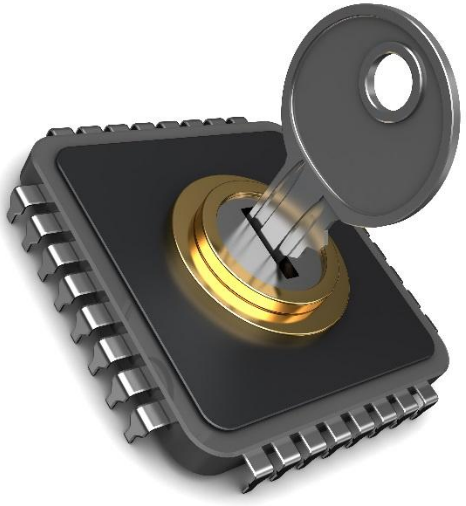
Introduction to Information Security

Tutorial C Security



Winter 2015/2016

TU Graz

Tutor David Bidner, www.iaik.tugraz.at

C Security - Assignment

Code Analysis

- Small C snippets with unsafe functions
- Find possible overflows/errors
- Exploit it

Reverse Engineering

- Analyse given binaries with gdb/radare2
- Understand what it does/behaviour on input etc.
- Create code which has the same behaviour

C Security - Tools

gdb

- GNU debugger
- Analyse runtime, find variable addresses
- https://www.gnu.org/software/gdb/ (apt-get install gdb)

radare2

- reverse engineering framework
- Disassemble and analyse binaries
- https://github.com/radare/radare2

C Security - Tools

gdb-peda

- Helpful gdb plugin
- Improves gdb usage
- https://github.com/longld/peda

pwntools

- CTF/exploit framework
- Makes it easier to write exploits
- https://github.com/Gallopsled/pwntools

C Security - Get started

Read a lot and try stuff out

- http://insecure.org/stf/smashstack.html
- http://www.tenouk.com/Bufferoverflowc/stackbasedbufferoverflow.html
- https://github.com/radare/radare2/wiki/Usage-Examples

Needed commands

- `gcc -m32 -g -ggdb -O0 -fno-stack-protector <source.c> -o <binary name>`
- `echo 0 > /proc/sys/kernel/randomize_va_space`

C Security - Spec part

Insecure C code with obvious stack overflow

- Get in touch with your tools, learn to handle them
- Exploit it by executing system(‘/bin/sh’)

Small binary

- Try to reverse the code of the given binary (language does not matter)
- Use radare2/gdb to get functionality in detail
- Do a short writeup about your steps

C Security - Spec part

You get now:

- ZIP file with the 2 challenges

You hand in:

- Short writeup about your steps to achieve an exploit
- Short writeup about your reverse engineering steps
- Exploit and Code of reversed binary

C Security - Further Challenges

Similar tasks as in the spec, but maybe harder

- Optimized code to reverse
- Differences in 32/64 bit
- Stack protection/ASLR

C Security - Grading

The Grade will depend on

- How many challenges were solved
- How understandable are the writeups
- How much work was done by _you_

C Security - Next Steps

After the spec talk:

- you get further challenges
- we’ll have another tutorial discussing “advanced” security

C Security - Main Assignment

Should include:

- Detailed write ups for all your exploits/reverse steps
- Shell logs of your exploits in working state
- Reproducible exploits

C Security - Questions

If you have any questions,

- ask them now.
- ask them in the newsgroup (general questions)
- send an email (questions for possible solutions etc.)