi]
6ffed41b 8b5004          mov     edx,dword ptr [eax+4]
6ffed41e 8bcf          mov     ecx,edi
6ffed420 ffd2          call    edx
6ffed422 84c0          test   al,al
6ffed424 75ea          jne    ACE_v6!ACE_WFMO_Reactor::update_state+0x110
(6ffed410)
6ffed426 8b06          mov     eax,dword ptr [esi]
6ffed428 8b9050010000  mov     edx,dword ptr [eax+150h]

0:001> t
eax=70001ba8 ebx=0732cf50 ecx=ffffffff edx=c0c00001 esi=0732ced0 edi=0732cf7c
```

```
eip=6ffed41b esp=074efdfc ebp=00000000 iopl=0          nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b          efl=00000246
ACE_v6!ACE_WFMO_Reactor::update_state+0x11b:
6ffed41b 8b5004      mov     edx,dword ptr [eax+4] ds:002b:70001bac=
{ACE_v6!ACE_WFMO_Reactor_Handler_Repository::changes_required (6ffaefb0)}
```

Looking into **eax**:

```
0:001> dps 70001ba8
70001ba8  6ffaee20
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::ACE_WFMO_Reactor_Handler_Repository+
0x60
70001bac  6ffaefb0 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::changes_required
70001bb0  6ffaefe0 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes
70001bb4  00000000
70001bb8  70009b88 ACE_v6!ACE_WIN32_Proactor::`vftable'+0x3364
70001bbc  6ffaf690 ACE_v6!ACE_Msg_WFMO_Reactor::`default constructor
closure'+0x20
70001bc0  6ffedc90 ACE_v6!ACE_WFMO_Reactor::open
70001bc4  6ffebbbb0 ACE_v6!ACE_Event_Handler::handle_timeout
70001bc8  6ffebbbc0 ACE_v6!ACE_WFMO_Reactor::set_sig_handler
70001bcc  6ffdd420 ACE_v6!ACE_Dynamic_Message_Strategy::dynamic_priority_max
70001bd0  6ffebbf0 ACE_v6!ACE_WFMO_Reactor::timer_queue
70001bd4  6ffed880 ACE_v6!ACE_WFMO_Reactor::close
70001bd8  6ffebbb90 ACE_v6!ACE_Recursive_Thread_Mutex::acquire
70001bdc  6ffaf3b0 ACE_v6!ACE_WFMO_Reactor::handle_events
70001be0  6ffaf3b0 ACE_v6!ACE_WFMO_Reactor::handle_events
70001be4  6ffaf3d0 ACE_v6!ACE_Msg_WFMO_Reactor::alertable_handle_events
70001be8  6ffaf3d0 ACE_v6!ACE_Msg_WFMO_Reactor::alertable_handle_events
70001bec  6ff71b90 ACE_v6!ACE_WFMO_Reactor::deactivated
70001bf0  6ffaf0f0 ACE_v6!ACE_WFMO_Reactor::deactivate
70001bf4  6ffaf210 ACE_v6!ACE_WFMO_Reactor::register_handler
70001bf8  6ffaf200 ACE_v6!ACE_WFMO_Reactor::register_handler
70001bfc  6ffafa90 ACE_v6!ACE_WFMO_Reactor::register_handler
70001c00  6ffaf9e0 ACE_v6!ACE_WFMO_Reactor::register_handler
70001c04  6ffaf7d0 ACE_v6!ACE_WFMO_Reactor::register_handler
70001c08  6ffaf930 ACE_v6!ACE_WFMO_Reactor::register_handler
70001c0c  6ffaf880 ACE_v6!ACE_WFMO_Reactor::register_handler
70001c10  6ffaf2a0 ACE_v6!ACE_WFMO_Reactor::remove_handler
70001c14  6ff7e390 ACE_v6!ACE_WFMO_Reactor::remove_handler
70001c18  6ffafe70 ACE_v6!ACE_WFMO_Reactor::remove_handler
```

```
70001c1c  6ffafe60 ACE_v6!ACE_WFMO_Reactor::remove_handler
70001c20  6ffafe30 ACE_v6!ACE_WFMO_Reactor::remove_handler
70001c24  6ffb0020 ACE_v6!ACE_WFMO_Reactor::suspend_handler
```

We see seven call to: `ACE_v6!ACE_WFMO_Reactor::register_handler`
and five to: `ACE_v6!ACE_WFMO_Reactor::remove_handler`

```
0:001> dps esi
0732ced0  7000603c ACE_v6!ACE_WFMO_Reactor::`vtable'
0732ced4  0733cff8
0732ced8  c0c0c001
0732cedc  0732ef78
0732cee0  c0c00101
0732cee4  07346f30
0732cee8  c0c0c001
0732ceec  32333730
0732cef0  43454543
0732cef4  36353132
0732cef8  c0c0c000
0732cefcc  c0c0c0c0
0732cf00  c0c0c0c0
0732cf04  c0c0c0c0
0732cf08  c0c0c0c0
0732cf0c  c0c0c0c0
0732cf10  c0c0c0c0
0732cf14  c0c0c0c0
0732cf18  c0c0c0c0
0732cf1c  c0c0c0c0
0732cf20  c0c0c0c0
0732cf24  c0c0c0c0
0732cf28  c0c0c0c0
0732cf2c  c0c0c0c0
0732cf30  c0c0c0c0
0732cf34  c0c0c0c0
0732cf38  c0c0c0c0
0732cf3c  c0c0c0c0
0732cf40  c0c0c0c0
0732cf44  c0c0c0c0
0732cf48  c0c0c0c0
0732cf4c  c0c0c0c0
```

```
0:001> dps edi
0732cf7c  70001ba8 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::`vftable'
0732cf80  0732ced0
0732cf84  00000040
0732cf88  0733ef00
0732cf8c  07340900
0732cf90  00000003
0732cf94  07344800
0732cf98  00000000
0732cf9c  07342800
0732cfa0  00000000
0732cfa4  00000000
0732cfa8  00000000
0732cfac  00000000
0732cfb0  00000134
0732cfb4  c0c0c000
0732cfb8  00000138
0732cfbc  c0c0c000
0732fcf0  70001b54 ACE_v6!ACE_Wakeup_All_Threads_Handler::`vftable'
0732fcf4  00000001
0732fcf8  00000000
0732fcfc  00000000
0732cfcd  6ffffe1ec
ACE_v6!ACE_Event_Handler::Reference_Counting_Policy::`vftable'
0732cfcd4  00000001
0732cfcd8  0000013c
0732cfdc  c0c0c000
0732cfec  00000000
0732cfec4  000009b4
0732cfec8  00000000
0732cfecf  000009b4
0732cff0  00000130
0732cff4  00000134
0732cff8  c0c0c001
```

0xc0c0c0c0 is a marker for uninitialized data.

```
0:001> !heap -p -a ecx
address 0732cf7c found in
_DPH_HEAP_ROOT @ 6c51000
```

| VirtAddr | VirtSize | UserAddr | UserSize |
|---|----------|--|----------|
| 732c000 | 2000 | 74f0e38: | 732ced0 |
| | | ACE_v6!ACE_WFMO_Reactor::`vftable' | 130 - |
| | | 70088d9c verifier!AVrfDebugPageHeapAllocate+0x00000023c | |
| | | 77bdfb79 ntdll!RtlDebugAllocateHeap+0x00000032 | |
| | | 77b79a73 ntdll!RtlpAllocateHeap+0x0003914a | |
| | | 77b40b43 ntdll!RtlAllocateHeap+0x0000014c | |
| | | 6fe93db8 MSVCR90!malloc+0x00000079 | |
| [f:\dd\vctools\crt_bld\self_x86\crt\src\malloc.c @ 163] | | | |
| | | 6fe93eb8 MSVCR90!operator new+0x0000001f | |
| [f:\dd\vctools\crt_bld\self_x86\crt\src\new.cpp @ 59] | | | |
| | | 6fff4853 ACE_v6!std::_Init_locks::operator=+0x0000029d | |
| | | 6ffcb0b7 ACE_v6!ACE_Reactor::ACE_Reactor+0x00000057 | |
| | | 6ffc1004 ACE_v6!ACE_OS_Thread_Adapter::invoke+0x00000074 | |
| | | 6fe53433 MSVCR90!_callthreadstartex+0x0000001b | |
| [f:\dd\vctools\crt_bld\self_x86\crt\src\threadex.c @ 348] | | | |
| | | 6fe534c7 MSVCR90!_threadstartex+0x00000069 | |
| [f:\dd\vctools\crt_bld\self_x86\crt\src\threadex.c @ 326] | | | |
| | | 7575919f KERNEL32!BaseThreadInitThunk+0x0000000e | |
| | | 77b4a8cb ntdll!__RtlUserThreadStart+0x00000020 | |
| | | 77b4a8a1 ntdll!_RtlUserThreadStart+0x0000001b | |

It's difficult to keep track of the "callings" by rewinding like this.

Exploring through other debugging features of windbg:

Getting kernel32!GetLastError after AV

```
0:001> !gle
LastErrorValue: (WinSock) 0x2736 (10038) - An operation was attempted on
something that is not a socket.
LastStatusValue: (NTSTATUS) 0xc0000008 - An invalid HANDLE was specified
```

A bit more detail on the error code:

| [Select language] | | Home | Contact | www.beckhoff.com | email this page | Search | | | | | | | | | | | | | | | | | | | |
|--|---|--|-----------------------------|---|-----------------------------|---|---------------------------------|--|-----------------------------|---|-------------------------------|---|--------------------------------|---|------------------------------------|---|------------------------------------|--|-------------------------------|---|--|--|--|--|--|
| BECKHOFF New Automation Technology | | | | | | | | | | | | | | | | | | | | | | | | | |
| [Select language] | | | | | | | | | | | | | | | | | | | | | | | | | |
| <ul style="list-style-type: none"> Category System TS1xxx Category Measurement TS3xxx Category Control TS4xxx Category Motion TS5xxx Category Communication TS6xx <ul style="list-style-type: none"> TS6000 TwiCAT AOS Core TS6100 TwiCAT OPC UA TS6100-0030 TwinCAT OPI TS6120 TwiCAT OPC Server TS6220-TS6222 TwinCAT I/O TS6250 TwinCAT Modbus TCP TS6250-0030 TwinCAT Moxa TS6255 TwinCAT PLC Modbus TS6255-0030 TwinCAT PLC TS6270-0030 PROFINET I/O TS6271-0030 TwinCAT PROF TS6280 TwinCAT Ethernet TS6280-0030 TwinCAT EthIP TS6300 TwinCAT FTP Client TS6310 TwinCAT TCP/IP Services TS6340 TwinCAT PLC Library TS6341 TwinCAT PLC Library TS6350 TwinCAT SMS/SM TS6350-0030 TwinCAT SMS <ul style="list-style-type: none"> Forward Overview TwinCAT SMTP Server <ul style="list-style-type: none"> Overview Configuration Function blocks Samples Annex <ul style="list-style-type: none"> SMTP Error Codes Windows Sockets TS6360 TwinCAT Virtual Serial Port TS6370 TwinCAT Drive Top TS6420 TwinCAT DataBase TS6420-0030 TwinCAT Database TS6421 TwinCAT XML Data TS6421-0030 TwinCAT XML TwinCAT PLC: ICC 60870-5-104 TS6600 TwinCAT PLC Library TS6610 TwinCAT PLC Library Category Building Automation TS TwinCAT Diagnostics Application Notes Search | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>This error describes item operations on nonblocking sockets that complete immediately, for example recv when no data is queued to be read from the socket. It is a nonfatal error, and the operation should be retried later. It is normal for WSAEWOULDBLOCK to be reported as the result from calling connect on a nonblocking SOCK_STREAM socket, since some time must elapse for the connection to be established.</p> <table border="1"> <tr> <td>WSAEINPROGRESS 10036</td><td><i>Operation now in progress.</i> A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per-task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error.</td></tr> <tr> <td>WSAEALREADY 10037</td><td><i>Operation already in progress.</i> An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling connect a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (WSAAAsyncGetXbyY) that has already been canceled or completed.</td></tr> <tr> <td>WSAENOTSOCK 10038</td><td><i>Socket operation on nonsocket.</i> An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for select, a member of an fd_set was not valid.</td></tr> <tr> <td>WSAEDESTADDRREQ 10039</td><td><i>Destination address required.</i> A required address was omitted from an operation on a socket. For example, this error is returned if sendto is called with the remote address of ADDR_ANY.</td></tr> <tr> <td>WSAEMSGSIZE 10040</td><td><i>Message too long.</i> A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself.</td></tr> <tr> <td>WSAEPROTOTYPE 10041</td><td><i>Protocol wrong type for socket.</i> A protocol was specified in the socket function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM.</td></tr> <tr> <td>WSAENOPROTOOPT 10042</td><td><i>Bad protocol option.</i> An unknown, invalid or unsupported option or level was specified in a getsockopt or setsockopt call.</td></tr> <tr> <td>WSAEPROTONOSUPPORT 10043</td><td><i>Protocol not supported.</i> The requested protocol has not been configured into the system, or no implementation for it exists. For example, a socket call requests a SOCK_DGRAM socket, but specifies a stream protocol.</td></tr> <tr> <td>WSAFSOCKTNOSUPPORT 10044</td><td><i>Socket type not supported.</i> The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a socket call, and the implementation does not support SOCK_RAW sockets at all.</td></tr> <tr> <td>WSAEOPNOTSUPP 10045</td><td><i>Operation not supported.</i> The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation.</td></tr> </table> | WSAEINPROGRESS 10036 | <i>Operation now in progress.</i> A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per-task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error. | WSAEALREADY 10037 | <i>Operation already in progress.</i> An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling connect a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (WSAAAsyncGetXbyY) that has already been canceled or completed. | WSAENOTSOCK 10038 | <i>Socket operation on nonsocket.</i> An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for select, a member of an fd_set was not valid. | WSAEDESTADDRREQ 10039 | <i>Destination address required.</i> A required address was omitted from an operation on a socket. For example, this error is returned if sendto is called with the remote address of ADDR_ANY. | WSAEMSGSIZE 10040 | <i>Message too long.</i> A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself. | WSAEPROTOTYPE 10041 | <i>Protocol wrong type for socket.</i> A protocol was specified in the socket function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM. | WSAENOPROTOOPT 10042 | <i>Bad protocol option.</i> An unknown, invalid or unsupported option or level was specified in a getsockopt or setsockopt call. | WSAEPROTONOSUPPORT 10043 | <i>Protocol not supported.</i> The requested protocol has not been configured into the system, or no implementation for it exists. For example, a socket call requests a SOCK_DGRAM socket, but specifies a stream protocol. | WSAFSOCKTNOSUPPORT 10044 | <i>Socket type not supported.</i> The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a socket call, and the implementation does not support SOCK_RAW sockets at all. | WSAEOPNOTSUPP 10045 | <i>Operation not supported.</i> The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation. | | | | | |
| WSAEINPROGRESS 10036 | <i>Operation now in progress.</i> A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per-task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAEALREADY 10037 | <i>Operation already in progress.</i> An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling connect a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (WSAAAsyncGetXbyY) that has already been canceled or completed. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAENOTSOCK 10038 | <i>Socket operation on nonsocket.</i> An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for select, a member of an fd_set was not valid. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAEDESTADDRREQ 10039 | <i>Destination address required.</i> A required address was omitted from an operation on a socket. For example, this error is returned if sendto is called with the remote address of ADDR_ANY. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAEMSGSIZE 10040 | <i>Message too long.</i> A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAEPROTOTYPE 10041 | <i>Protocol wrong type for socket.</i> A protocol was specified in the socket function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAENOPROTOOPT 10042 | <i>Bad protocol option.</i> An unknown, invalid or unsupported option or level was specified in a getsockopt or setsockopt call. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAEPROTONOSUPPORT 10043 | <i>Protocol not supported.</i> The requested protocol has not been configured into the system, or no implementation for it exists. For example, a socket call requests a SOCK_DGRAM socket, but specifies a stream protocol. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAFSOCKTNOSUPPORT 10044 | <i>Socket type not supported.</i> The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a socket call, and the implementation does not support SOCK_RAW sockets at all. | | | | | | | | | | | | | | | | | | | | | | | | |
| WSAEOPNOTSUPP 10045 | <i>Operation not supported.</i> The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation. | | | | | | | | | | | | | | | | | | | | | | | | |

<https://docs.microsoft.com/en-us/windows/win32/api/errhandlingapi/nf-errhandlingapi-getlasterror>

Follow up this exception/handle issue in windbg:

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
0:001> !analyze -v
***** Exception Analysis *****
Current verifier stop:
APPLICATION_VERIFIER_NETWORKING_WSP_SOCKET_ALREADY_CLOSED (e104)
Attempt to use a closed SOCKET
A SOCKET from a Winsock base provider was used after it had been closed. This generally indicates a fault in a layered service provider (an LSP - a DLL between the app
To identify the routine that tried to use the closed SOCKET, dump the current stack trace by using the 'k' command in the debugger. To dump the stack trace of
Arguments:
Arg1: 00003dc, SOCKET being accessed.
Arg2: 05b15f5c, Stack trace of the function that closed the SOCKET. Use dps to dump the stack trace if not NULL
Arg3: 00000000, Not used
Arg4: 00000000, Not used

Previous verifier stop:
APPLICATION_VERIFIER_NETWORKING_WSP_SOCKET_ALREADY_CLOSED (e104)
Attempt to use a closed SOCKET
A SOCKET from a Winsock base provider was used after it had been closed. This generally indicates a fault in a layered service provider (an LSP - a DLL between the app
To identify the routine that tried to use the closed SOCKET, dump the current stack trace by using the 'k' command in the debugger. To dump the stack trace of
Arguments:
Arg1: 00003dc, SOCKET being accessed.
Arg2: 05b15f5c, Stack trace of the function that closed the SOCKET. Use dps to dump the stack trace if not NULL
Arg3: 00000000, Not used
Arg4: 00000000, Not used

*****
*** Either you specified an unqualified symbol, or your debugger
*** doesn't have full symbol information. Unqualified symbol
*** resolution is turned off by default. Please either specify a
*** fully qualified symbol module\symbolname, or enable resolution
*** of unqualified symbols by typing ".sympct- 100". Note that
*** enabling unqualified symbol resolution with network symbol
*** server shares in the symbol path may cause the debugger to
*** appear to hang for long periods of time when an incorrect
*** symbol name is typed or the network symbol server is down.
*** For some commands to work properly, your symbol path
*** must point to .pdb files that have full type information.
*** Certain .pdb files (such as the public OS symbols) do not
*** contain the required information. Contact the group that
*** provided you with these symbols if you need this command to
*** work.
*** Type referenced: kernelbase!qpServerNlsUserInfo
< 0:001>

```

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
NTGLOBALFLAG: 2000100
PROCESS_BAM_CURRENT_THROTTLED: 0
PROCESS_BAM_PREVIOUS_THROTTLED: 0
APPLICATION_VERIFIER_FLAGS: 80000001
APPLICATION_VERIFIER_LOADED: 1
EXCEPTION_RECORD: {(.exr -1)}
ExceptionAddress: cdcecafa
  ExceptionCode: c0000008 (Invalid handle)
  ExceptionFlags: 00000000
NumberParameters: 0
Thread tried to close a handle that was invalid or illegal to close
FAULTING_THREAD: 00000834
PROCESS_NAME: dbman.exe
ERROR_CODE: (NTSTATUS) 0xc0000008 - An invalid HANDLE was specified.
EXCEPTION_CODE_STR: c0000008
STACK_TEXT:
0x0cf24 77ab1ec nt!NtDeviceIoControlFile+0xc
0x0cf24 0fecc0d MSWSOCK!SockImportHandle+0xd0
0x0cf9c8 72fe47fb MSWSOCK!SockFindAndReferenceSocket+0x134d8
0x0cf4a0 61fd4edc vnet!VfHookWSPEGetSockOpt+0x5c
0x0cf4a8 75503c50 WS2_32!DCATALOG_FindIFSPProviderForSocket+0xb5
0x0cf4d4 754f5aa0 WS2_32!DSOCKET_FindIFSSocket+0x38
0x0cf4d40 754e13d1 WS2_32!WSAEnumNetworkEvents+0x2a
0x0cf4d58 61fd75d2 vnet!VfHookWSAEnumNetworkEvents+0x52
0x0cf4d7c 61fd6dd1f4 ACE_v6!ACE_WFMO_Reactor::complex_dispatch_handler+0x64
0x0cfdb8 61f772c06 vfcuzzi!CuzzScheduleInternal2+0x44
0x0cfde8 09fe0ed0 unknown!unknown+0x0
0x0cfef18 61fdcd184 ACE_v6!ACE_WFMO_Reactor::dispatch+0x34
0x0cfef28 61f69f1a3 ACE_v6!ACE_WFMO_Reactor::safe_dispatch+0x53
0x0cfef64 61f6dc01b ACE_v6!ACE_WFMO_Reactor::event_handling+0xbb

SYMBOL_NAME: MSWSOCK!SockImportHandle+d0
MODULE_NAME: MSWSOCK.dll
IMAGE_NAME: MSWSOCK.dll
STACK_COMMAND: .ecxr : kb ; ** Pseudo Context ** Pseudo ** Value: 9 ** ; kb
FAILURE_BUCKET_ID: INVALID_HANDLE_AVRF_c0000008_MSWSOCK.dll!SockImportHandle
< 0:001>

```

```

Command - "C:\Program Files\iMC\dbman\bin\dbman.exe" - WinDbg:10.0.22000.194 X86
09f0ef30 c0c0c0c0
09f0ef34 c0c0c0c0
09f0ef38 c0c0c0c0
09f0ef3c c0c0c0c0
09f0ef40 c0c0c0c0
09f0ef44 c0c0c0c0
09f0ef48 c0c0c0c0
09f0ef4c c0c0c0c0
0:001> dd 09f0ef40
09f0ef40 09f1eff8 00000001 09f10f78 <`00,.....x.09F0
09f0ef44 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef48 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef52 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef56 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef60 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef64 09f10f78 00000001 30463930 <`00,.....x.09F0
09f0ef68 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef72 09f10f78 00000001 30463930 <`00,.....x.09F0
09f0ef76 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef80 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef84 09f10f78 00000001 30463930 <`00,.....x.09F0
09f0ef88 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef92 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef96 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef9a c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef9e c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef40 09f10f78 00000001 30463930 <`00,.....x.09F0
09f0ef44 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef48 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef52 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef56 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef60 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef64 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef68 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef72 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef76 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef80 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef84 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef88 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef92 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef96 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef9a c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef9e 00363336 c0c0c0c0 c0c0c0c0 EEEC636.
09f0ef40 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef44 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef48 c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .
09f0ef4c c0c0c0c0 c0c0c0c0 c0c0c0c0 ..... .

```

Wrap up -- situation:

Exploitability:

As mentioned earlier, difficult it is to spot the vulnerability from a static analysis only perspective.

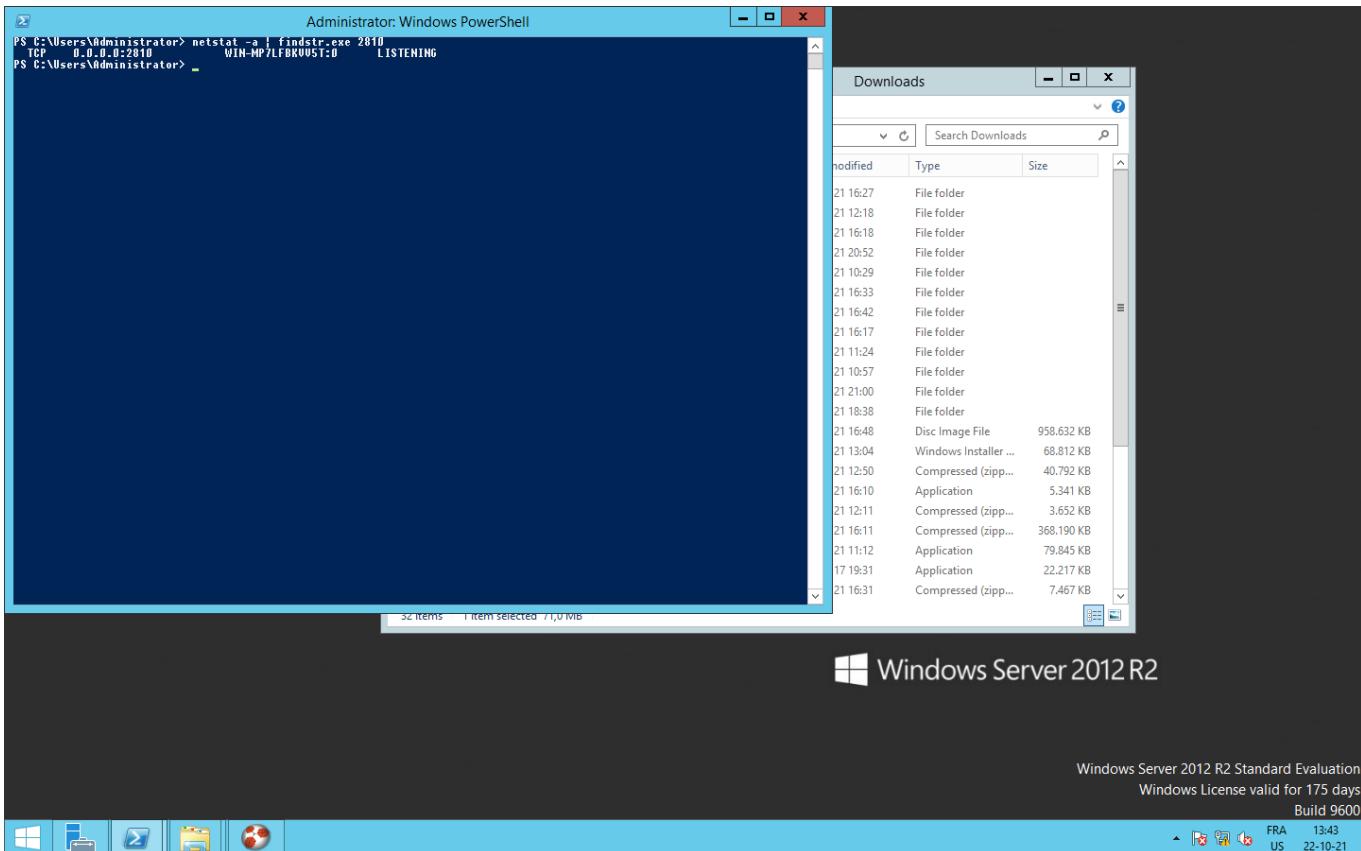
Stack-based buffer overflow

this section hasn't been thoroughly explored

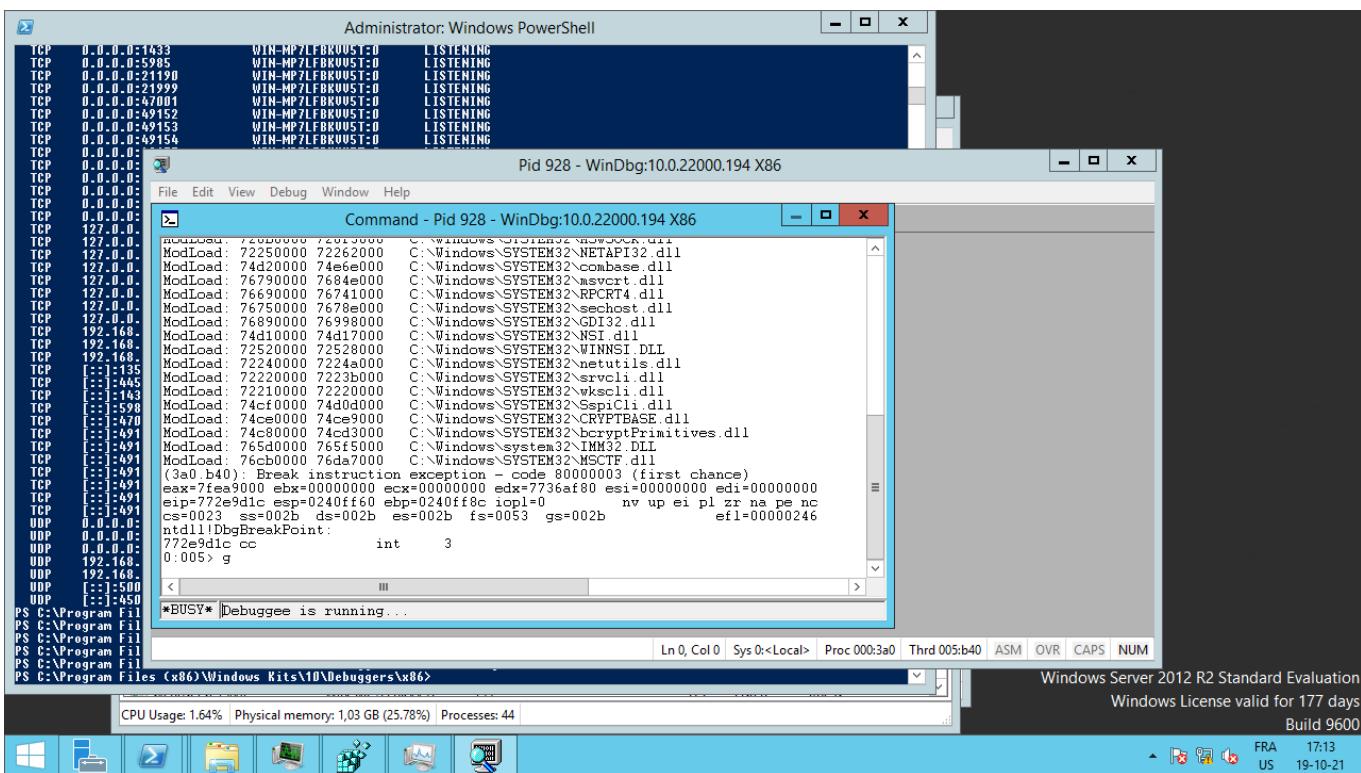
(iMC_PLAT_7.3_E0504_Ent_Win-003)

The following bug looks pretty much as a stack-based buffer overflow. At first look, it concerns the command opcode 10001.

To observe the behaviour of *dbman.exe* attaching a debugger can be handy. In a first place, I am going for *winDbg* and *msecdbg* extension. First, we assure that *dbman.exe* is running i.e. listening on port 2810:



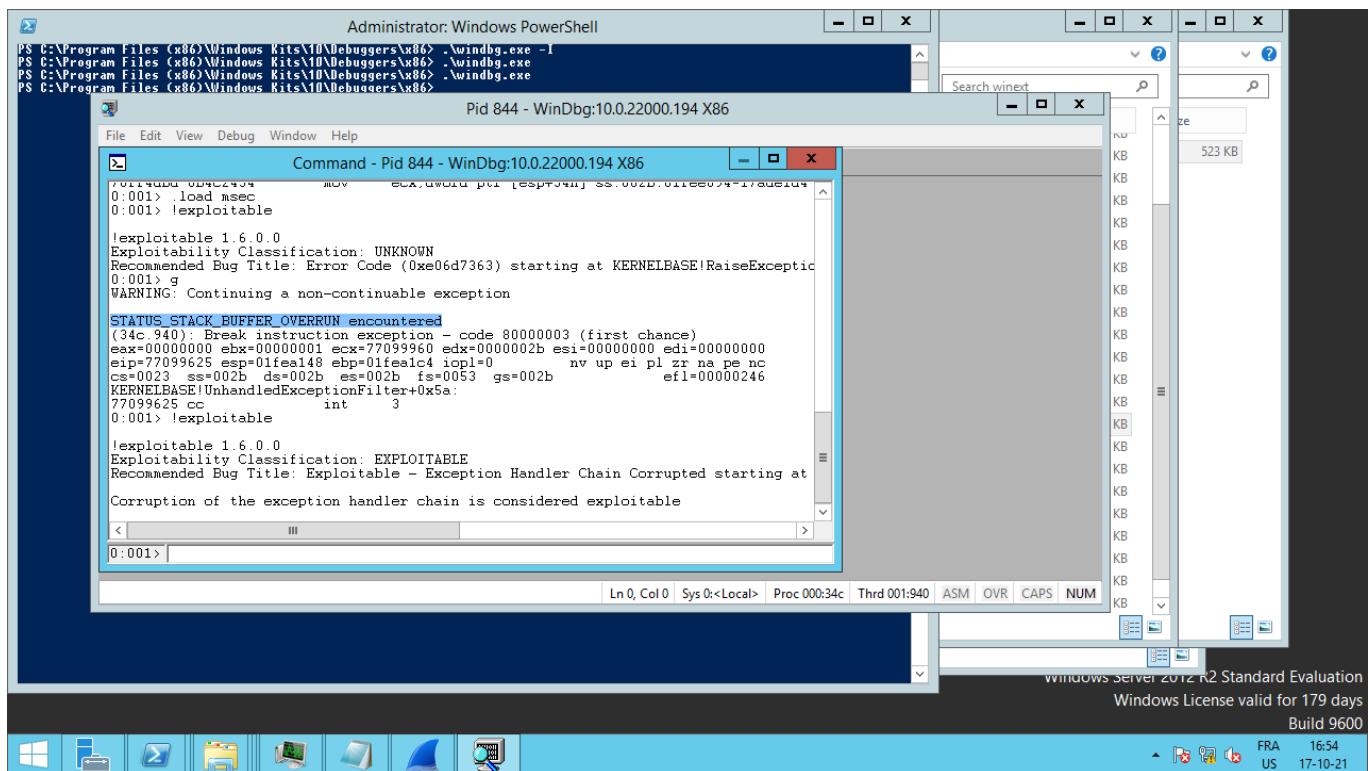
Attach WinDbg to dbman.exe:



Send Payload to port 2810:

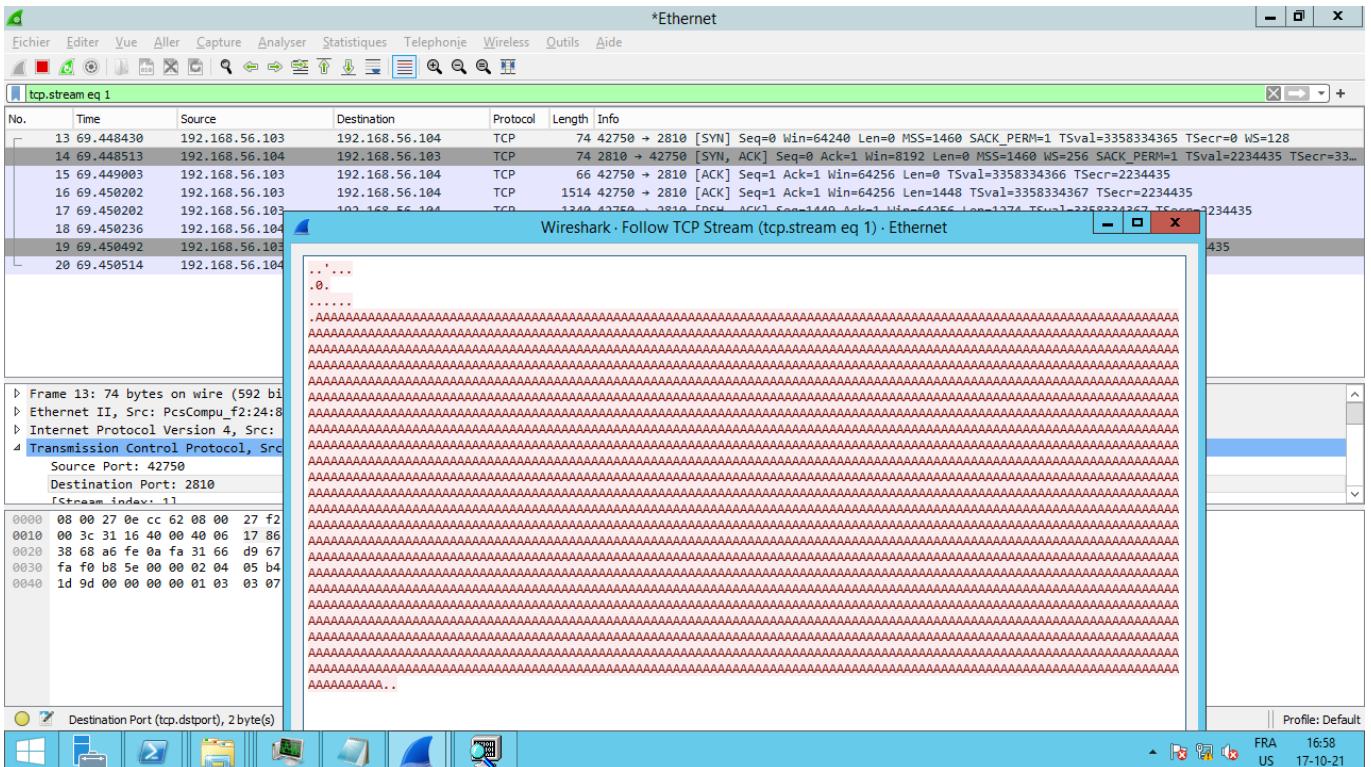
```
└─$ python3 fuzz.py
[+] Opening connection to 192.168.56.104 on port 2810: Done
[*] Closed connection to 192.168.56.104 port 2810
```

Observe behaviour, load msec extension and check exploitability:

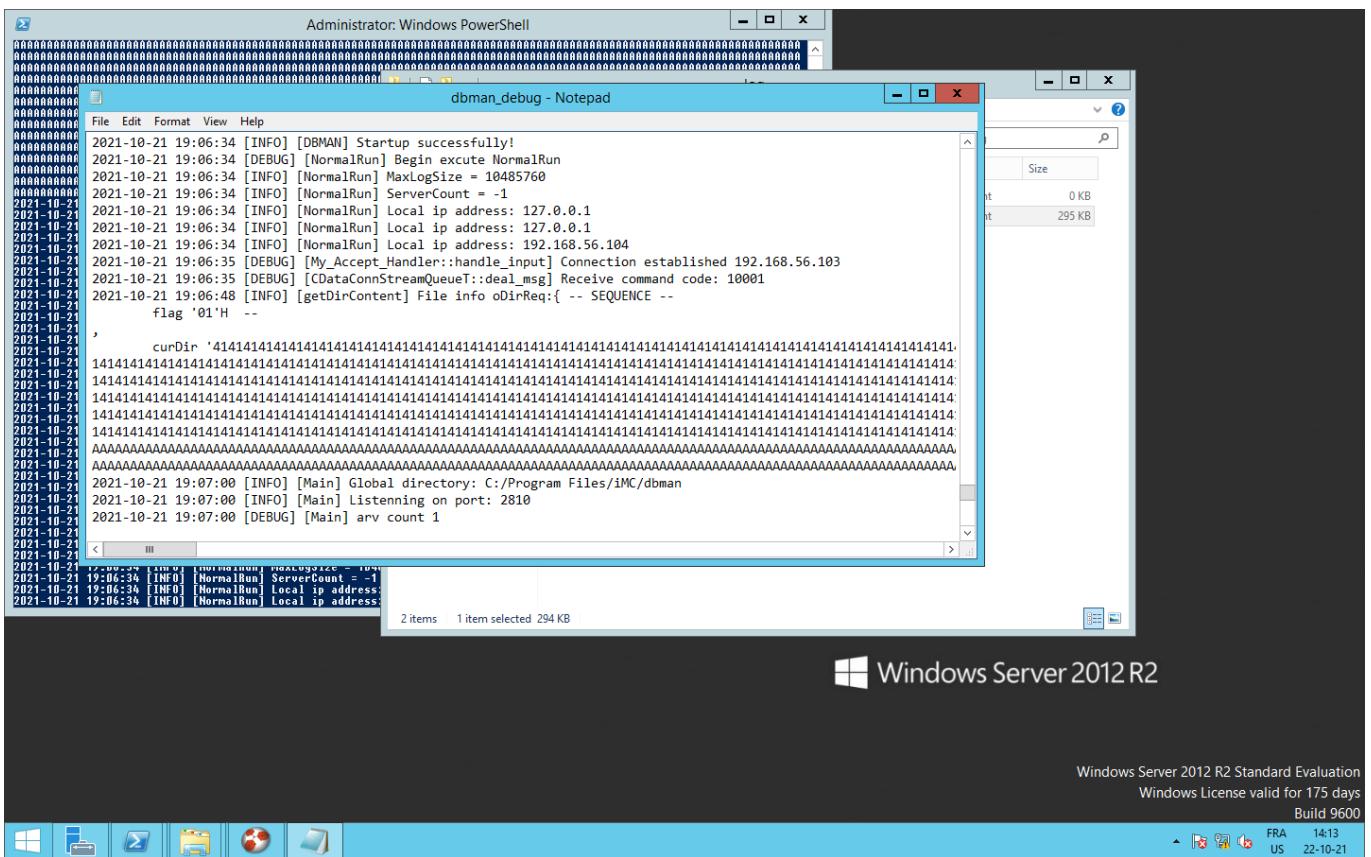


The msec extension tells us that it's potentially exploitable.

Looking at payload in wireshark:



Looking at the dbman_debug log file:



Analyzing potential exploitability:

The exploitability requires at least the following two conditions:

- Can we "control" the instruction pointer?
- Can we store, and have enough space to place our payload/shellcode?

The question of PE file security mitigations comes into play. As previously mentioned, it seems that the *dbman.exe* has been compiled with /GS flag i.e., stack cookie.

A few possibilities exist to bypass this mitigation, one option would be to take advantage of the SEH records to avoid this protection.

Wrap up, backtracking and lessons learned so far

In this case, working backwards is a rather cumbersome process. Manually keeping track of the unfolding processes isn't straightforward.

In the case of UAF, the issue seems to be related to some handlers but couldn't pinpoint the location. Moreover, is this a particular case of a UAF e.g., a double free or other flavor cannot be determined with precision with this RE approach.

The approach taken was/is sub-optimal to say the least. The size of the software at rest should have acted as a warning.

This calls for better Free/Open Source software/tooling(or money), something like `strace` on steroids. Timeless debugging could probably have helped, as briefly discussed hereafter couldn't make use of windbg preview time-travel feature. To my knowledge and personal look up, the options in this field are rather sparse.

Haven't looked yet into this deeply but maybe it could be possible to lower the RE overhead using `panda-re` ; could it act as a timeless debugger.

Encountered Issues

Time-network link related:

- Dealing with large file size took me some time to download(network bandwidth quite low on my side, 4G modem/router)
- setting up the environments, downloading iso/vms.

Windows ISO deployment/installation issue

By using the following ISO:

https://archive.org/details/Windows_Server_2008_R2_x64.iso_reupload

I wasn't able to install stuff because of a missing service in MS Windows. After several attempts, I decided to try another version i.e. WinServer2012R2.

Database related issue

iMC accepts databases from Oracle, SQLServer and MySQL.

I couldn't make it work using different versions of MySQL on the same host(iMC and db on the same machine). Another possibility could have been to use a remote db on a RedHat/CentOS machine.

[GitHub - Justinfact/imc-ilms: HPE IMC Linux MySQL Script \(ILMS\)](#)

OpenVas 21.4.2

After HPE iMC installation, I wanted to quickly check using another logic if I could quickly confirm the vulnerability ; wasn't sucessful.

Tried unsuccessfully(obviously) running("manually") the following script from Tenable:

https://vulners.com/nessus/HP_INTELLIGENT_MANAGEMENT_CENTER_7.3_E0504P04.NASL

Windbg Preview

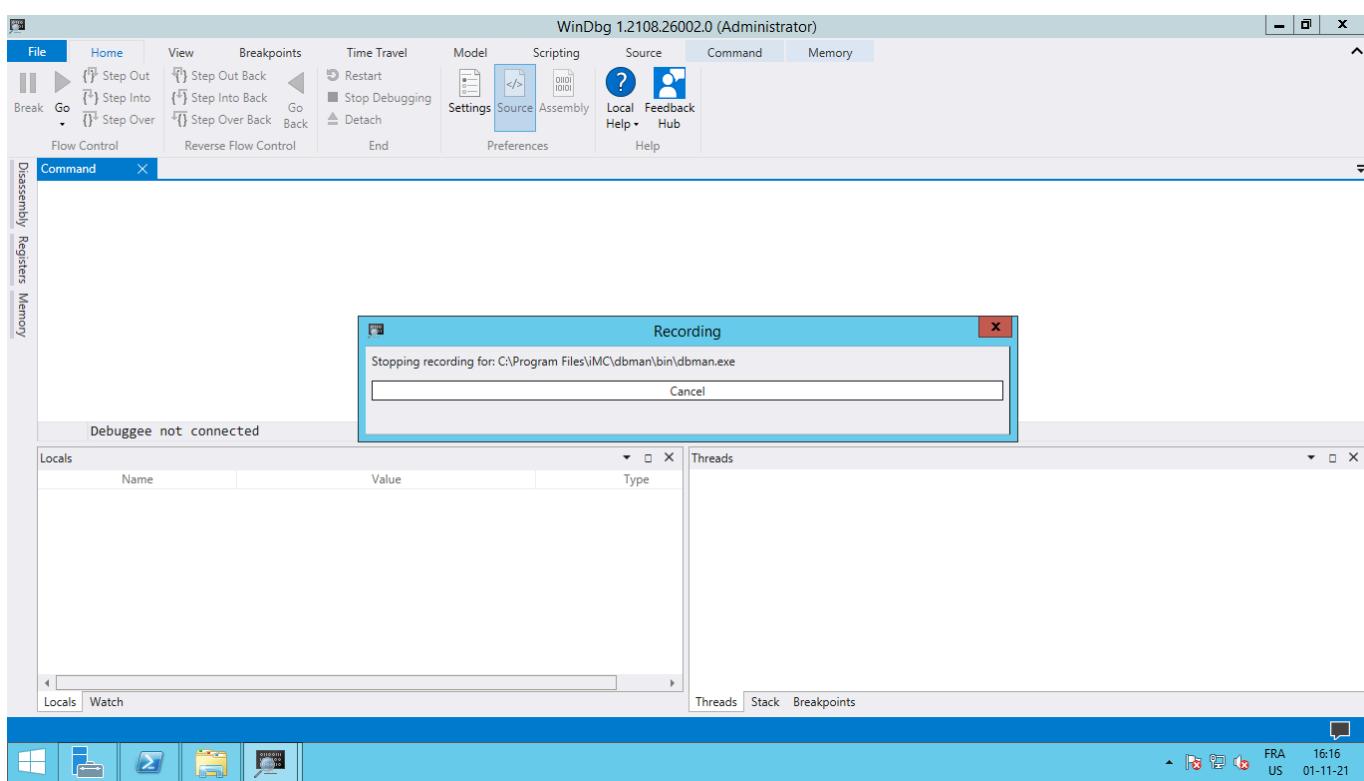
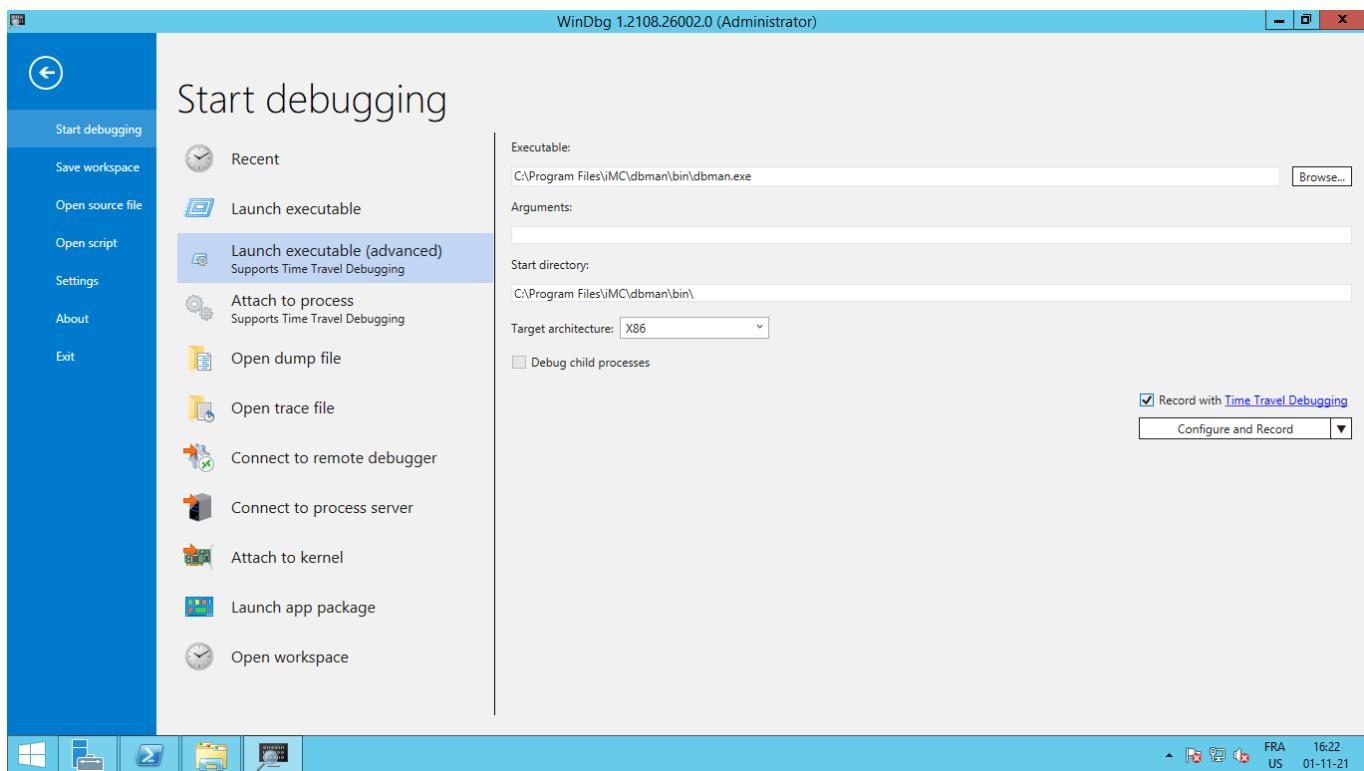
Managed to install windbg preview on MS WindowsServer2012R2 following:

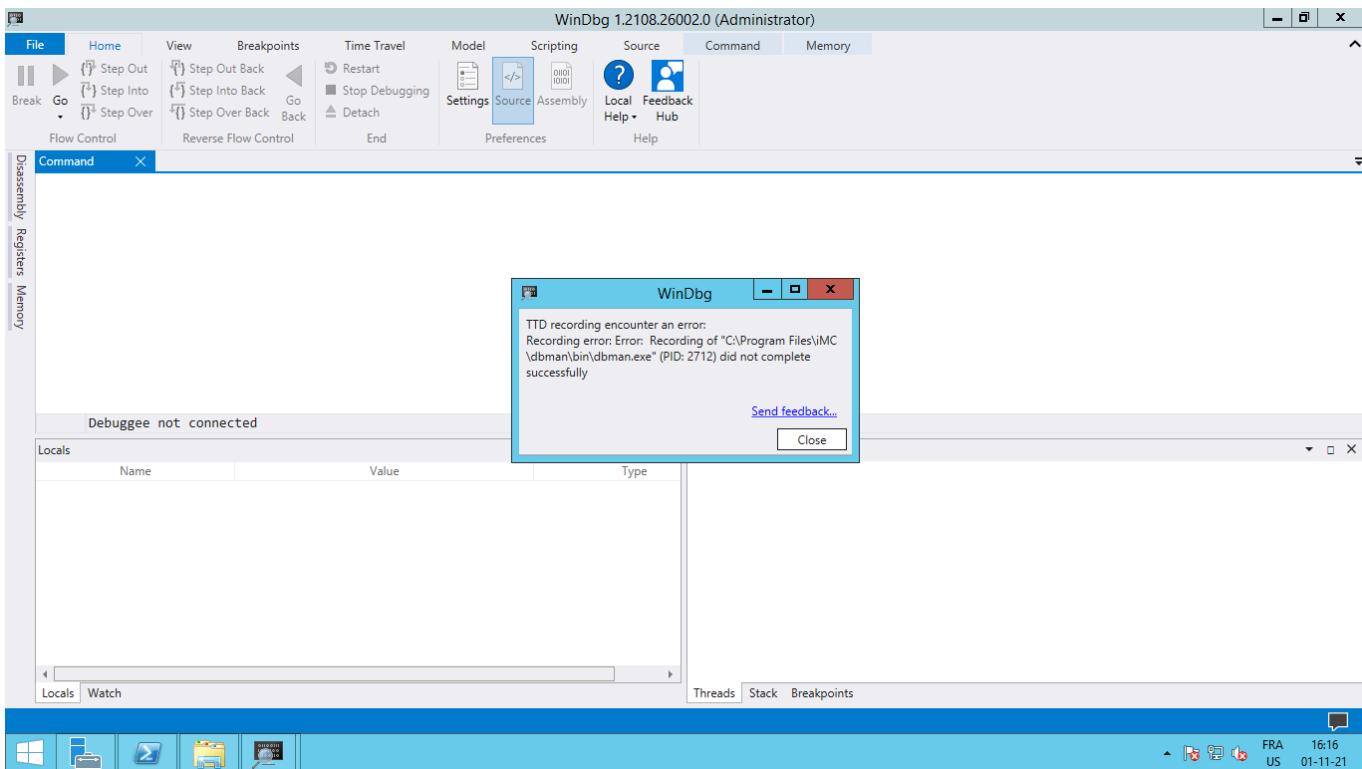
<https://stackoverflow.com/questions/67971698/time-travel-debugging-how-to-install-it-without-the-store>

It comes down to downloading 7zip and the `.appx` file followed by:

- unarchiving the `.appx` using 7zip,
- launch `DbgX.Shell.exe` from within the extracted folder

For some reason, the recording of the time-travel for `dbman.exe` fails:





Using that same feature on a simpler process e.g., a `ping` issued from a `cmd.exe` works.

At this point, the cause of failure hasn't been investigated, maybe it's linked to the daemon/service nature of dbman executable.

Links:

<https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/time-travel-debugging-object-model>

<https://www.microsoft.com/en-us/p/windbg-preview/9pgjgd53tn86>

iMC general deployment notes

Log:

Tried deploying on:

- Win7sp1 (AMD64) - vm
- Win10 (AMD64) - physical host
- Win2k8 (x86) - vm
- Win2kR2 (AMD64) - vm
- Win2k12R2 (AMD64) - vm

using variations for SQLServer versions.

Looking for other tooling

The motivation for accessing iMC under panda-re is because we want to be able to go back in time in the execution flow. As previously described, on a `WinServer2012R2`, it is possible to deploy windbg preview with the time-travel feature.

I couldn't make it work/apply it with `dbman.exe`.

One option that could help is to check if `panda-re` may be of some assistance. Theoretically, this looks like a really powerful tool.

PANDA: Platform for Architecture Neutral Dynamic Analysis

Deploying Intelligent Management Center PLAT 7.3 (E0504P04) to a panda-re friendly format(qcow2) requires non-negligible modifications to our initial setup.

This is because:

- `panda-re` introspection plugin works for a `win7x86` version
- The version used in the initial setup was a `WinServer2012R2` AMD64

From previous analysis, we're interested in a UAF bug which means that we're looking for heap allocation/freeing.

Highlights of the setup

In the optic of using `panda-re` against iMC 7.3 (E0504P04), I am going for a Windows Server 2008 because of the [win7x86intro](#) plugin.

This version of WinServer is motivated by the following:

Windows debuggers

The Windows debuggers can run on x86-based, x64-based, or ARM-based processors, and they can debug code that is running on those same architectures. Sometimes the debugger and the code being debugged run on the same computer, but other times the debugger and the code being debugged run on separate computers. In either case, the computer that is running the debugger is called the *host computer*, and the computer that is being debugged is called the *target computer*. The Windows debuggers support the following versions of Windows for both the host and target computers.

- Windows 10 and Windows Server 2016
- Windows 8.1 and Windows Server 2012 R2
- Windows 8 and Windows Server 2012
- `Windows 7 and Windows Server 2008 R2`

<https://github.com/MicrosoftDocs/windows-driver-docs/blob/staging/windows-driver-docs-pr/debugger/index.md>

Note: the following installation/setup steps won't be as thorough as previously.

High level overview:

- Install a Win2k8(x86), not really a install because it's a VHD
- Quick test to see if `panda-system-i386` can be used to load the image/disk
 - Install iMC 7.3 (E0504P04)
- Install panda-re
 - Some configuration is required(in a first place, on the qemu side)

Win2k8(x86) installation related

Fetching a Win2k8:

<https://www.microsoft.com/en-us/download/details.aspx?id=14527>

Windows Server 2008 Enterprise Edition x86 (Full Install) VHD

Test if it can be used with panda-re:

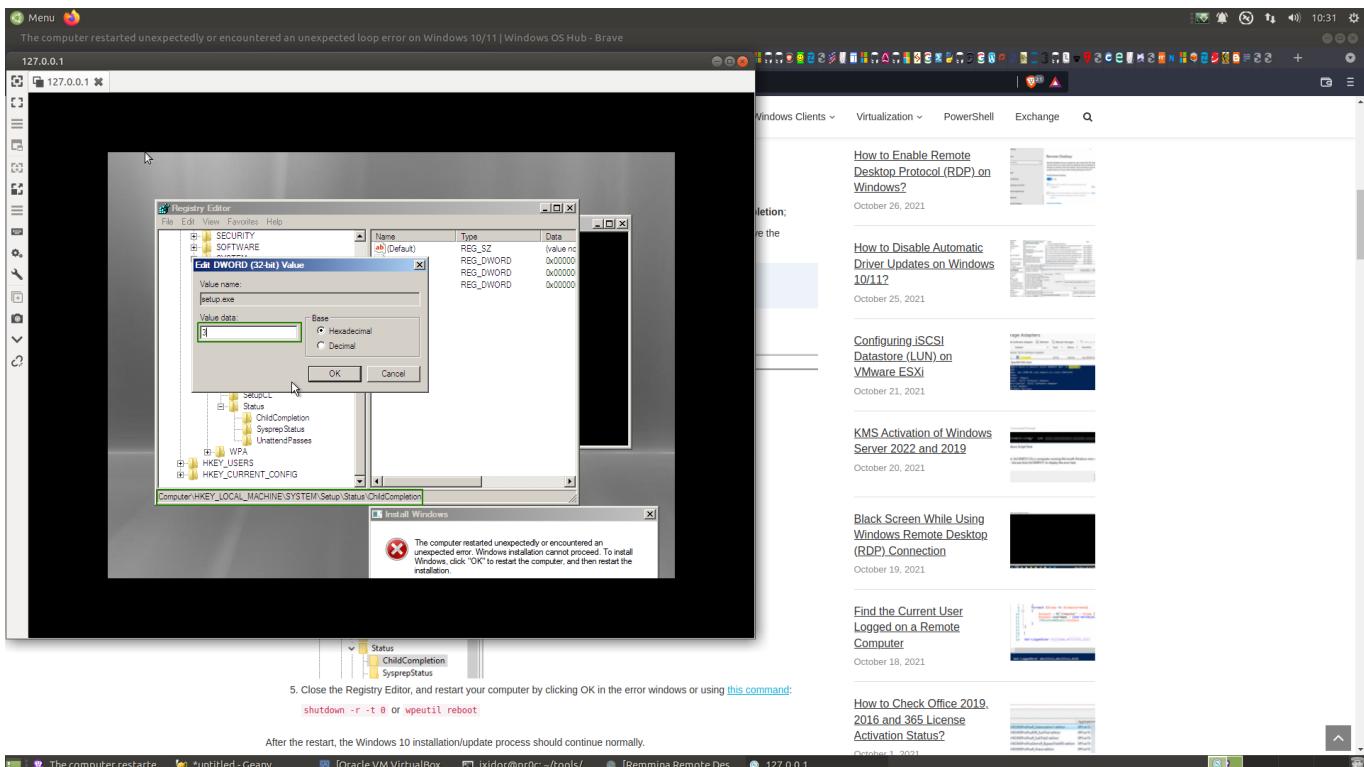
Convert the VHD to qcow2:

```
qemu-img convert -f vpc Windows2008Fullx86Ent.vhd -O qcow2  
Windows2008Fullx86Ent.qcow
```

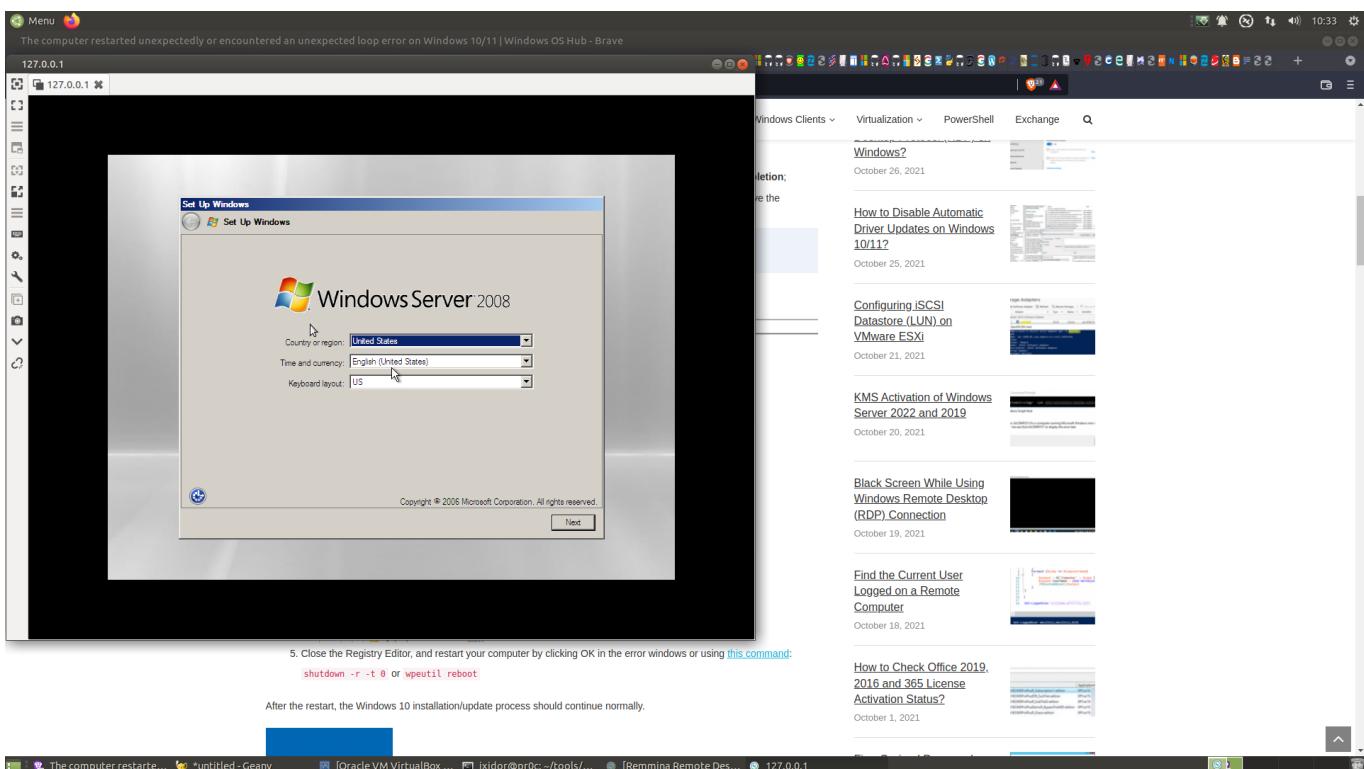
Try loading the image with `panda-system-i386`:

```
ixidor@pr0c:~/tools/panda/build/i386-softmmu$ ./panda-system-i386 -m 4096 -  
monitor stdio -hda ~/virt/bak/Windows2008Fullx86Ent.qcow2
```

The first load triggers the following error, which can be fixed by setting that registry key to 3:



After a reboot, it should proceed smoothly:



```

ixidor@pr0c:~/tools/panda/build/i386-softmmu$ ./panda-system-i386 -m 4096 -monitor stdio -hda
File Edit View Search Terminal Help
wavcapture path [frequency [bits [channels]]] -- capture audio to a wave file (default frequency)
x /fmt addr -- virtual memory dump starting at 'addr'
x_colo_lost_heartbeat -- Tell COLO that heartbeat is lost,
                           a failover or takeover is needed.
xp /fmt addr -- physical memory dump starting at 'addr'
(qemu)
(qemu) q
ixidor@pr0c:~/tools/panda/build/i386-softmmu$ ./panda-system-i386 -m 4096 -monitor stdio -hda
QEMU 0.1.1 monitor - type 'help' for more information
(qemu) VNC server running on 127.0.0.1:5900
help
acl_add aclname match allow/deny [index] -- add a match rule to the access control list
acl_policy aclname allow/deny -- set default access control list policy
acl_remove aclname match -- remove a match rule from the access control list
acl_reset aclname -- reset the access control list
acl_show aclname -- list rules in the access control list
balloon target -- request VM to change its memory allocation (in MB)
begin_record [file_name] -- begin recording for later replay
begin_record_from [snapshot] [file_name] -- begin recording from given snapshot for later replay
begin_replay [file_name] -- begin to replay a record
block_job_cancel [-f] device -- stop an active background block operation (use -f
                             if the operations are currently paused)
block_job_complete device -- stop an active background block operation
block_job_pause device -- pause an active background block operation
block_job_resume device -- resume a paused background block operation
block_job_set_speed device speed -- set maximum speed for a background block operation
block_passwd block_passwd device password -- set the password of encrypted block devices
block_resize device size -- resize a block image
block_set_io_throttle device bps bps_rd bps_wr lops iops_rd tops_wr -- change I/O throttle limits
block_stream device [speed [base]] -- copy data from a backing file into a block device
boot_set_bootdevice -- define new values for the boot device list
change_device filename [format [read-only-mode]] -- change a removable medium, optional format
chardev-add args -- add chardev
chardev-del id -- remove chardev
client_migrate_info protocol hostname port tls-port cert-subject -- set migration information
closedfd closefd name -- close a file descriptor previously passed via SCM rights
commit-device all -- commit changes to the disk images (if -snapshot is used) or backing files
cpu index -- set the default CPU
cpu-add id -- add cpu
c|con -- resume emulation
delvm tag|id -- delete a VM snapshot from its tag or id
device_add driver[...][prop=value][...] -- add device, like -device on the command line
device_del device -- remove device
drive_add [-n] [-domain:] [-bus:] [-slot:]
        [-file:] [-if-type:] [-bus:]
        [-unit:] [-medium:] [-index:]
        [-cyls:] [-heads:] [-secs:] [-trans:]
        [-snapshot:on|off] [-cache:on|off]
        [-readonly:on|off] [-copy-on-read:on|off] -- add drive to PCI storage controller
drive_backup [-n] [-f] [-c] device target [format] -- initiates a point-in-time
        copy for a device. The device's contents are
        copied to the new image file, excluding data that
        is written after the command is started.
        The -n flag requests QEMU to reuse the image found
snapshot_blkdev [-n] device [new-image-file] [format] -- initiates a live snapshot
        of device. If a new image file is specified, the
        new image file will become the new root image.
        If format is specified, the snapshot file will
        be created in that format.
        The default format is qcows2. The -n flag requests QEMU
        to reuse the image found in new-image-file, instead of
        recreating it from scratch.
snapshot_blkdev_internal device name -- take an internal snapshot of device.
        The format of the image used by device must
        support it, such as qcows2.

snapshot_delete_blkdev_internal device name [id] -- delete an internal snapshot of device.
        If id is specified, qemu will try delete
        the snapshot matching both id and name.
        The format of the image used by device must
        support it, such as qcows2.

stop -- stop emulation
stopcapture capture index -- stop capture
sum addr size -- compute the checksum of a memory region
system_powerdown -- send system power down event
system_reset -- reset the system
system_wakeup -- wakeup guest from suspend
trace-event name on|off vcpu -- changes status of a specific trace event (vcpu: vCPU to set,
unload plugin) index: unload a plugin
usb-add-service -- add USB device (e.g., 'usb3ibus:addr' or 'host:vendor_id:product_id')
usb-del-device -- remove USB device 'bus:addr'
watchdog_action [reset|shutdown|poweroff|powerdown|none] -- change watchdog action
wavcapture path [frequency [bits [channels]]] -- capture audio to a wave file (default frequency)
x /fmt addr -- virtual memory dump starting at 'addr'
x_colo_lost_heartbeat -- Tell COLO that heartbeat is lost,
                           a failover or takeover is needed.
xp /fmt addr -- physical memory dump starting at 'addr'
(qemu)
(qemu) q

```

```

ixidor@pr0c:~/tools/panda/build/i386-softmmu$ ./panda-system-i386 -m 4096 -monitor stdio -hda
File Edit View Search Terminal Help
qemu-io [device] "[command]" -- run a qemu-io command on a block device
qom-list path -- list QOM properties
qom-set prop property value -- set QOM property
q|quit -- quit the emulator
ringbuf_read device size -- Read from a ring buffer character device
ringbuf_write device data -- Write to a ring buffer character device
savevm [tag|id] -- save a VM snapshot. If no tag or id are provided, a new snapshot is created
secondvm [filename] -- save screen into PPM image "filename"
sendkey keys [hold_ms] -- send keys to the VM (e.g. 'sendkey ctrl-alt-f1', default hold time=1000ms)
set_link name on|off -- change the link status of a network adapter
set_password protocol password action-if-connected -- set spice/vnc password
singlestep [on|off] -- run emulation in singlestep mode or switch to normal mode
snapshot_blkdev [-n] device [new-image-file] [format] -- initiates a live snapshot
        of device. If a new image file is specified, the
        new image file will become the new root image.
        If format is specified, the snapshot file will
        be created in that format.
        The default format is qcows2. The -n flag requests QEMU
        to reuse the image found in new-image-file, instead of
        recreating it from scratch.
snapshot_blkdev_internal device name -- take an internal snapshot of device.
        The format of the image used by device must
        support it, such as qcows2.

snapshot_delete_blkdev_internal device name [id] -- delete an internal snapshot of device.
        If id is specified, qemu will try delete
        the snapshot matching both id and name.
        The format of the image used by device must
        support it, such as qcows2.

stop -- stop emulation
stopcapture capture index -- stop capture
sum addr size -- compute the checksum of a memory region
system_powerdown -- send system power down event
system_reset -- reset the system
system_wakeup -- wakeup guest from suspend
trace-event name on|off vcpu -- changes status of a specific trace event (vcpu: vCPU to set,
unload plugin) index: unload a plugin
usb-add-service -- add USB device (e.g., 'usb3ibus:addr' or 'host:vendor_id:product_id')
usb-del-device -- remove USB device 'bus:addr'
watchdog_action [reset|shutdown|poweroff|powerdown|none] -- change watchdog action
wavcapture path [frequency [bits [channels]]] -- capture audio to a wave file (default frequency)
x /fmt addr -- virtual memory dump starting at 'addr'
x_colo_lost_heartbeat -- Tell COLO that heartbeat is lost,
                           a failover or takeover is needed.
xp /fmt addr -- physical memory dump starting at 'addr'
(qemu)
(qemu) q

```

At this point, we notice that the performance are rather slow.

To proceed with the installation of iMC, I chose to reconvert the image to a **VDI**, perform the installation in VirtualBox ; and finally, reconvert back to **qcows2** to do the analysis.

Convert qcows2 to VDI:

```
qemu-img convert -f qcow2 Windows2008Fullx86Ent.qcow2 -O vdi  
Windows2008Fullx86Ent.vdi
```

We still want windbg to see if we observe the same behavior as previously.

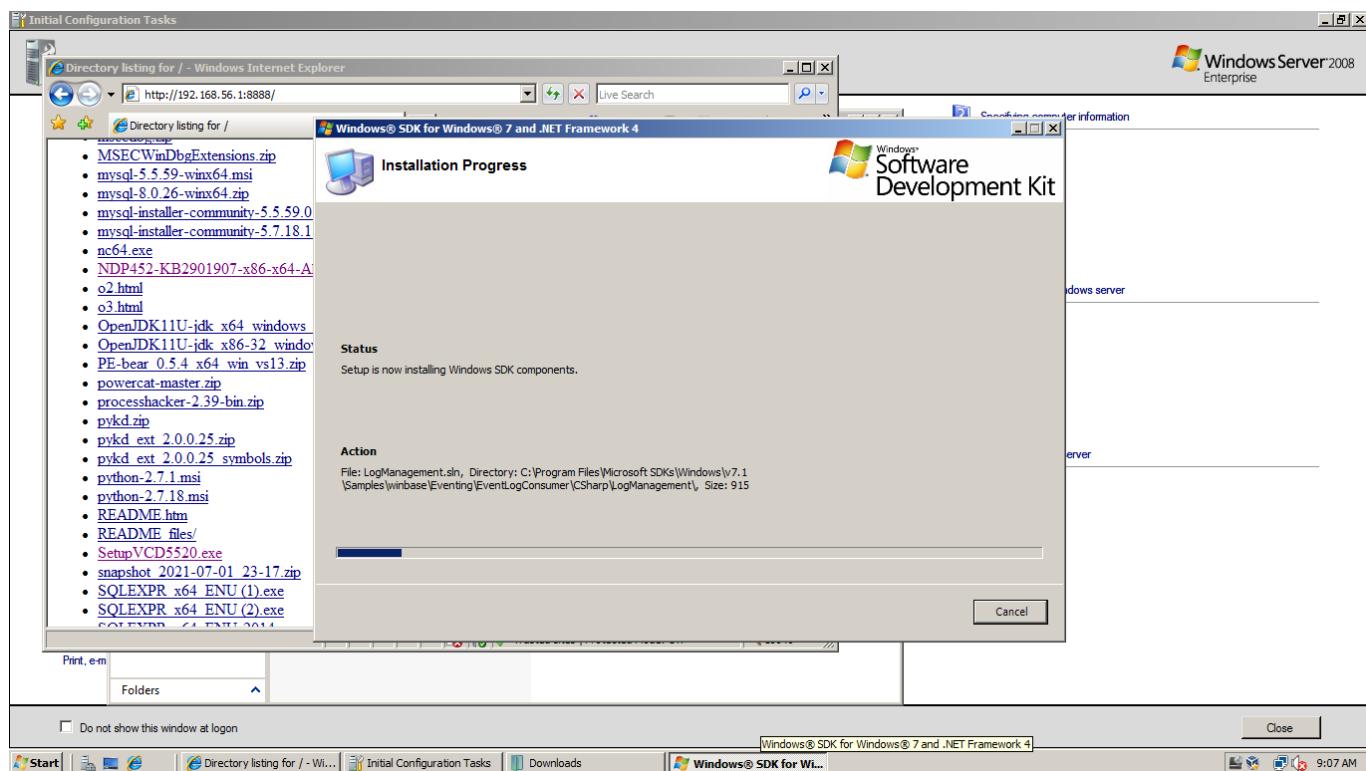
To get windbg:

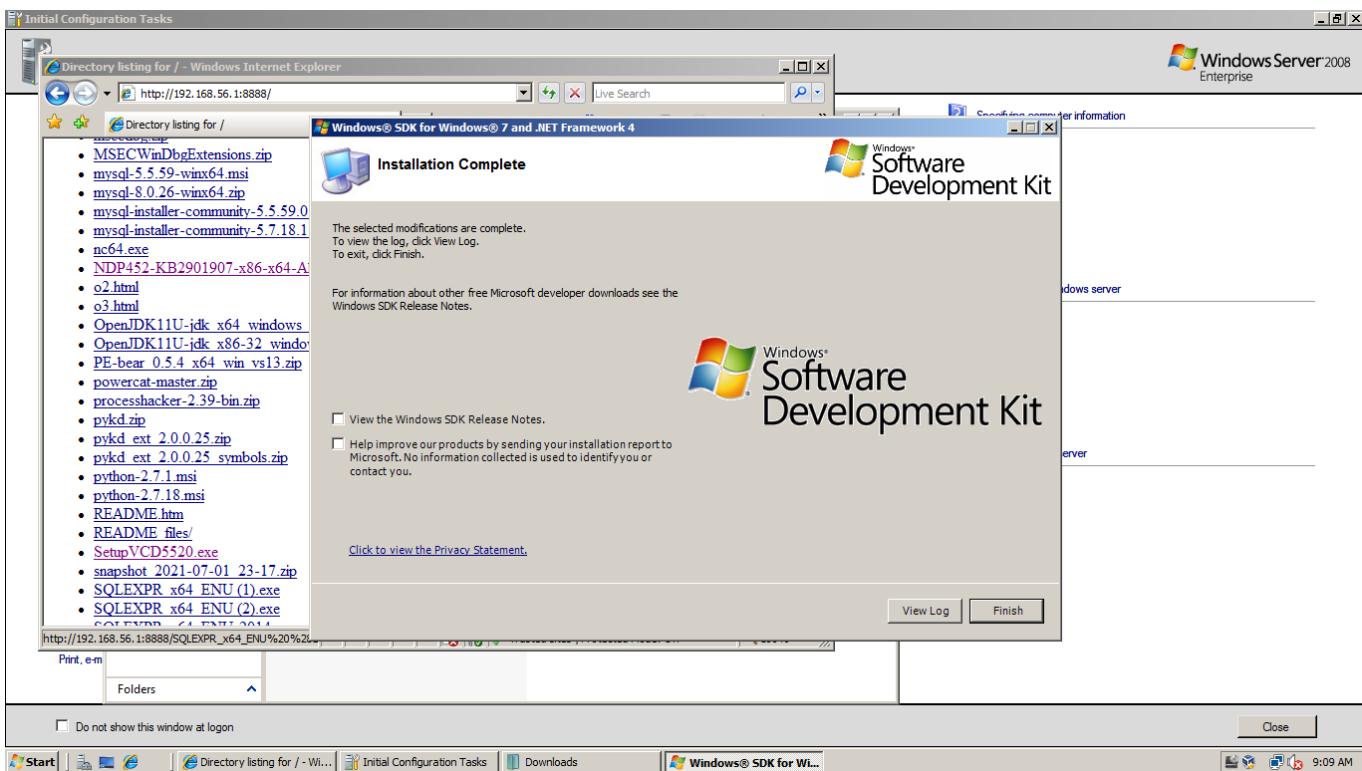
<https://www.elby.ch/en/products/vcd.html>

VirtualCloneDrive

<https://www.microsoft.com/en-us/download/details.aspx?id=8442>

Microsoft Windows SDK for Windows 7 and .NET Framework 4 (ISO)





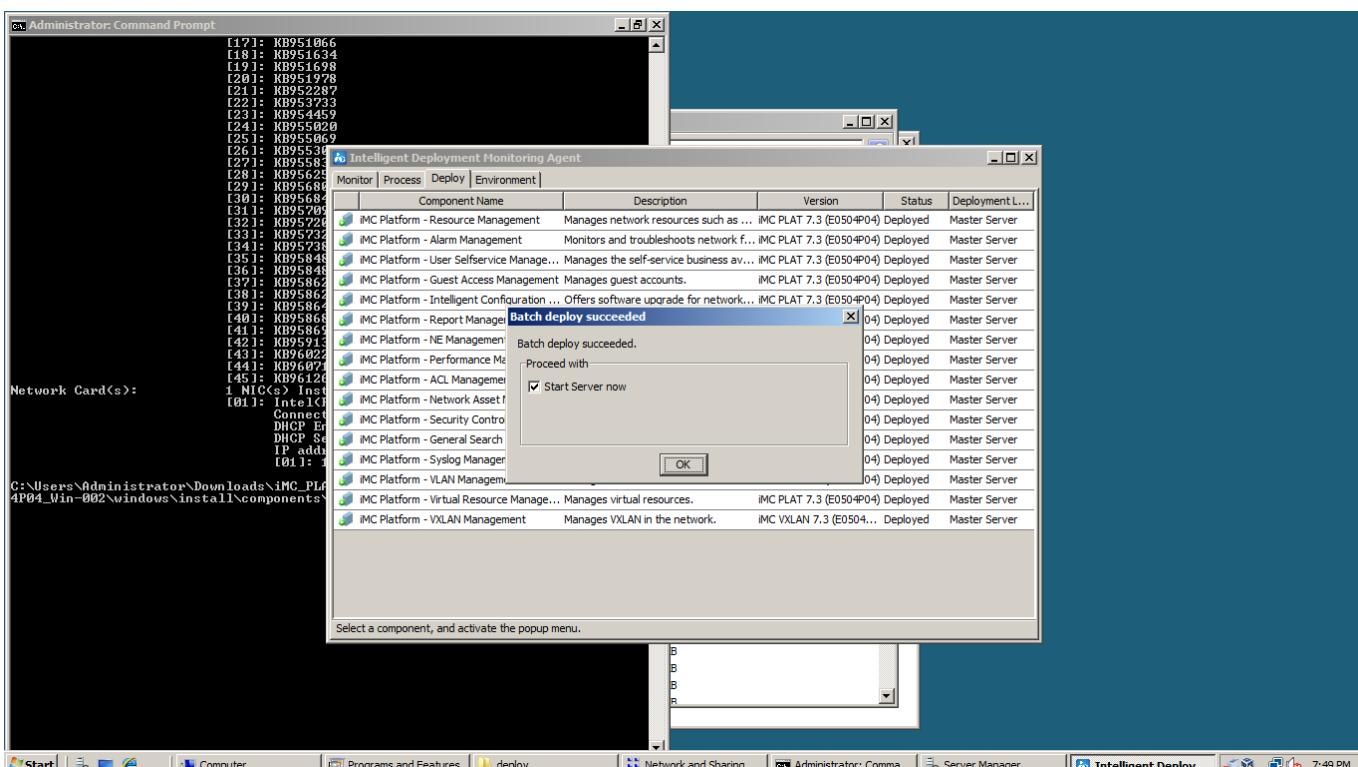
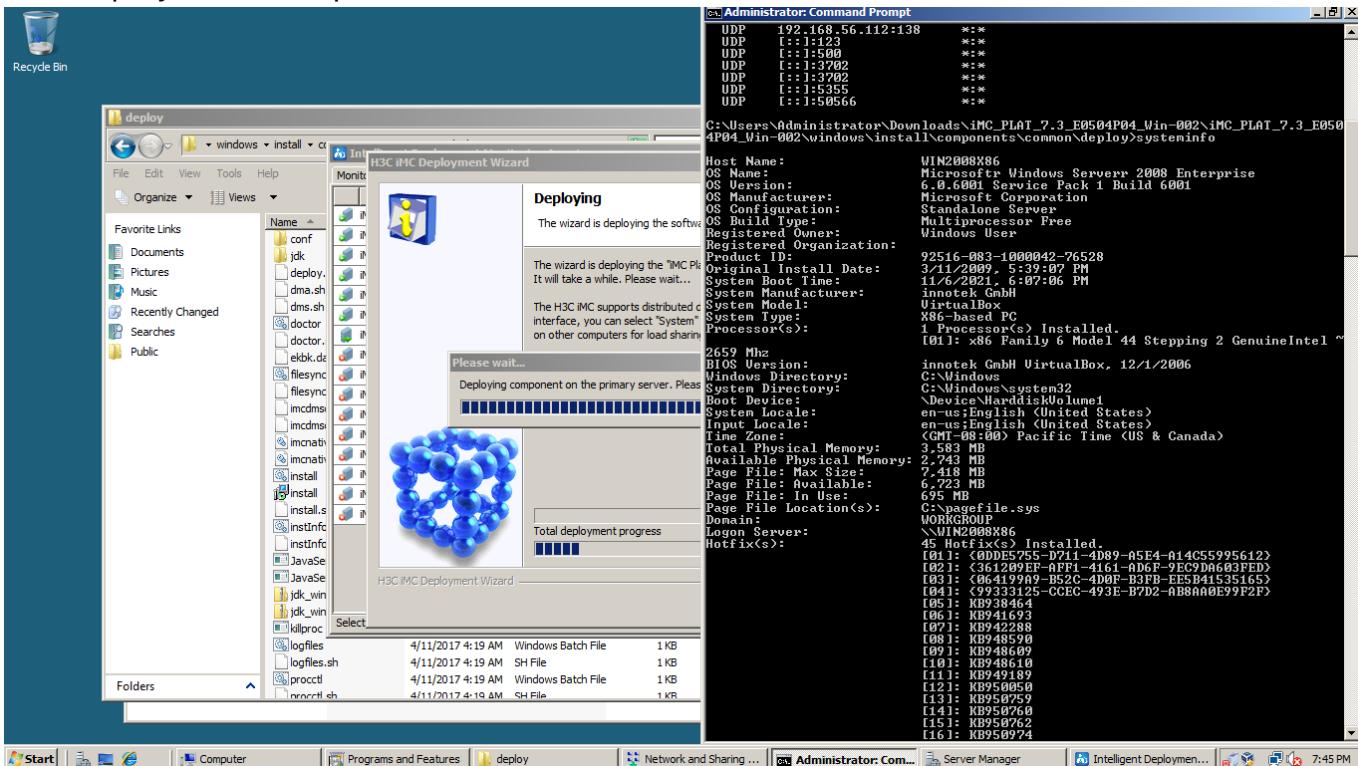
<https://www.microsoft.com/en-us/download/details.aspx?id=11800>

Windows Driver Kit Version 7.1.0

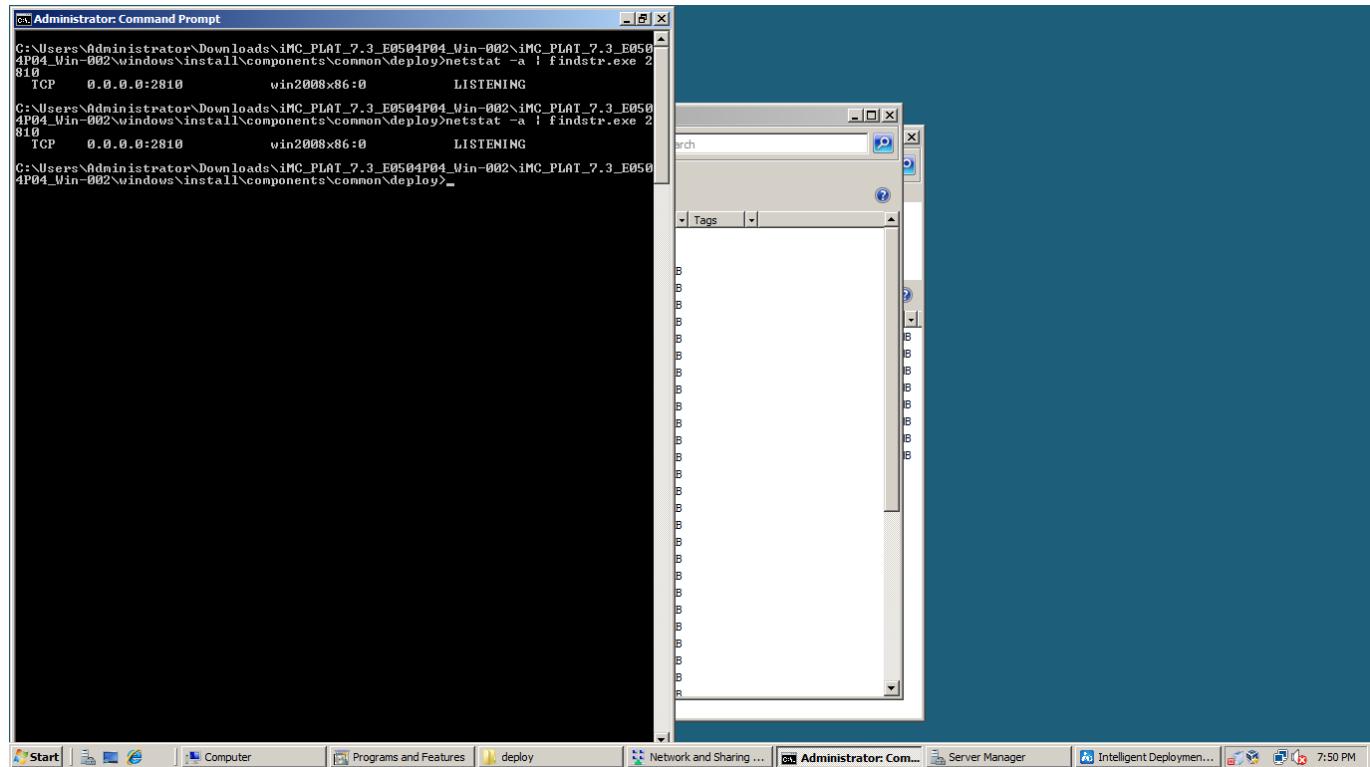
iMC 7.3 (E0504P04)

The installation of iMC is almost identical to the Win2k12R2 version except that we're using the SQLServerExpress2008R2 database software provided in one of the iMC archive.

iMC deployment final phases:



dbman.exe is running:



Finally, we convert vdi to qcows2:

```
qemu-img convert -f vdi Windows2008Fullx86Ent.vdi -O qcows2
```

```
Windows2008Fullx86Ent.qcows2
```

Note: it is likely that we could avoid converting from one format to another but that path hasn't been explored.

Armed with `Windows2008Fullx86Ent.qcows2`, we can feed it to `panda-system-i386`.

panda-re

panda-re -- install

Installing panda-re on Ubuntu 20.04.3 LTS:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.3 LTS
Release:       20.04
Codename:      focal
```

Clone:

```
git clone https://github.com/panda-re/panda
```

I made two pull requests to the main branch because of a small execution flow and permission issue in `install_ubuntu.sh`.

There could be a possible third pull request but I assume that people perform an `apt-get update && apt-get upgrade -y` before launching the `install_ubuntu.sh` script.

The installation takes a while(getting dependencies + compilation) and requires space up to twenty gigas ; this is a rough estimate.

panda-re -- config

In order to be able to load/boot/emulate the `Windows2008Fullx86Ent.qcow2` machine, a number of arguments for `panda-system-i386` are needed.

Version:

```
ixidor@pr0c:~/tools/panda/build/i386-softmmu$ ./panda-system-i386 -version
QEMU emulator version 2.9.1 (-dirty)
Build date Nov  5 2021
Copyright (c) 2003-2017 Fabrice Bellard and the QEMU Project developers
```

I am purposely not going into the details of this command line:

```
ixidor@pr0c:~/tools/panda/build/i386-softmmu$ sudo ./panda-system-i386 -enable-kvm
-m 4096 -net nic -net tap,ifname=tap0,script=no -usbdevice tablet -monitor stdio -
-vga virtio -hda ~/virt/bak/Windows2008Fullx86Ent.qcow2
```

- `-enable-kvm`: use kvm acceleration, increases performance
- `-m`: dedicated memory
- `-net nic -net tap,ifname=tap0,script=no`: adding a network card to the guest and communicating through a tap interface(L2), avoid calling `panda-bridge-helper`; there is at least one cleaner way to do this but this works
- `-usbdevice tablet`: sync mouse cursors between the host and the guest
- `-vga virtio`: enable the possibility to increase screen resolution

Looking for the bug through panda/qemu hypervisor

Trigger av in `dbman.exe` in windbg within the `qemu` environment:

```

ModLoad: 758a0000 758a5000 C:\Windows\System32\wship6.dll
(a70.b9c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=f0f0f0 ebx=00000003 ecx=00000184 edx=f0f0f0f0 esi=0312daa0 edi=03120884
eip=6cfeb81d esp=03cffdc8 ebp=0312024c icpl=0 nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000 efl=00010206
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\iMC\dbman\bin\ACE_v6.dll -
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x16d:
6cfeb81d 8b5228      mov     edx,dword ptr [edx+28h] ds:0023:f0f0f118=???????
0:001> kb
ChildEBP RetAddr  Args to Child
WARNING: Stack unwind information not available. Following frames may be wrong.
03cffde8 6cfaef8 031201a0 6cfed419 6bf69669 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x16d
03cffdf0 6cfed419 6bf69669 00000000 031201a0 ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes+0x8
03cfffe1c 6cfafe18 6cfafe1b2 6bf695cd 00000000 ACE_v6!ACE_WFMO_Reactor::update_state+0x119
00000000 00000000 00000000 ACE_v6!ACE_WFMO_Reactor::safe_dispatch+0x88
0:001> u
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x16d:
6cfeb81d 8b5228      mov     edx,dword ptr [edx+28h]
6cfeb820 85c0          test    eax,eax
6cfeb822 8b442420      mov     eax,dword ptr [esp+20h]
6cfeb826 50             push    eax
6cfeb827 51             push    ecx
6cfeb828 8bc8          mov     ecx,esi
6cfeb82a 0f94c3        sete   bl
6cfeb82d ffd2          call    edx
0:001> ub eip . L 16
ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x129:
6cfeb7d9 8d0c8a        lea     ecx,[ecx+ecx*4]           [ ]
6cfeb7dc 33d2          xor    edx,edx
6cfeb7de c6411800      mov     byte ptr [ecx+18h].0
6cfeb7e2 895104        mov     dword ptr [ecx+4],edx
6cfeb7e5 8811          mov     byte ptr [ecx].dl
6cfeb7e7 897108        mov     dword ptr [ecx+8].esi
6cfeb7ea 89510c        mov     dword ptr [ecx+0Ch].edx
6cfeb7ed 885110        mov     byte ptr [ecx+10h].dl
6cfeb7f0 885111        mov     byte ptr [ecx+11h].dl
6cfeb7f3 895114        mov     dword ptr [ecx+14h].edx
6cfeb7f6 8b4d0c        mov     ecx,dword ptr [ebp+0Ch]
6cfeb7f9 893481        mov     dword ptr [ecx+eax*4].esi
6cfeb7fc 017514        add    dword ptr [ebp+14h].esi
6cfeb7ff 8b4c2414      mov     ecx,dword ptr [esp+14h]
6cfeb803 3bca          cmp    ecx,edx
6cfeb805 7439          je    ACE_v6!ACE_WFMO_Reactor_Handler_Repository::make_changes_in_current_infos+0x190 (6cf840)
6cfeb807 e8b4dafaff    call    ACE_v6!ACE_Event_Handler::reference_counting_policy (6cf992c0)
6cfeb80c 8b08          mov     ecx,eax
6cfeb80e e83dfef8ff    call    ACE_v6!ACE_Process_Options::creation_flags (6cf7b650)
6cfeb813 8b742414      mov     esi,dword ptr [esp+14h]
6cfeb817 8b4c2418      mov     ecx,dword ptr [esp+18h]
6cfeb81b 8b16          mov     edx,dword ptr [esi]

```

The result is in the same vein of experiments conducted in the win2k12R2 version.
At this point, except the hypervisor and the MS Windows version nothing has changed.

We should be able to make use of some [panda](#) magic.

Other links

<http://woshub.com/windows-install-error-computer-restarted-unexpectedly/>

<https://techpiezo.com/linux/convert-disk-images-to-various-formats-using-qemu-img/>

<https://web.archive.org/web/20210125093418/http://download.microsoft.com/download/2/6/1/261fca42-22c0-4f91-9451-0e0f2e08356d/Windows6.0-KB942288-v2-x86.msu>

Windows Installer 4.5

[QEMU's new -nic command line option - QEMU](#)

Looking at Java Classes

In this context, the iMC software makes use of both Open Source:

```
C:\Program Files\iMC\server\bin\
```

```
ACE.dll  
ACE_SSL.dll  
ACE_v6.dll
```

even though compiled, it probably refers to the following software stack:

The ADAPTIVE Communication Environment

<http://www.dre.vanderbilt.edu/~schmidt/ACE-overview.html>

and probably closed source software:

```
C:\Program Files\iMC\anti-vm\bin\anti-vm.exe
```

```
md5sum  
dfffccea7c76052be489028287acbee31 anti-vm.exe
```

Java Classes

In this greybox setting, we can look at *jars*, e.g.:

```
C:\Program Files\iMC\deploy\deploy.jar
```

Using tools such as `gadgetinspector`:

Tool description taken from:

```
https://github.com/JackOfMostTrades/gadgetinspector
```

Gadget Inspector

This project inspects Java libraries and classpaths for gadget chains. Gadgets chains are used to construct exploits for deserialization vulnerabilities. By automatically discovering possible gadgets chains in an application's classpath penetration testers can quickly construct exploits and application security engineers can assess the impact of a deserialization vulnerability and prioritize its remediation.

This project was presented at Black Hat USA 2018. Learn more about it there! (Links pending)

DISCLAIMER: This project is alpha at best. It needs tests and documentation added. Feel free to help by adding either!

Running the tool:

```
(kali㉿kali)-[~/gadgetinspector]
$ nl gadget-chains.txt
 1 org/apache/commons/lang/enums/Enum.equals(Ljava/lang/Object;)Z (1)
 2 org/apache/commons/lang/enums/Enum.getNameInOtherClassLoader(Ljava/lang/Object;)Ljava/lang/String; (1)
 3 java/lang/reflect/Method.invoke(Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object; (0)

 4 javax/security/auth/kerberos/KerberosTicket.equals(Ljava/lang/Object;)Z (1)
 5 javax/security/auth/kerberos/KerberosTicket.getRenewTill()Ljava/util/Date; (0)
 6 java/util/Date.clone()Ljava/lang/Object; (0)
 7 org/jfree/chart/plot/CategoryPlot.clone()Ljava/lang/Object; (0)
 8 org/jfree/util/ObjectUtilities.clone(Ljava/lang/Object;)Ljava/lang/Object; (0)
 9 java/lang/reflect/Method.invoke(Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object; (0)

10 org/springframework/aop/framework/JdkDynamicAopProxy.invoke(Ljava/lang/Object;Ljava/lang/reflect/Method;[Ljava/lang/Object;)Ljava/lang/Object; (0)
11 org/springframework/aop/target/ThreadLocalTargetSource.getTarget()Ljava/lang/Object; (0)
12 edu/emory/mathcs/backport/java/util/concurrent/ConcurrentSkipListSet$ConcurrentSkipListSubSet.add(Ljava/lang/Object;)Z (0)
13 edu/emory/mathcs/backport/java/util/concurrent/ConcurrentSkipListMap$ConcurrentSkipListSubMap.put(Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (0)
14 edu/emory/mathcs/backport/java/util/concurrent/ConcurrentSkipListMap$ConcurrentSkipListSubMap.checkKey(Ljava/lang/Object;)V (0)
15 edu/emory/mathcs/backport/java/util/concurrent/ConcurrentSkipListMap$ConcurrentSkipListSubMap.inHalfOpenRange(Ljava/lang/Object;)Z (0)
16 edu/emory/mathcs/backport/java/util/concurrent/ConcurrentSkipListMap$ConcurrentSkipListSubMap.inHalfOpenRange(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Z (2)
17 edu/emory/mathcs/backport/java/util/concurrent/ConcurrentSkipListMap.compare(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)I (2)
18 org/apache/commons/lang/enums/ValuedEnum.compareTo(Ljava/lang/Object;)I (1)
19 org/apache/commons/lang/enums/ValuedEnum.getValueInOtherClassLoader(Ljava/lang/Object;)I (1)
20 java/lang/reflect/Method.invoke(Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object; (0)

21 org/apache/commons/lang/enum/Enum.equals(Ljava/lang/Object;)Z (1)
22 org/apache/commons/lang/enum/Enum.getNameInOtherClassLoader(Ljava/lang/Object;)Ljava/lang/String; (1)
23 java/lang/reflect/Method.invoke(Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (0)

24 javax/security/auth/kerberos/KerberosTicket.equals(Ljava/lang/Object;)Z (1)
25 javax/security/auth/kerberos/KerberosTicket.getRenewTill()Ljava/util/Date; (0)
26 java/util/Date.clone()Ljava/lang/Object; (0)
27 org/jfree/data/KeyToGroupMap.clone()Ljava/lang/Object; (0)
28 org/jfree/data/KeyToGroupMap.clone(Ljava/lang/Object;)Ljava/lang/Object; (0)
29 java/lang/reflect/Method.invoke(Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object; (0)
```

As stated in the project's github page, this doesn't mean that there actually is a vulnerability. **gadgetinspector** looks for gadgets inside Java classes and builds up a chain. The question if this chain can be **triggered** needs to be tested in an operational environment.

Links and References

<https://www.zerodayinitiative.com/advisories/ZDI-17-836/>

<https://www.cvedetails.com/cve/CVE-2017-12561/>

<https://github.com/pwnslinger/exploit-repo/tree/master/CVE-2017-12561>

<https://raw.githubusercontent.com/googleprojectzero/0days-in-the-wild/main/0day-RCAs/template.md>

<https://medium.com/tenable-techblog/hpe-imc-hunting-the-hunted-2619889802ab>

[https://deepsec.net/docs/Slides/2012/DeepSec_2012_Rosario_Valotta_-_Taking_Browsers_Fuzzing_to_the_next_\(DOM\)_Level.pdf](https://deepsec.net/docs/Slides/2012/DeepSec_2012_Rosario_Valotta_-_Taking_Browsers_Fuzzing_to_the_next_(DOM)_Level.pdf)

<https://www.itu.int/rec/T-REC-X.690>

<https://www.cybersecurity-help.cz/vdb/SB2017100402>

Papers

Finding the needle in the heap: combining binary analysis techniques to trigger use-after-free

<https://tel.archives-ouvertes.fr/tel-01681707v2/document>

<https://hal.univ-grenoble-alpes.fr/hal-01721539/document>

Document Editing

<https://obsidian.md/>

<https://typora.io/>