CITS3007 Secure Coding
Introduction

Unit coordinator: Arran Stewart

Introduction

Outline

- Goals
  - What is this course about?
  - What do we cover, and why?
- Admin
  - Teaching staff
  - Unit website & announcements
  - Teaching activities
  - Assessment & feedback
  - Prerequisites
- Assessment tips
- C programming environment
- Security introduction

What is this unit about?

This unit is about **software security**, which is part of the larger field of **information security**.

We look at weaknesses (vulnerabilities) that can be present in software systems, and can lead to security being compromised.

We ask: How can we build software that is more secure?

And we look at two main approaches:

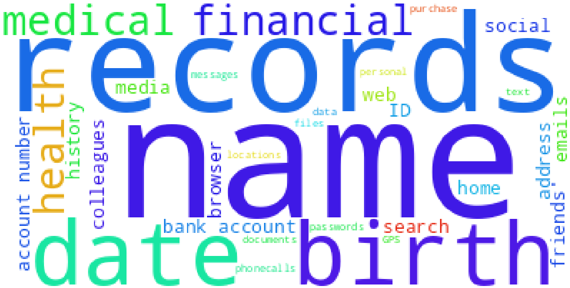
- finding vulnerabilities in existing software
- avoiding vulnerabilities in new software (whether at the design or the implementation phases)

Related areas

- The job of administering and operating software once it *is* developed is part of **system administration** or **security administration**.
- We look at security testing – the process of attempting to find vulnerabilities in software – but it is a large area, and we cannot cover it all.
  - In particular, **penetration testing** – simulating attacks on a computer system – is the subject of a separate unit (CITS3006).
- We touch on ways that computer hardware can assist in developing secure software – but **secure hardware design** is its own subject.
  - (UWA does not have a unit dedicated to it, although some universities such as MIT University do.)

Why care?

Everyone has items of personal information stored in many different computer systems . . .



How happy are you for others to access this information? Does it matter who is doing the accessing?

Cost of cyber-crime

Estimates of the annual cost of cyber crime to the Australian economy range from $33 billion[1] to $42 billion[2].

Which is a lot.

[1] Australian Cyber Security Centre, 2021.
[2] UNSW Canberra, 2021.
[3] Deloitte Access Economics, 2021.

Cost of cyber-crime

Estimates of the annual cost of cyber crime to the Australian economy range from $33 billion[1] to $42 billion[2].

Which is a lot.

On the other hand, natural disasters are *also* estimated to cost that much to the Australian economy – about $38 billion per year[3] – so we should probably be equally worried about both cyber crime and the environment.

And both amount to about 2% of Australia's annual GDP (Gross Domestic Product) of $US 1.3 trillion.

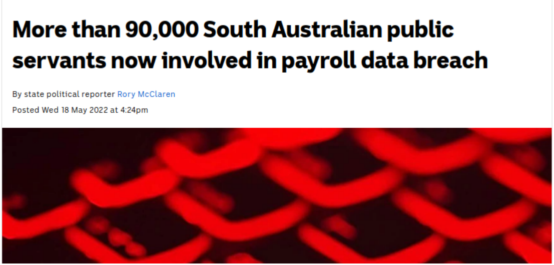
[1] Australian Cyber Security Centre, 2021.

[2] UNSW Canberra, 2021.

[3] Deloitte Access Economics, 2021.

Challenges

Large-scale computer data breaches have been occurring since at least 1984,[4] and organisations still seem unable to adequately protect users' data:[5]



More than 90,000 South Australian public servants now involved in payroll data breach

By state political reporter Rory McClaren

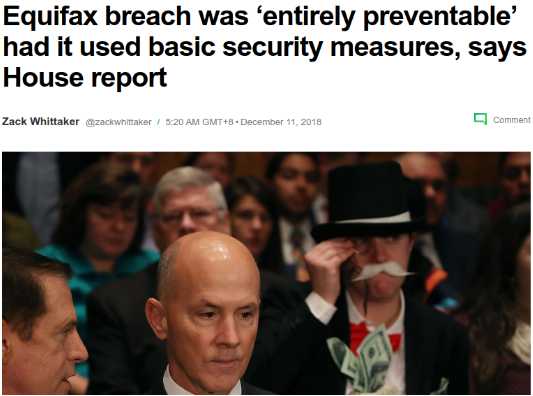
Posted Wed 18 May 2022 at 4:24pm

[4] See David Kalat, "The First Major Data Breach: 1984" (2020).

[5] See Rory McClaren, "More than 90,000 South Australian public servants now involved in payroll data breach" (ABC News, May 2022).

Challenges

Often, we know basic bad practices to avoid, and good ones to adopt – but organizations still ignore these very basic security practices.[6]



Equifax breach was ‘entirely preventable’ had it used basic security measures, says House report

Zack Whittaker @zackwhittaker / 5:20 AM GMT+8 • December 11, 2018

[6] See Zack Whittaker, “Equifax breach was ‘entirely preventable’ had it used basic security measures, says House report” (TechCrunch, 2018)

Preventing computer security failures

Many (but not all) security failures can be prevented through improved coding practices:

- validating input received from untrusted sources
- sanitizing or escaping output
- requiring authentication for all resources not specifically intended to be public
- not disclosing sensitive information in error responses/pages
- implementing the "Principle of Least Privilege" – granting users, systems or programs only the access they need in order to perform their tasks
- encrypting the transmission of all sensitive information
- avoiding insecure uses of memory

Admin

Teaching staff

Unit Coordinator

Arran Stewart
Email: cits3007-pmc@uwa.edu.au
Phone: +61 8 6488 1945
Office: Rm G.08 CSSE Building
Consultation: 4–5pm Wednesdays, or email for an appointment.

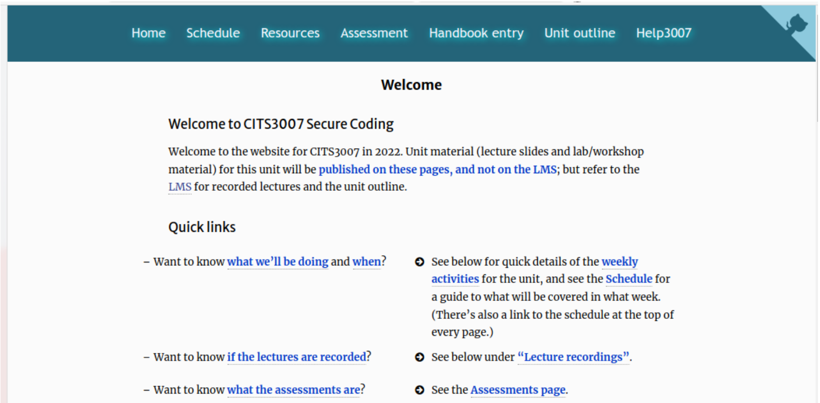
Lab facilitators

- Carl Alvares
- Nicodemus Ong
- Santiago Rentería

Unit website

Nearly all content for the unit will be available from the unit website, which is hosted on GitHub.

The easiest way to find it is to search on Bing or Google for “CITS3007 github”, or to bookmark https://github.com/cits3007.



Welcome

Welcome to CITS3007 Secure Coding

Welcome to the website for CITS3007 in 2022. Unit material (lecture slides and lab/workshop material) for this unit will be published on these pages, and not on the LMS; but refer to the LMS for recorded lectures and the unit outline.

Quick links

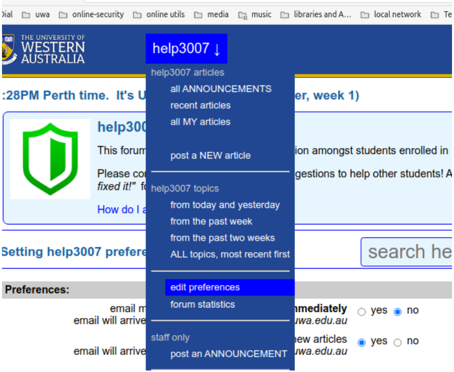
– Want to know what we’ll be doing and when?	➲ See below for quick details of the weekly activities for the unit, and see the Schedule for a guide to what will be covered in what week. (There’s also a link to the schedule at the top of every page.)
– Want to know if the lectures are recorded?	➲ See below under “Lecture recordings”.
– Want to know what the assessments are?	➲ See the Assessments page.

Unit website

You **won't** need to visit the Blackboard LMS to obtain teaching materials – you'll only need to visit it to access lecture recordings.

Announcements

- Announcements will be made in lectures, and on the unit help forum, `help3007`.
- It's important to check the forum regularly – at least twice a week.
- If you log in and visit the forum site, you can set it to alert you via email when new postings are made.



:28PM Perth time. It's U … er, week 1)

help300

This forum … ion amongst students enrolled in

Please co … gestions to help other students! A

fixed it!" f

How do I a

Setting help3007 prefere

Preferences:

email m … **mediately** ○ yes ● no
email will arrive … *uwa.edu.au*

… ew articles ● yes ○ no
email will arrive … *uwa.edu.au*

Problems

Who should I contact if I have an issue?

- For most matters – the unit coordinator (UC), Arran
  - If it's a problem other students are likely to have, it's suggested you post to Help3007 so other students can benefit from the answer.
  - If you require personal communication with the UC, feel free to email me on cits3007-pmc@uwa.edu.au.
- In labs – feel free to ask the lab facilitators about any of the teaching and learning materials presented in labs or lectures.

Unit contact hours – lectures

- You should attend one lecture (1 hour 50 mins) per week – I recommend attending in person (so you can ask and answer questions), but if you are unable to attend, you can also watch the recorded lecture.

  Recorded lectures are available via the university's LMS, at https://lms.uwa.edu.au/.

- You'll get more out of lectures if you read the lecture slides (and work through the recommended reading) *before* the lecture.

  Then the lecture time can be spent clarifying your understanding of the material, rather than me going over content that you already have.

- I therefore aim to have slides for the week posted on the unit website early in the week.

Unit contact hours – labs

- You should attend one lab (1 hour 50 mins) each week, starting in week *two*.

  If there is room available for you, you are welcome to attend other lab sessions as well. (See the website to find the times for labs other than the one you're allocated to.)
- In the labs, we will work through practical exercises related to the unit material.
- You will need a laptop for the labs – we'll be experimenting with software and configurations that aren't available on University computers.

Lecture slides and lab worksheets

- Lecture slides and lab worksheets will go up on the website as the semester progresses.

Non-timetabled hours

A six-point unit is deemed to be equivalent to one quarter of a full-time workload, so you are expected to commit 10–12 hours per week to the unit, averaged over the entire semester.

Outside of the contact hours (3 hours per week) for the unit, the remainder of your time should be spent reading the recommended reading, attempting exercises and working on assignment tasks.

Moodle exercises

Periodically, I'll post (unassessed) exercises on the school's Moodle server.

You can complete these in your own time, and they will help you improve your understanding of secure coding concepts.

(All assessments will be completed using the Moodle server, too.)

More information about Moodle will be available in the first lab.

CITS3007 unit content

See the CITS3007 website at https://cits3007.arranstewart.io/schedule/ for the list of topics covered.

The main topics are:

- memory safety and arithmetic errors
- inter-process communication and input validation
- accessing files and resources safely
- cryptography
- secure software development processes

Books you should have

You'll need access to a good C textbook in order to do well in the unit.

YouTube videos or online tutorials will **not** be sufficient!

C is fairly small, as languages go, but some of the details relating to security are subtle.

An **operating systems** textbook will also be helpful – see here.

These aren't textbooks, per se – they cover recommended prior knowledge you should have before starting the unit.

Recommended readings

There is no one textbook that covers all the unit content.

Instead, there are recommended readings for each week, listed on the unit schedule:

- https://cits3007.arranstewart.io/schedule

If you are finding any of the concepts difficult, the recommended readings are a good place to look for clarification.

Working through the readings *before* lectures will also make lectures more useful to you, since you won't be encountering topics for the first time.

Copies of readings should be available via the UWA Library (some as hard-copy textbooks, some as online extracts) – look in the LMS under "Unit Readings".

Assessment

The assessment for CITS3007 consists of an online quiz, a mid-semester take-home test, a project, and a final examination.

All details are on the Assessment page of the unit website at:

- https://cits3007.arranstewart.io/assessment/

All assessments are to be done individually – there is no group or pair work.

Feedback

There'll be an opportunity to give feedback on how the unit is going around week 5 – we'll post a survey form in MS Teams.

Please do make use of the opportunity to comment on the course!

There is *also* an opportunity to provide feedback via the SELT (Student Experience of Learning and Teaching) survey[7] at the end of semester – but that will come too late for us to make any changes *this* semester.

[7] See SELT-Policy.doc (Word document) for UWA's SELT policy.

Prerequisites

The prerequisites for this unit are 12 points of programming units. At UWA, that should mean you're familiar with at least one object-oriented programming language (Java or Python).

If you aren't – let me know.

Advisable prior studies:

Although the prerequisite for this unit is only 12 points of programming, you are advised to take this unit in the third year of your course to ensure you have a comprehensive understanding of computer systems, and will be able to do well in this unit.

Assessment tips – tests and exams

- Quizzes and tests are intended to assess not just theoretical knowledge but *practical* skills, so they're done on computer, and will often ask you to complete exercises using the standard CITS3007 development environment (more on this in the first lab).

  So it's a good idea to have access to the development environment while completing them.
- You're encouraged to use the development environment to work out or check your answers.
- In the exam, to ensure all students have equitable access to the same environment and software, you'll be limited to using a Web browser and Moodle.

Assessment tips – tests and exams

- Quizzes and tests *can* be sat from anywhere you like, and you have some flexibility over exactly when you start them.
- But it is a good idea to make sure you have a reliable Internet connection.
    - You can sit them from your laptop on campus (or from a UWA computer)

Assessment tips – written work

- The project and exam will include not just programming, but explaining and justifying security approaches in English.

  Communicating with others – for instance, documenting your work, writing a security testing plan, or justifying a particular technical approach – is an important part of software engineering.

Assessment tips – written work (cont'd)

- I suggest taking a look at the UWA Library's "Study support" web pages at https://www.uwa.edu.au/library/Help-and-support/Study-support, especially the Communication and Research Skills (CARS) module and materials.

  These web pages provide advice on writing tasks like:

  - finding, evaluating and critiquing evidence
  - making an argument
  - writing a report
- Make sure you're careful in your use of terminology. If your answer is unclear or confusing, you are unlikely to be awarded high marks for an assessment.

Assessment tips – programming work

- Security-critical code doesn't just need to do the "right thing" – it needs to be easily understood and maintained by others, so that it can be *verified* to do the right thing.
- Your code is expected to be clearly written, well-formatted, and easy for others to understand.
- It's better to be *clear* than *clever*. Brian Kernighan (who co-authored the first C programming book), said

  Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.

Assessment tips – programming work (cont'd)

- In the lectures and labs, we'll look at various techniques that can be applied to improve the security of your code.
- In the assessments, it will be up to you to actually *use* them.
- For instance, you'll be expected to enable your compiler's warnings, use static and dynamic analysis on your code to detect problems, and test your code.

Programming environment

We will be using the C programming language (and occasionally, Python).

Project code will be expected to compile and run correctly on a standard Linux environment[8] which we provide in the form of virtual machine (VM) images.

If assessments ever refer to “the CITS3007 standard development environment”, this is the environment they mean.

[8] The standard environment contains C development tools installed on an Ubuntu 20.04 Linux distribution, running Linux kernel version 5.4.0 on an x86-64 processor.

Programming environment, cont'd

The VM images are hosted at

- https://app.vagrantup.com/arranstewart/boxes/cits3007-ubuntu2004

and in the first lab we will look at how you can access them using the open source tools VirtualBox and Vagrant.

Security introduction

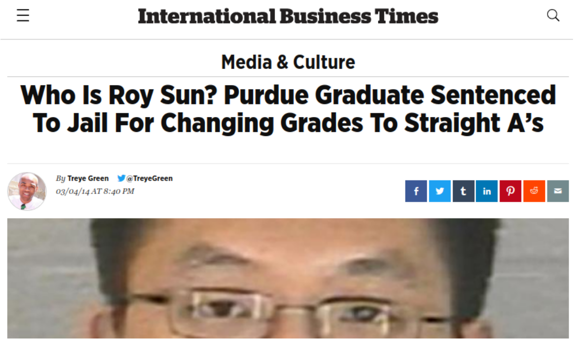
Security goals

Traditionally, information security is based on three goals ("C I A"):

- **Confidentiality** – preventing the unauthorised disclosure of information
- **Integrity** – preventing the unauthorised modification of information
- **Availability** – ensuring timely and reliable access to and use of information by authorised users

Purdue University case

Which security goal was compromised here?



International Business Times

Media & Culture

Who Is Roy Sun? Purdue Graduate Sentenced To Jail For Changing Grades To Straight A’s

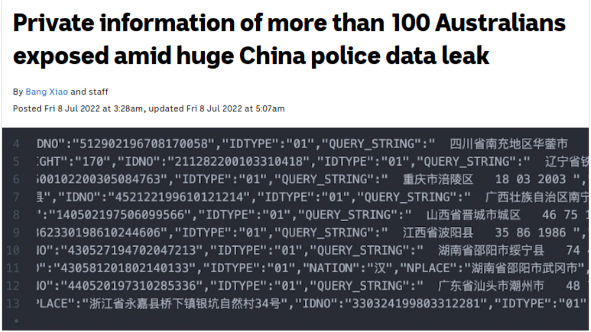
By Treye Green @TreyeGreen
03/04/14 AT 8:40 PM

“During his senior year, Sun missed all of his classes but one. However, with the help of [an accomplice’s] scheme, he was still able to receive straight A’s.”[1]

[1] See Treye Green, “Who Is Roy Sun? Purdue Graduate Sentenced To Jail For Changing Grades To Straight A’s” (International Business Times, 2014)

Chinese police hacking case

How about here?



Private information of more than 100 Australians exposed amid huge China police data leak

By Bang Xiao and staff

Posted Fri 8 Jul 2022 at 3:28am, updated Fri 8 Jul 2022 at 5:07am

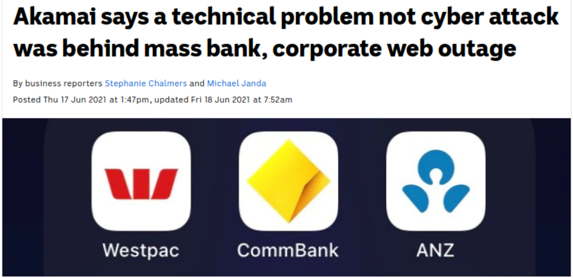
```
DNO":"512902196708170058","IDTYPE":"01","QUERY_STRING":"  四川省南充地区华蓥市
GHT":"170","IDNO":"211282200103310418","IDTYPE":"01","QUERY_STRING":"  辽宁省铁
500102200305084763","IDTYPE":"01","QUERY_STRING":"  重庆市涪陵区   18 03 2003 ",
县","IDNO":"452122199610121214","IDTYPE":"01","QUERY_STRING":"  广西壮族自治区南宁
':"140502197506099566","IDTYPE":"01","QUERY_STRING":"  山西省晋城市城区   46 75
362330198610244606","IDTYPE":"01","QUERY_STRING":"  江西省波阳县   35 86 1986 ",
IO":"430527194702047213","IDTYPE":"01","QUERY_STRING":"  湖南省邵阳市绥宁县   74
O":"430581201802140133","IDTYPE":"01","NATION":"汉","NPLACE":"湖南省邵阳市武冈市",
IO":"440520197310285336","IDTYPE":"01","QUERY_STRING":"  广东省汕头市潮州市   48
PLACE":"浙江省永嘉县桥下镇银坑自然村34号","IDNO":"330324199803312281","IDTYPE":"01"
```

"A hacker claimed in an online forum that they had stolen 1 billion records, mostly belonging to Chinese citizens, in an ongoing bid to sell the information for 10 bitcoins, or almost $300,000."[1]

[1] See Bang Xiao, "Private information of more than 100 Australians exposed amid huge China police data leak" (ABC News, 7 July 2022)

Akamai outage

And here?



Akamai says a technical problem not cyber attack was behind mass bank, corporate web outage

By business reporters Stephanie Chalmers and Michael Janda

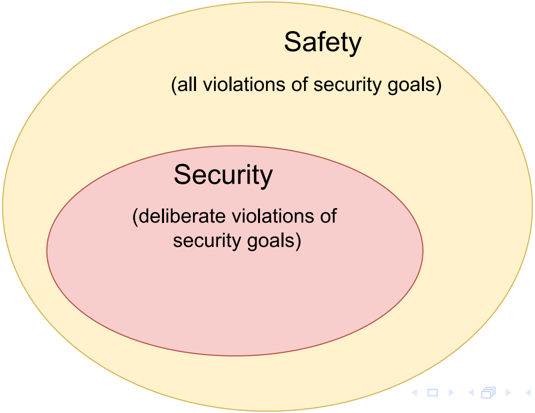
Posted Thu 17 Jun 2021 at 1:47pm, updated Fri 18 Jun 2021 at 7:52am

> *"The company responsible for a mass web outage that hit three of Australia's big four banks, Virgin and Australia Post, among others, has said a routing table error was to blame for the service disruption, not a cyber attack."*[1]

[1] See Stephanie Chalmers and Michael Janda, "Akamai says a technical problem not cyber attack was behind mass bank, corporate web outage" (ABC News, 17 June 2021)

Safety versus security

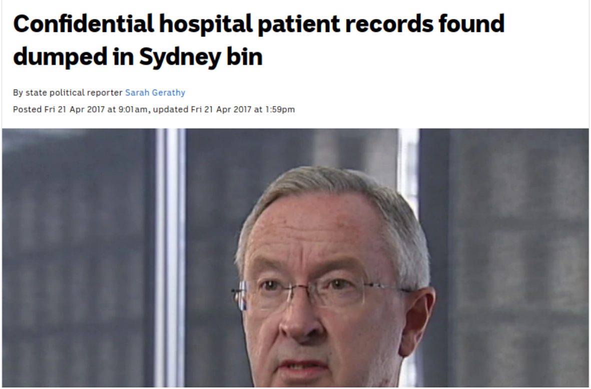
- Generally when we talk about software security, we mean ensuring that bad things don't happen due to *deliberate* actions by others.
- But a related goal is software *safety*, which is ensuring that bad things don't happen, whether deliberate or not.


Safety
(all violations of security goals)
Security
(deliberate violations of security goals)

Safety versus security

- The security goals we mentioned can often be compromised by accident, as well as intentionally –
  - confidentiality can be threatened by (say) accidentally leaving a USB drive full of employee details in a car-park
  - integrity can be threatened if we fail to keep proper backups and suffer a power outage
  - availability can be threatened if we make a mistake in routing Internet traffic

Accidental compromise



Confidential hospital patient records found dumped in Sydney bin

By state political reporter Sarah Gerathy

Posted Fri 21 Apr 2017 at 9:01am, updated Fri 21 Apr 2017 at 1:59pm

"More than 700 public patients and hundreds more in the private system have had their privacy breached after letters from their specialists to GPs were found dumped in a Sydney bin."[9]

[9] See Sarah Gerathy, "Confidential hospital patient records found dumped in Sydney bin" (ABC News, 21 April 2017)

Other security goals

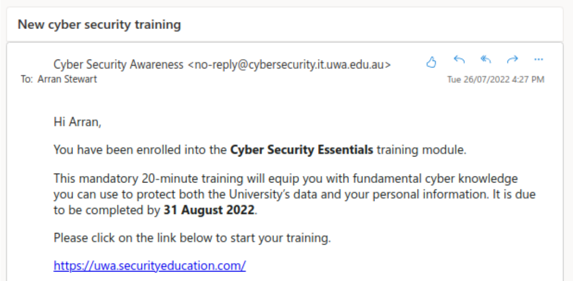
Most security experts augment the “C I A” triad with additional goals; two commonly proposed ones are:

- **Authenticity** – being confident in or able to verify the genuineness of a message or information
- **Accountability** – the ability to trace actions back uniquely to the entity that took those actions.
  (Or, sometimes, **Non-repudiation** – the creation of evidence that an action has occurred, so that a user cannot later falsely deny taking that action.)

Example – alleged training email

I received the following email, purporting to be from UWA's IT Services.

Is it genuine?



New cyber security training

Cyber Security Awareness <no-reply@cybersecurity.it.uwa.edu.au>
To: Arran Stewart
Tue 26/07/2022 4:27 PM

Hi Arran,

You have been enrolled into the **Cyber Security Essentials** training module.

This mandatory 20-minute training will equip you with fundamental cyber knowledge you can use to protect both the University's data and your personal information. It is due to be completed by **31 August 2022**.

Please click on the link below to start your training.

https://uwa.securityeducation.com/

Example – alleged training email

About the email:

- It asks me to click on a non-UWA link, https://uwa.securityeducation.com/, which in turn asks me to provide my UWA user ID and password.
- IT Service's page on phishing emails[10] says that "Any email from a legitimate business such as the University or your bank will give a telephone number and postal address", which this email does not.

[10] At https://cybersecurity.it.uwa.edu.au/stay-secure/email-scams-phishing

Example – alleged training email

About the email:

- It asks me to click on a non-UWA link, https://uwa.securityeducation.com/, which in turn asks me to provide my UWA user ID and password.
- IT Service's page on phishing emails[10] says that "Any email from a legitimate business such as the University or your bank will give a telephone number and postal address", which this email does not.
- In fact, the email *is* from UWA's IT services.

[10] At https://cybersecurity.it.uwa.edu.au/stay-secure/email-scams-phishing

Threats, vulnerabilities, incidents & attacks

These concepts all relate to the ways in which information security can be or is compromised.

- **Threat**: Anything that has the potential to cause harm or loss.
  - Threats can be *natural threats* (floods, hurricanes, solar flares), *unintentional threats* (an intern accidentally deletes everything from your server's filesystem), or *intentional threats* (activities done deliberately: e.g. altering or deliberately deleting server data).
  - Could be thought of as "a source of danger".

When we talk about *security* threats, we mean harm or loss due to a compromise of a security goal.

Threats, vulnerabilities, incidents & attacks

- **Vulnerability**: A flaw or weakness in a system's design, implementation or configuration that could be exploited to compromise security.

  Some examples:

  - Hard-coded default passwords
  - Buffer overflows
  - Poor validation and sanitization of program input

Threats, vulnerabilities, incidents & attacks

- **Attack**: A situation where someone (the attacker) deliberately exploits a vulnerability and compromises security goals.
- **Incident**: Much the same, except it arises from a non-deliberate act. Security incidents can still be costly and harmful, however, so we need to take them into account.

References

- Australian Cyber Security Centre, *ACSC Annual Cyber Threat Report: 1 July 2020 to 30 June 2021* (Kingston ACT, 2021), https://www.cyber.gov.au/sites/default/files/2021-09/ACSC%20Annual%20Cyber%20Threat%20Report%20-%202020-2021.pdf
- UNSW Canberra, “Cybercrime an Estimated $42 Billion Cost to Australian Economy,” UNSW Canberra Latest News, created December 6, 2021, accessed July 28, 2022, https://www.unsw.adfa.edu.au/newsroom/news/cybercrime-estimated-42-billion-cost-australian-economy.
- Deloitte Access Economics, *Update to the Economic of Natural Disasters in Australia*, Special Report (Sydney, NSW, 2021), https://www.iag.com.au/sites/default/files/Newsroom%20PDFs/Special%20report%20_Update%20to%20the%20economic%20costs%20of%20natural%20disasters%20in%20Australia.pdf.