

Courses as Code: The Aquinas Learning System

W. Michael Petullo

University of Wisconsin–La Crosse

As of August 6, 2022



```
#include <stdio.h>

int main(void)
{
    printf(" Hello , _world!\n" );
}
```



git

L^AT_EX

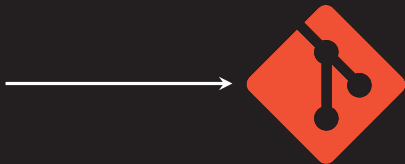
Makefiles

Bourne shell scripts



git

Lesson plans
Slides
Exercises
Virtual machine definitions



git

Everything-as-code approach



Motivation 1: Everything-as-Code

What if we built an online learning system around the everything-as-code approach?

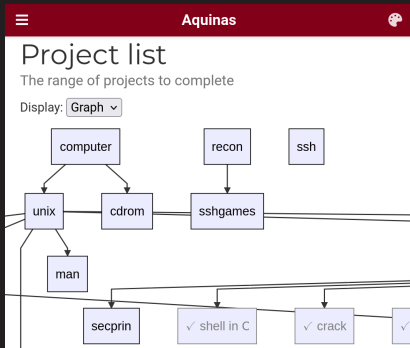


Motivation 2: Capture-the-Flag Format

Challenges of gamified education in computer science and security.

- ▶ Infrastructure to build
- ▶ Exercises can be fragile
- ▶ Limited-time events
- ▶ Competition often limits formal lessons

View of Samantha the Student



```
Scratch ~ git clone samantha@git.aquinas.dev:/home/samantha/ropC
Cloning into 'ropC'...
Scratch ~ cd ropC/
Scratch ~ vi rop.c
Scratch ~ git add rop.c
Scratch ~ git commit -s -m "Update ROP chain"
[master 9c8e73b] Update ROP chain
1 file changed, 1 insertion(+), 1 deletion(-)
Scratch ~ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: 309fb085-c602-4dad-88b0-2f61f4d26561
To git.aquinas.dev:/home/samantha/ropC
6f69c90..9c8e73b master -> master
Scratch ~
```

Assignment

Setup

Clone the Git repository at:

```
samantha@git.aquinas.dev:/mnt/xvdb/samantha/ropC
```

Specification

Write a program that takes a hostname and port as command-line arguments, connects using TCP/IPv4 to the given port on the given hostname, writes data that causes the remote program to print its secret value, and prints this value to standard output. A file named `flag` that exists in the working directory of the service as it executes and contains the secret value.

Submission

Complete this project using the C programming language. Your entire program must be contained in a single file named `rop.c`, and this file must exist at the root of your Git repository. Commit your work and push it back to `git.aquinas.dev` for grading.

Last submission

```
failed building ropC: rop.c:62:1: error: unknown type name 'ssize_t'
62 | ssize_t recvuntil(int sockfd, const char c, char *buf, size_t siz
| ^~~~~~
```

Records

Commit	Time	Result
19dbcdf6d164f4c3c76f703074bd1436a7fa62fb	2020/05/23 19:05:23	FAIL

View of Tom the Teacher

```
\section*{Lesson}
```

The AMD64 architecture arrived with a new feature: a no-execute bit that causes the processors' memory management unit to deny the execution of certain regions of memory designated by the operating system. The idea is that if the stack were not executable, then an attacker could not introduce instructions for execution by way of a buffer overflow. This does indeed prevent the classic Aleph One attack~\cite{one96smashingStack}

```
{
  "name": "rop",
  "summary": "Attack that makes use of ROP",
  "languages": [
    "C",
    "Go",
    "Python"
  ],
  "tags": {
    "Exploitation": true
  },
  "prerequisites": [
    "nop"
  ],
  "checks": [
    {
      "kind": "basic",
      "parameters": {
        "command": "./rop",
        "stdin": null,
        "stdout": "[REDACTED]",
        "stderr": null,
        "exitCode": 0
      }
    }
  ],
  "services": [
    {
      "source": "service-rop.c",
      "port": 1032,
      "publish_binary": true
    }
  ],
  "service_files": [
    "flag"
  ]
}
```




- ▶ US Army's Cyber Solutions Development Detachment
- ▶ UW-L CS120, Software Design I
- ▶ UW-L CS456, Secure Software Development
- ▶ UW-L CS455, Fundamentals of Information Security
- ▶ UW-L CS410, Open Source Development
- ▶ UW-L CS356, Software Exploitation
- ▶ UW-L Camp Quantum, lessons for high school students



Conclusions: See the Paper for Details

Summaries of student surveys

- ▶ Non-CS majors can learn Git in their first programming course—and some of them like it!
- ▶ Even non-CS majors reported favorably about using the command line!
- ▶ Students prefer immediate feedback!

Post-evaluation improvements

The most common suggestion from students was to improve the feedback provided in response to a failed submission. This led to “compare-type” checks.

<https://www.flyn.org/projects/Aquinas/>

<https://www.aquinas.dev/>