

# Secure Programming Lecture 1: Introduction

David Aspinall

Informatics @ Edinburgh

# Orientation

- ▶ This course is **Secure Programming**.
- ▶ More generally, it is about **Software Security**.
- ▶ Aimed at Informatics **MSc and 4th/5th year**
- ▶ Primarily: those anticipating a career in software
  - ▶ **programming**: architects, developers, testers, ...
  - ▶ **security**: pentesters, malware/reverse engineers
  - ▶ **researchers**: verification, compilers, languages, ...
- ▶ It is taught by **David Aspinall**.

Public home page: <https://opencourse.inf.ed.ac.uk/sp>

Lecture recordings are available via [Learn](#) (UoE only).

# Outline

**Motivations**

Course syllabus

Software security overview

Practicalities

Structure of course

Summary

## Ubiquitous software is broken (2014)



## Ubiquitous hardware is flawed (2018)



## Old systems break operations (2017)

# NHS cyber-attack: GPs and hospitals hit by ransomware

© 13 May 2017



 Share



# Attacks can cause physical damage (2014)

**BBC** Sign in News Sport Weather iPlayer TV Radio

**NEWS**

Home UK World Business Politics Tech Science Health Education Entertainment

Technology

## Hack attack causes 'massive damage' at steel works

© 22 December 2014 | Technology



APP

The hack attack led to failures in plant equipment and forced the fast shut down of a furnace

The image shows a cross-section of a glowing, molten furnace interior, likely a blast furnace, with intense heat and structural damage. The furnace is surrounded by dark, industrial equipment. The text below the image states: 'The hack attack led to failures in plant equipment and forced the fast shut down of a furnace'.

# Nobody can keep online records safe (2015)

Forbes / Tech

The Little Black Book of Billionaire Secrets

JUN 11, 2015 @ 09:12 PM 15,154 VIEWS

## Federal Union Says OPM Data Breach Hit Every Single Federal Employee





# Known good practice ignored (2015)

NEWS

## TalkTalk discloses possible breach, admits some data not encrypted



A woman walks past a company logo outside a TalkTalk building in London, Britain October 23, 2015. Credit: [BEUTERS/Stefan Wermuth](#)

### MORE LIKE THIS



TalkTalk hit by data breach and ransom demand



Police arrest 15-year-old in TalkTalk hack



UK police arrest third person in TalkTalk breach investigation

on IDG Answers

How is data stored and accessed in the cloud?

# IoT easily raises a DDoS botnet army (2016)



# Rogue updates via supply chains (2020)

## **SUNBURST SolarWinds Orion IT management platform security advisory**

by Deft | Dec 16, 2020 | Security



## Why does this happen?

Ostensibly, **many security failures are due to software vulnerabilities**. Are they inevitable?

Many surrounding questions. Can we:

- ▶ *find* vulnerabilities (before attacks)?
- ▶ *detect exploits* in-the-wild?
- ▶ *repair* vulnerabilities (routinely/automatically)?
- ▶ *program better* to avoid vulnerabilities?
- ▶ *measure risk* associated with software?
- ▶ *design* or *verify* to prevent them?
- ▶ *develop new technology* for all the above?

Questions beyond the technical, too. Can we:

- ▶ *insure* against cyber incidents?
- ▶ *regulate* for better security?

# Outline

Motivations

**Course syllabus**

Software security overview

Practicalities

Structure of course

Summary

# What is this course about?

Building software that's more secure

- ▶ finding flaws in existing software
- ▶ avoiding flaws in new software (design and code)
- ▶ techniques, tools and understanding to do this

The *infrastructure* around secure software:

- ▶ language, libraries, run-time; other programs
- ▶ data storage, distribution, protocols and APIs
- ▶ development and deployment methods

And first of all, setting *policies* for security

- ▶ what should be protected
- ▶ who/what is trusted
- ▶ risk assessment: cost of defences.

# Target audience

- ▶ Aimed at MSc, 4th/5th year UGs
- ▶ Have passed *Computer Security* or similar
  - ▶ Basic notions, crypto, protocols
- ▶ **Programming practice**
  - ▶ should be confident in programming
  - ▶ necessarily will use a range of languages
  - ▶ ... including assembler, C, Java
  - ▶ but don't have to be "master hacker"
  - ▶ grounded in software engineering
- ▶ **Programming theory**
  - ▶ interest in PL concepts and design
  - ▶ knowledge of *compilers* useful
  - ▶ also software engineering, esp, *testing*
  - ▶ theory courses helpful, *semantics*

# Why should you take this course?

Want to work in the **cyber security industry**?

- ▶ security appraisal, system and code reviewing
- ▶ pen-testing, ethical hacking
- ▶ malware analysis, reverse engineering
- ▶ operations and response (SOCs)
- ▶ cyber defence, attack, espionage
- ▶ innovation: found a cyber start-up

Want to work in **security research**?

- ▶ academic (conceptual advances, fixing, breaking)
- ▶ commercial (breaking, fixing, defending)

(Hopefully): you think it's **fun and interesting!**



## Why should you *not* take this course?

- ▶ None of the previous points apply
- ▶ You don't have the right background (see next slide)
- ▶ Perhaps: you know (almost all of) it already

Warning: We try to keep the course up-to-date so it is sometimes “rough at the edges”.

# Expected background

Please see [Guide to Background Needed](#) on course homepage.

1. Security properties: C, I, A, non-repudiation, privacy

# Expected background

Please see [Guide to Background Needed](#) on course homepage.

1. Security properties: C, I, A, non-repudiation, privacy
2. **Attacks** against each of these

# Expected background

Please see [Guide to Background Needed](#) on course homepage.

1. Security properties: C, I, A, non-repudiation, privacy
2. **Attacks** against each of these
3. **Defences**: Au x 2, access control, crypto, networks

# Expected background

Please see [Guide to Background Needed](#) on course homepage.

1. Security properties: C, I, A, non-repudiation, privacy
2. **Attacks** against each of these
3. **Defences**: Au x 2, access control, crypto, networks
4. **Coding** skills and problem solving

# Expected background

Please see [Guide to Background Needed](#) on course homepage.

1. Security properties: C, I, A, non-repudiation, privacy
2. **Attacks** against each of these
3. **Defences:** Au x 2, access control, crypto, networks
4. **Coding** skills and problem solving
5. **Practicals: command line Linux**

## Learning outcomes

1. Know how to respond to (software) security alerts.
2. Identify possible security programming errors when conducting code reviews.
3. Be able to define a methodology for security testing and use appropriate tools in its implementation.
4. Apply new security-enhanced programming models and tools which help ensure security goals, e.g., with access control, information flow tracking, protocol implementation, or atomicity enforcement.

# Outline

Motivations

Course syllabus

**Software security overview**

Practicalities

Structure of course

Summary



## Safety versus security

**Safety** is concerned with ensuring bad things don't happen *accidentally*. For example, aeroplanes don't fall out of the sky because maintenance checks are forgotten.

**Security** is concerned with with ensuring that bad things don't happen because of *malicious actions by others*. For example, terrorists cannot drive bombs into [airport departure halls](#).

The distinction is sometimes blurred, and the two interact in intriguing ways. (Q. why?)

# The challenge of software security

Software artefacts are among the most complex built.

- ▶ **Design flaws** are likely

# The challenge of software security

Software artefacts are among the most complex built.

- ▶ **Design flaws** are likely
- ▶ **Bugs** seem inevitable

# The challenge of software security

Software artefacts are among the most complex built.

- ▶ **Design flaws** are likely
- ▶ **Bugs** seem inevitable

# The challenge of software security

Software artefacts are among the most complex built.

- ▶ **Design flaws** are likely
- ▶ **Bugs** seem inevitable

Flaws and bugs lead to *vulnerabilities* which are exploited by *attackers*.

Often to learn secrets, obtain money. But many other reasons: a security risk assessment for a system should consider different attackers and their motives.

Cost estimates are difficult

# **THE COST OF CYBER CRIME.**

**A DETICA REPORT IN PARTNERSHIP  
WITH THE OFFICE OF CYBER  
SECURITY AND INFORMATION  
ASSURANCE IN THE CABINET OFFICE.**

But it's agreed they're increasing. . .

Crystal Ball

## Your personal files are encrypted!



Your important files **encrypted** produced on this computer: photos, videos, documents, etc. **1888** is a complete list of encrypted files, and you can personally verify this.

Encryption was produced using a **unique** public key **016-2048** generated for this computer. To decrypt files you need to obtain the **private key**.

The **single copy** of the private key, which will allow you to decrypt the files, located on a secret server on the Internet, the server will **destroy** the key after a time specified in the window. After that, **nobody and never will be able** to restore files.

To **obtain** the private key for this computer, which will automatically decrypt files, you need to pay **300 USD / 300 EUR / other amount in another currency**.

Click **Next** to select the method of payment and the currency.

**Any attempt to remove or damage this software will lead to the immediate destruction of the private key by server.**

Private key will be destroyed on  
8/26/2013  
3:54 PM

Time left  
**71 : 59 : 52**

016-2048-0000

# Cyber warfare is real





# Privacy is being eroded



## A privacy reminder from Google

Scroll down and click "I agree" when you're ready to continue to S explore other options on this page.

To be consistent with data protection laws, we're asking you to ta to review key points of Google's Privacy Policy. This isn't about a c we've made - it's just a chance to review some key points.

### Data we process when you use Google

- When you search for a restaurant on Google Maps or watch a video on Y example, we process information about that activity - including informa video you watched, device IDs, IP addresses, cookie data and location.
- We also process the kind of information described above when you use that use Google services like ads, Analytics and the YouTube video play



# But maybe there is hope...



What does GDPR mean for you? | The Register  
the-register.com



What is GDPR? Everything you need to know  
netpage.com



GDPR and its implications on data  
dataintegrity.com



GDPR Compliance for WordPress  
wpdata.com



GDPR 1 - Regain Control - Medium  
medium.com



Is Your Website GDPR Compliant? | GDPR  
dataintegrity.com



GDPR can help with GDPR compliance  
dataintegrity.com



What GDPR Means for U.S. Brands  
netpage.com



Data Protection Regulation  
netpage.com



GDPR / WordPress.org  
wordpress.org



General Data Protection Regulation  
dataintegrity.com



GDPR - Basic info - Overview  
netpage.com



General Data Protection Regulation  
netpage.com



Legal Announcement: General Data  
netpage.com



Data Protection Regulation  
netpage.com



GDPR Compliance - Protecting Customer  
netpage.com



What is GDPR and how will it affect you?  
netpage.com



GDPR - British Privacy  
netpage.com



Download to GDPR Compliance - Welcome  
netpage.com



GDPR and email marketing: what you need to know  
netpage.com



What is GDPR? | HelpSystems  
helpsystems.com



GDPR: Where are we now? | CSD Online  
netpage.com



One Month to GDPR. Are you ready?  
netpage.com



GDPR: What is it and how right is it?  
netpage.com



What is GDPR Compliance? Your Questions  
netpage.com



GDPR Compliance - It's Time to Prepare  
netpage.com



Release GDPR Resource Center - What, Why, How  
netpage.com



GDPR Compliance is an Opportunity for  
netpage.com



Are you GDPR compliant? - Netpage.com  
netpage.com



Does GDPR apply to Canada?  
netpage.com

# Why isn't software security better?

**What if Microsoft breaches its warranty?** If Microsoft breaches its limited warranty, your only remedy is the repair or replacement of the software. We also have the option to refund to you the price you paid for the software (if any) instead of repairing or replacing it. Prior to refund, **you must uninstall the software and return it to Microsoft, with proof of purchase.**

**What if Microsoft breaches any part of this agreement?** If you have any basis for recovering damages from Microsoft, you can recover only direct damages up to the amount that you paid for the software (or up to \$50 USD if you acquired the software for no charge). **You may not recover any other damages, including consequential, lost profits, special, indirect, or incidental damages.** The damage exclusions and limitations in this agreement apply even if repair, replacement or a refund for the software does not fully compensate you for any losses or if Microsoft knew or should have known about the possibility of the damages. Some states and countries do not allow the exclusion or limitation of incidental, consequential, or other damages, so those limitations or exclusions may not apply to you. **If your local law allows you to recover other damages from Microsoft even though this agreement does not, you cannot recover more than you paid for the software (or up to \$50 USD if you acquired the software for no charge.)**

# Why (else) isn't software security better?

- ▶ Asymmetry: attackers have the advantage
  - ▶ just need to find one viable attack route
  - ▶ defenders have to anticipate all
- ▶ Attackers focus on weakest links:
  - ▶ since 1990s, network defences vastly improved
  - ▶ rise of insider threats
- ▶ Current *penetrate-and-patch* approach is broken
  - ▶ understandable by managers (“show me the problem!”)
  - ▶ but no substitute for secure design

# What's the outlook?

## New frontiers:

- ▶ PCs in decline, but connected devices increasing
- ▶ Mobile a target point (convergence, mobility)
- ▶ Internet of Things: repeating same mistakes!
- ▶ Cloud: XaaS (X=software,storage,platforms,. . .)
- ▶ Cyber resilience: speedy, automatic recovery
- ▶ Data sharing and its limits: **privacy**

## Emerging new solutions:

- ▶ *Build Security In, Secure By Design*
- ▶ Defensive technologies continuing to evolve
- ▶ New cryptographic, verification techniques
- ▶ Old ideas re-appear: MLS, containment, isolation
- ▶ Updates: automatic, pushed patching

# Outline

Motivations

Course syllabus

Software security overview

**Practicalities**

Structure of course

Summary

# Delivery and assessment

We will have

- ▶ **16-18** lectures covering core course topics

Lecture **slides** will be made available in several formats.

They [have numerous embedded links](#) to useful resources (the links are more noticeable in the online versions).

Lecture **recordings** will be available, systems permitting. These are intended as a **backup** for attending in person.

# Delivery and assessment

We will have

- ▶ **16-18** lectures covering core course topics
- ▶ **4** lab sessions

Lecture **slides** will be made available in several formats.

They [have numerous embedded links](#) to useful resources (the links are more noticeable in the online versions).

Lecture **recordings** will be available, systems permitting. These are intended as a **backup** for attending in person.



# Delivery and assessment

We will have

- ▶ **16-18** lectures covering core course topics
- ▶ **4** lab sessions
- ▶ **1** coursework contributing 30% of final mark

Lecture **slides** will be made available in several formats.

They [have numerous embedded links](#) to useful resources (the links are more noticeable in the online versions).

Lecture **recordings** will be available, systems permitting. These are intended as a **backup** for attending in person.

# Delivery and assessment

We will have

- ▶ **16-18** lectures covering core course topics
- ▶ **4** lab sessions
- ▶ **1** coursework contributing 30% of final mark
- ▶ **1** written exam contributing 70% of final mark

Lecture **slides** will be made available in several formats.

They [have numerous embedded links](#) to useful resources (the links are more noticeable in the online versions).

Lecture **recordings** will be available, systems permitting. These are intended as a **backup** for attending in person.

## Lab sessions

Four core 3hr lab sessions (see [home page](#)):

- ▶ Weeks 3,5,7,9: **Wed 1pm-4pm** Appleton Tower 6.06 [TBC]

Each session will examine software vulnerabilities: why they exist, how they can be *discovered*, *exploited*, and *repaired*.

**Working together is encouraged.** We want to foster a supportive learning environment. Students who have prior knowledge or expertise are especially welcome.

We use the [SEED Labs](#) developed at Syracuse University, New York. They are free to access for your own use.

## Formative feedback during Labs

One reason to introduce labs in this course is to allow us to give face-to-face discussion and feedback on your learning.

Lab sessions will be run by me together with the course TA **Robert Flood**.

# Coursework

The coursework will follow a similar pattern to the lab exercises: *discover, explain, exploit* then *repair*.

1. as usual: **your work should be your own**
2. **no publication**, please do not publish solutions even after the deadline

(at least two reasons for last point).

The coursework is split into stages, with a Stage 1 (informal) deadline in Week 6 and final deadline in Week 9.

## An ethical point (reminder)

***Nothing in this course is intended as incitement to crack into running systems!***

- ▶ Breaking into systems to “demonstrate” security problems at best causes a headache to overworked sysadmins, at worst compromises systems for many users and could lead to **prosecution**
- ▶ If you spot a security hole in a running system, **don't exploit it**, instead contact the relevant administrators or developers confidentially.
- ▶ To experiment with security holes, play with your own machine, or better, your **own private network of machines**.

# Communications

- ▶ Fast moving subject, evolving course:
  - ▶ **honest, constructive feedback is very welcome**
- ▶ As with any course, I welcome
  - ▶ **questions after lectures**
  - ▶ **questions on Piazza** (preferred)
  - ▶ **questions by email**

# Exam

Will follow the format:

- ▶ Choose 2 questions to answer from 3
- ▶ Two hours allowed

Towards the end of the course I will provide:

- ▶ a list of topics and concepts that may be examined
- ▶ a hint about the format of the questions

Some guidance will be published on the course web pages.



# Outline

Motivations

Course syllabus

Software security overview

Practicalities

**Structure of course**

Summary

# Dimensions: practice and theory

## Practice

- ▶ Programming securely, identifying security issues
- ▶ Mistakes in language, APIs, crypto, comms. . .
- ▶ Ultimately: *detailed, highly specific* knowledge

## Theory

- ▶ Understand reasons for failure, ways to mitigate
- ▶ Understand advanced techniques, automated tools
- ▶ In general: *transferable* concepts and methods.

This is not really a “vocational” course. I hope it will give you the foundation to allow you to *rapidly develop* detailed specific knowledge needed later. There are a number of certification schemes for building practical knowledge.

# Overview of topics

General organisation:

1. **Threats**
2. **Vulnerabilities**
3. **Defences**
4. **Processes**
5. **Emerging Methods**

We'll look at details under each of these headings (in various orders).

# Outline

Motivations

Course syllabus

Software security overview

Practicalities

Structure of course

**Summary**

# Review questions

## **Safety versus Security**

- ▶ Explain the difference between these two, and why ensuring security may be harder.

## **Security flaws and their impact on society.**

- ▶ Explain some recent secure programming flaws that made the news and explain what the underlying problems were.
- ▶ Discuss the fundamental reasons that software security fails and the wider questions around *cyber security*.

## References and reading

Some slides contain links which you can [click on](#) to find referenced or connected material.

There is no single recommended course textbook, although a few books will be mentioned and are on the resource list.