Practical 10 Minutes Security Audit Oracle Case

Cesar Cerrudo
Argeniss

Overview

- Introduction
- The technique
- Finding 0days in Oracle
- Getting technical
- Owning Oracle
- Conclusions
- References

Introduction

- Sometimes it's needed a way to infer how trustable and secure a software is before purchasing and/or deploying
- A full auditing takes a lot of time and resources
- A quick and very easy audit technique can help
  - It can be done by non very technically skilled people
  - It reduces auditing time and costs
  - Many of these kind of techniques can be combined for better results
  - If you can find issues in a couple of minutes then you can be almost sure that the software is not very secure

The technique

- This technique is for easily and quickly auditing Windows applications
- It is as simple as looking at process objects identifying weak permissions
  - Weak permissions allow object manipulation by unprivileged users
    - Changing permissions on objects can crash the process
    - Depending on the object type sometimes is even possible to get arbitrary code execution as it will be demonstrated later

The technique

- The following tools are needed:
  - Process Explorer
  - WinObj
  - Pipeacl Tools
- Install and run the software to be audited
- Identify software processes
  - Mostly we should care about privileged process like services
  - Regular processes should be audited if the application will be used in a shared environment such as Terminal Services, Citrix, etc.
  - Demo

The technique

- Start looking at process objects permissions
  - Look at named objects created by the process that can be opened from other processes such as events, mutexes, semaphores, sections, pipes, threads, etc.
  - Demo
- Identify weak permissions
  - Look for low privileged accounts with “Change Permissions” or “Write DACL” permissions
  - If no groups or user accounts are listed then the object was created with a null DACL
    - Then all users have full control over the object
  - Demo

The technique

- Change permissions on objects found and interact with the audited application
  - Process Explorer doesn't let to edit permissions on some objects
    - WinObj and Pipeacl tools can help
  - Look if the application crash or stop responding

Findings 0days in Oracle

- Let's see the technique in action
- Let's audit Oracle 10g R2
  - Extremely secure software
  - In house audited with next generation tools
  - The proud of Oracle security engineering
  - Hard challenge for finding vulnerabilities
  - It makes Windows unbreakable
- Demo

Getting technical

- Objects weak permissions problem is because improper use of SetSecurityDescriptorDacl() function
  - If third function parameter (pDacl) is set as NULL a NULL DACL is assigned to the security descriptor and no protection is assigned to the object
  - Documented on MSDN
    - It seems some Oracle people is allergic to read Microsoft related stuff
  - Identifying bad usage of SetSecurityDescriptorDacl() function is a 5 minutes IDA job
    - Demo

Getting technical

- Oracle has always nice surprises for us
  - SetKernelObjectSecurity() is being used for changing the DACL on the process
  - Looking at process permissions we can see Everyone group has PROCESS_DUP_HANDLE rights
  - Why would someone do that?
    - Maybe it's on Oracle superior secure coding guides
    - Very bad design and coding
  - Let's see now how to exploit it

Owning Oracle

- With PROCESS_DUP_HANDLE rights, how can we get arbitrary code execution?
  - We can duplicate data files handles and read all the data but we want arbitrary code execution
  - We can duplicate impersonation tokens but low privileged users can't impersonate :(
  - What about duplicating a thread and changing context to execute our code?
    - We only need a way to put our code at known location
    - We can put the code in the shared section we previously saw (remember it has full permissions for Everyone)
  - Demo

Conclusions

- ***Very easy and quick technique***
- ***Just making click on proper tools you can quickly identify these vulnerabilities***
- ***If you like to work at low level, using IDA to identify these vulnerabilities is even faster***
- ***Most of these vulnerabilities can be exploited to just cause a DoS but in some cases they can be exploited to run arbitrary code***

Conclusions

- Total spent time: **10 minutes**
- Skills needed: **none**
- Number of vulnerabilities found: **5 or more**
- Oracle database versions affected: **ALL**
- PoC exploit code provided: **YES**
- Money invested: **$ 0.00**
- Having fun with Oracle software and pointing out Oracle security excellence: **priceless**

Oracle continues showing that it's extremely hard to break!

References

- *Thunder and MAD weblog*

http://blogs.oracle.com/maryanndavidson/

- *Process Explorer*

http://www.sysinternals.com

- *WinObj*

http://www.sysinternals.com

- *Pipeacl Tools*

http://www.bindview.com/Services/razor/Utilities/Windows/pipeacltools1_0.cfm

- *WLSI – Windows Local Shellcode Injection*

http://www.argeniss.com/research/WLSI.zip

References

- *Hacking Windows Internals*

http://www.argeniss.com/research/hackwininter.zip

- *SetSecurityDescriptorDacl() API*

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/setsecuritydescriptordacl.asp

- *SetKernelObjectSecurity() API*

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/setkernelobjectsecurity.asp

Fin

- Questions?
- Thanks
- Contact: cesar>at<argeniss>dot<com

Argeniss – Information Security

http://www.argeniss.com/