Software Security and Automated Reasoning (SS & AR)

Lucas Cordeiro and Renate Schmidt
lucas.cordeiro@manchester.ac.uk
renate.schmidt@manchester.ac.uk
70 percent of all security bugs are memory safety issues

- “The majority of vulnerabilities are caused by developers inadvertently inserting memory corruption bugs into their C and C++ code. As Microsoft increases its code base and uses more Open Source Software in its code, this problem isn’t getting better, it’s getting worse (2019).”

Security Vulnerabilities

```c
int getPassword() {
    char buf[4];
    gets(buf);
    return strcmp(buf, "SMT");
}
```

```c
void main(){
    int x=getPassword();
    if(x){
        printf("Access Denied\n");
        exit(0);
    }
    printf("Access Granted\n");
}
```

- What happens if the user enters “SMT”?
- On a Linux x64 platform running GCC 4.8.2, an input consisting of 24 arbitrary characters followed by $j$, `<ctrl-f>`, and `@`, will bypass the “Access Denied” message.
- A longer input will run over into other parts of the computer memory.

Exciting research projects concerning SS & AR:

- EnnCore
- SCorCH
- ELEGANT
Automated Reasoning

- It is fundamental to CS and AI, and provides powerful tools for
  - **Ensuring correct functioning of complex systems** (software, security protocols, hardware, product configuration, ...)
  - Microsoft, Intel, NASA, Mercedes, Toyota, Airbus
  - **AI in Health:** underpins medical terminological services to enable consistent data capture in EHRs, data sharing, smart data analysis across the NHS
  - Researchers at Manchester have teamed up with SNOMED Intl to develop bespoke approach for content extraction and sharing in the medical ontology SNOMED CT
  - **Many other difficult problems:** professional sports scheduling, planning, optimisation, ...
Automated Reasoning

- Is a truly international subject area that has attracted outstanding scholars
- Prof Wu Wenjun (吴文俊), Herbrand Award Winner 1997
- Prof Andre Voronkov, Herbrand Award Winner 2015
Software Security and Automated Reasoning

Our theme will embrace various techniques and tools that exist to **prevent and detect software flaws**, which are typically hard to be manually found, including **modelling, code reviews, fuzzing, static and dynamic code analyses, code tainting, and automated reasoning**.
What will you learn?

- Basics: modelling of knowledge, propositional/first-order logic, …
- Approaches underpinning modern AR&V systems
- Techniques to achieve efficiency: backjumping, orderings, redundancy elimination...
- Solving a variety of reasoning problems, incl. verification and security protocol analysis

You will understand some of the most powerful and efficient automated reasoning methods, and how and why they work
COMP63342 - Software Security

• What will you learn?
  – Approaches to formally build verified trustworthy software systems to ensure confidentiality, integrity and availability
  – Understand risk assessment to guide software developers and provide rules for secure coding to avoid exploitable vulnerabilities
  – Detection of software vulnerabilities using static and dynamic analysis
  – Use verification techniques to reason about the AI system’s safety and security
Assessment
(COMP60332 and COMP63342)

• How will you learn?
  – Lectures, workshops, tutorials, labs/practicals

• COMP63342:
  – 70% Coursework
    • Lab exercises = 40%
    • Blackboard Quizes = 10%
    • Seminars = 20%
  – 30% Exam
    • Format: 2 hours, 3 questions, all the material

• COMP60332:
  – Weekly coursework 5 x 10%
  – Written exam 50%
Some advice on choosing themes

- The **Software Security & Automated Reasoning** theme can be combined with any other theme
- Has no prerequisites, no pre/co-requisite to any theme
- It goes well with all themes
  - *Cyber Security, Software Engineering*
  - *Data on the Web*
  - *Data Engineering & Systems Governance, Learning from Data*
- Can be chosen in all pathways; core in the Computer Security pathway

**Questions?** Please email:

lucas.cordeiro@manchester.ac.uk,
renate.schmidt@manchester.ac.uk