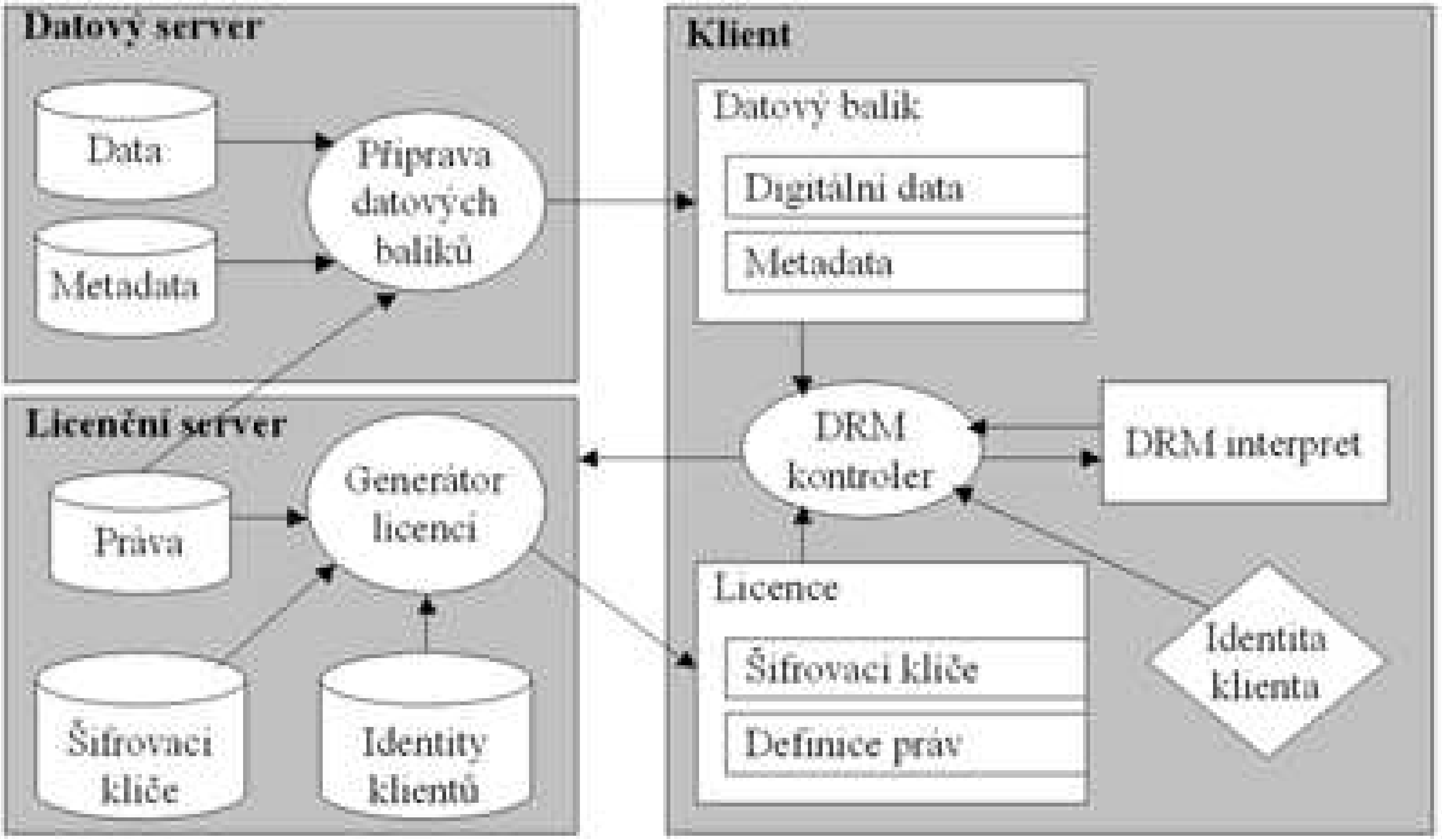
PB173 - Tématický vývoj aplikací v C/C++ (podzim 2012)

Skupina: Aplikovaná kryptografie a bezpečné programování

https://minotaur.fi.muni.cz:8443/pb173_crypto

Petr Švenda, *svenda@fi.muni.cz*
Konzultace: G201, Pondělí 16-16:50

Architecture overview



What we should get at the end

- Separate binaries representing
  - Data server
  - License server
  - DRM controller (+ possible data interpretation)
- Communication between parties
  - realized via file system (read&process&generate file)
  - no network communication is required to be implemented
- Exchanged data are
  - integrity protected&authenticated
  - confidentiality protected (selected parts)
- Suitable key exchange mechanism implemented

Debugging with debugger

Release vs. Debug

- Optimizations applied (compiler-specific settings)
  - gcc –Ox (http://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html)
    - -O0 no optimization (Debug)
    - -Og debug-friendly optimization
    - -O3 heavy optimization
  - msvc /Ox /Oi (http://msdn.microsoft.com/en-us/library/k1ack8f1.aspx)
    - MSVS2010: Project properties→C/C++→optimizations
- Availability of debug information (symbols)
  - gcc –g
    - symbols inside binary
  - msvc /Z7, /Zi
    - symbols in detached file ($projectname.pdb)

Debugger commands (MSVC shortcuts)

- http://www.fi.muni.cz/~xsvenda/VS_debugging.html
- http://cecko.eu/public/qtcreator#debugging
- Insert breakpoint F9
- Run in debug mode F5
- Step over, step into F10, F11
- Watch variable (Autos, Locals, Watch)
  - R-Click → Add watch
  - R-Click on variable (Decimal ↔ Hexadecimal display)
- Change variable (Watch tab)

Debugger commands (2)

- Conditional breakpoint
  - Insert breakpoint & R-Click (MSVS2010), Edit breakpoint (QTC)
  - Condition…, Hit count…, When hit…
  - Filter… (multithreading)
- Data breakpoint (MSVS2010)
  - First run in debug mode
  - Debug → New breakpoint → New data breakpoint
  - &(temp[5]), size 4
  - (must be address of memory by &, not only temp[5])
- Disassembly info
  - MSVS2010: Run debug, R-Click → Go to disassembly
  - QTC: Debug → Operate by instruction

Edit and Continue

- Possibility to continue in debugging session even after code change (fix of partial “bug”)
  - Code is changed, recompiled and debugging continues
- Supported by only some IDE/Debuggers (MSVS)
- Edit and continue
  - Must be enabled before symbols are generated
  - Properties → C/C++ → General → Debug information format

- Possibility to move instruction pointer to ordinary place
  - Move arrow to any line in code, IP is updated
  - Usually moved only few instructions above current IP (same function)
  - Be careful: only IP is updated, not the stack etc.

Debugger commands (3)

- Debugging of Release binary
  - possible, similar to Debug mode
  - but asm code with optimizations
  - with or without symbols
  - WinDbg
- Debugging of running process
  - MSVS2010: Debug → Attach to process
- Reverse engineering
  - OllyDbg, http://www.ollydbg.de/
  - IDAPro http://www.hex-rays.com/products/ida/index.shtml

Additional reading

- http://eli.thegreenplace.net/programs-and-code/how-debuggers-work/

Practical assignment

- Finish implementation of data packets
  - data packet from Data server
    - encryption/decryption of data
    - integrity and authentication
  - license from License server
    - XML packet with assigned rights
    - protected key for data packet
- DRM controller
  - access key in license (controller’s key in PKCS#11 token)
  - decrypt and verify data packet
  - enforce license rights and output data from data packet

Practical assignment (2)

- Make dcoumentation of your design choices
  - A4 page with architecture overview
  - format of your data packets (text & graphical description)
  - complete and generate Doxygen documentation
- Write missing unit tests as usual
- Use debugger to catch the problems ☺

Next assignment will be code review of your code by other groups!