DTiF
Digital Technologies in focus
Initiative of and funded by the Australian Government Department of Education, Skills and Employment

acara AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY

CLASSROOM IDEAS: YEARS 9–10

Privacy and security

Image source: Wikimedia
Attribution: Cinemantique

The need for security

A huge amount of our personal information is now stored online. We rely on it being secure and private. If we are not able to manage our personal information, we expose ourselves to huge risks. Our money could be used by others, our reputation ruined and our identity stolen.

One way to reduce the likelihood of our personal information being compromised is to look at how cybercriminals and others might try to gain access, and then we can think and act defensively against them. That is, if we can think like attackers we can reduce the chances of systems and data being compromised.

Contents

Almost all software systems today face a variety of threats, and the number of threats grows as technology changes. Threats can come from outside or within organisations, and they can have devastating consequences. Attacks can disable systems entirely or lead to the leaking of sensitive information, which would diminish consumer trust in the system provider.

To prevent threats from taking advantage of system flaws, administrators can use threat-modelling methods to inform defensive measures. Threat-modelling methods are used to create:

- an abstraction of the system
- profiles of potential attackers, including their goals and methods
- a catalogue of potential threats that may arise.

Many threat-modelling methods have been developed. They can be combined to create a more robust and well-rounded view of potential threats. Not all of them are comprehensive; some are abstract and others are people-centric. Some methods focus specifically on risk or privacy concerns.

Threat modelling should be performed early in a system's development cycle when potential issues can be caught early and remedied, preventing a much costlier fix later. Using threat modelling to think about security requirements can lead to proactive architectural decisions that help reduce threats from the start.
Any student activities should be conducted using a 'Good faith code'. That is, students should always make known the purpose of collecting data or why they are accessing systems.

Class discussion: insecure 'forgotten password' approaches

People forget their passwords now and then. In the past, some online services would let you enter your email address that you originally signed up with, and then would send your password to that email address.

1. Why is this a bad idea?
2. What does it tell you about the way the online service stores its passwords?
3. What might be a more secure way for the online service to use email to help you when you forget your password?

Image source: Pixabay

Multi-factor authentication

Passwords on their own provide one barrier to keep unauthorised people out. As well as having secure passwords, we can also require extra ways of proving who we are to get to our online content.

Multi-factor authentication, as its name implies, requires a user to pass at least 2 tests of their identity. One could be the traditional password, while the other(s) could include fingerprint or face recognition, a message sent to a mobile phone, a security question or some other proof that you are who you say you are. cyber.gov.au has a good summary.

Image source: Pixabay

Student activity 1: Attackers and multi-factor authentication

1. Pick one form of authentication (other than a password) used in multi-factor authentication.
2. Devise possible ways in which a determined attacker might be able to get past that form of authentication.
3. Create an infographic listing these to warn others. You could use an online tool such as Canva www.canva.com.

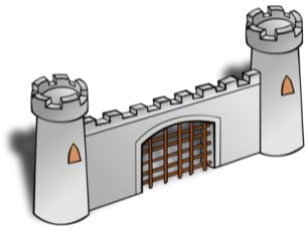

Image source: Openclipart
Attribution: Nicubunu

Stopping attackers from getting passwords from end applications

We have looked at how to stop attackers guessing or cracking passwords, but there is another problem. Every system that we log into must be able to check that our password is correct, so it has to have something stored to do this. If attackers could get access to whatever is stored in these systems, surely our passwords would be compromised.

Luckily, that is not the case. A good application does not store passwords directly; it stores a highly modified version of them. A common way to do this is via a hash algorithm. The application takes the actual password, passes it through this hash algorithm and stores whatever the algorithm produces. When you try to access the application with your password, the application puts the attempted password through the same algorithm and compares what comes out with what it has previously stored. If they match, your password is correct.

The algorithm (typically the SHA-256 algorithm) is mathematically quite powerful and is a sort of one-way process. It is almost impossible to work out the password from the hashed output. It's a bit like the paint colour systems used by paint suppliers – if they add a particular amount of each of the various tints into a particular base paint, they will get a specific colour. If they do the same thing 10 times, they will get that same colour every time. But just looking at the final colour, it is not easy to figure out the exact amount of each tint that was used.

Student activity 2: Using a hashing tool

Image source: Pixabay

1. Use the SHA2 hashing tool on csfieldguide.org.nz/en/chapters/coding-encryption/storing-passwords-securely/ to see how a simple password gets converted into a much more complicated hashed value.
2. Try the same password several times.
3. Try changing one character.
4. Change a character from upper to lower case or vice versa.
5. What do you notice about the hashed output?

Student activity 3: Can hashed passwords be guessed?

Image source: Pixabay

1. Use the application at csfieldguide.org.nz/en/interactives/password-guesser/?salted=false to see if you can work out the passwords of the various people listed, by trying to match the stored hash value. You might notice that, even using the supplied clues, it is not always easy to find the passwords.
2. Explore the different versions of hashing algorithms by using an online generator: passwordsgenerator.net/sha256-hash-generator
   By comparing the output from SHA1, MD5, SHA256 and SHA512, what do you notice about the evolution of the
3. hashing algorithms?
4. Read about the development of hashing algorithms over time: medium.com/@rauljordan/the-state-of-hashing-algorithms-the-why-the-how-and-the-future-b21d5c0440de
   Create a 2-minute video to explain it to others.

Student activity 4: Understanding the terminology

Threat-modelling terminology is specific to the task at hand. You have recently been appointed as the security engineer for a local company. Create a document explaining the key terms that all employees can easily access and understand. This could be in the form of an FAQ document or an interactive dictionary. Following are some key terms to get you started. Can you add to the list?

- Mitigations
- Controls
- Preventions
- Likelihood
- Impact
- Data flow diagram
- Trust boundary

Student activity 5: Types of threat models

Attack trees

Using attack trees to model threats is one of the oldest and most widely applied techniques on systems, some of which are autonomous or semi-autonomous systems. Attack trees were initially applied as a standalone method and have since been combined with other methods and frameworks.

Attack trees are diagrams that depict attacks on a system in tree form. See Figure 1. The root of the attack tree is the attack goal (green), and the leaves are ways to achieve that goal (grey). Each goal is represented as a separate tree.

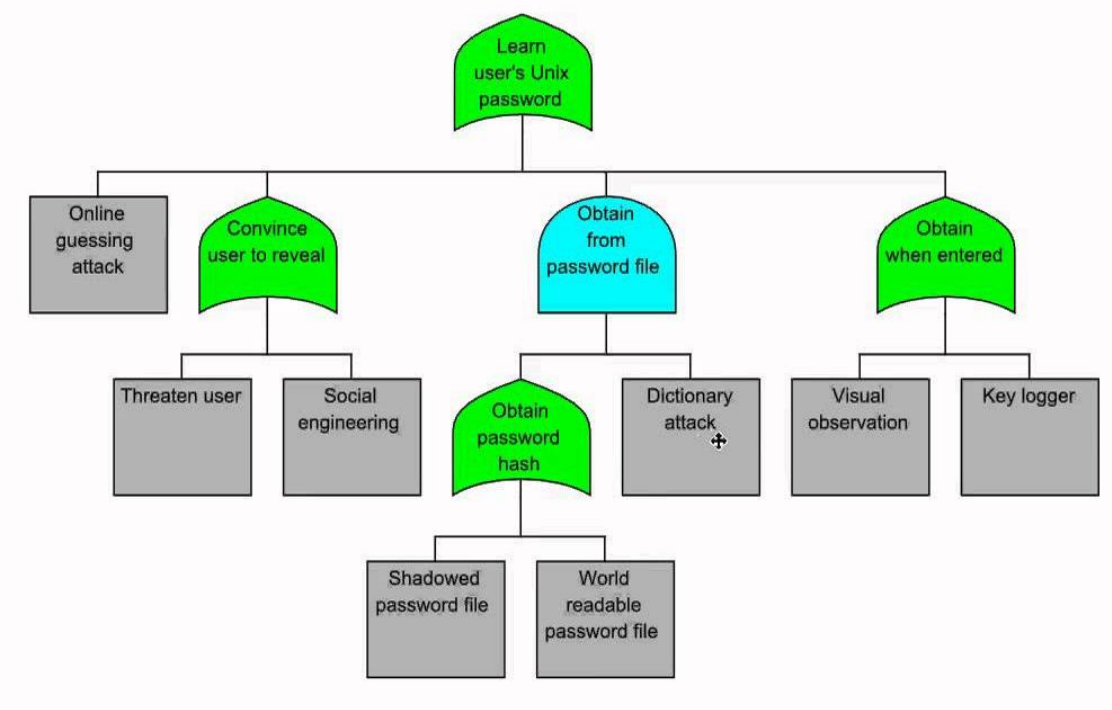

Figure 1: Screen shot source: www.youtube.com/watch?v=2ZT3xNOa6iQ

There are many other models that could be used or combined. In your group, explore various threat models such as PASTA, TRIKE, OCTAVE, VAST Modelling or hTMM.

Student activity 6: Developing a cyber security threat model

In your new role as a software developer, you have been tasked with creating a cyber security threat model. Choose one of the threat models from Student activity 5 to focus on. Using a template similar to the one below, work through the process of creating a suitable threat model for an application.

Generally, developers perform threat modelling in 4 steps:

- Diagram**:** What are we building?
- Identify threats**:** What could go wrong?
- Mitigate**:** What are we doing to defend against threats?
- Validate**:** Have we acted on each of the previous steps?

1. Choose an application or system which is key to the running of a company. This might be a communication application, an online ordering system, or a financial or banking system.
2. Identify application objectives. It is important to identify the objective of the application you are assessing, including identifying security and compliance requirements.
3. Draw a data flow diagram of the entire system. Understanding the design of the application is key to performing threat modelling. Figure 2 shows an example of a data flow diagram for an online food delivery company.

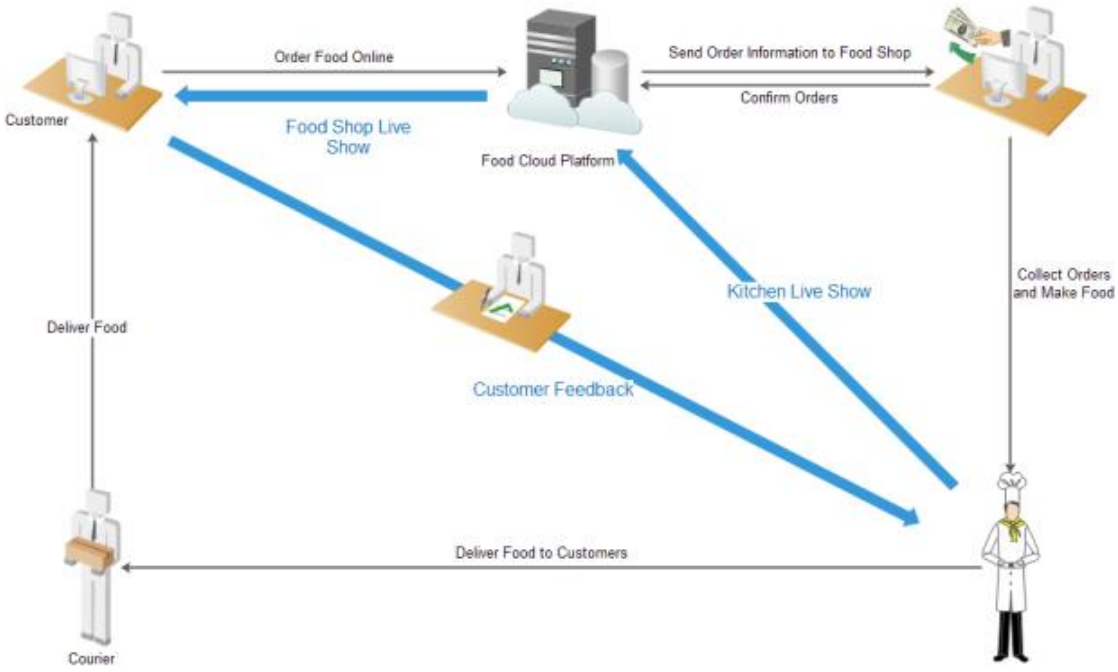

Figure 2: Data flow diagram www.edrawsoft.com/template-online-food-ordering-workflow.html

4. Decompose and model the system, identifying threats and any mitigations required.
5. Present and explain your threat model to the class.

Australian Privacy Principles

Many organisations are bound by the Australian Privacy Principles: www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-quick-reference/

The principles describe how organisations should collect, manage and inform users about personal information; who they can disclose it to and how they must ensure its quality and accuracy. See Figure 3 for a quick reference poster.

Figure 3: Australian Privacy Principles poster
www.oaic.gov.au/__data/assets/pdf_file/0020/1289/app-quick-reference-tool.pdf

However, even though those organisations keep users' data private and secure, an individual's own access to data via password systems is still reliant on that individual (and only that individual) having access to a password.

It is no surprise that there are people in the world who spend their time trying to find out the passwords so they can use the online services or funds for their own purposes. Similarly, governments and law enforcement are always looking for ways to prevent this happening, and to find ways to catch these criminals.

The process of unlawful system penetration is referred to as 'attacking'; however, some people still wrongly call it 'hacking'. Hence the common complaint: "My Facebook account/bank account/Netflix account has been HACKED". It is important to distinguish between ethical hacking, where professionals attempt to access systems to uncover vulnerabilities, and cyber attacks, which are unlawful attempts to access systems for criminal activities. Refer to document Teaching Tips – Hacker verses Attacker

By looking at some of the techniques used by cybercriminals we can make our own online presence more secure and less likely to be attacked.

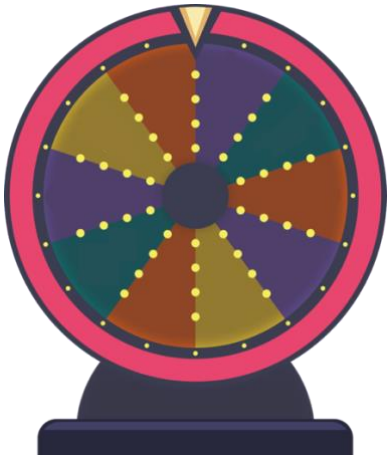

Image source: Pixabay

Student activity 7: Cracking a single-character password

1. Use this program scratch.mit.edu/projects/562849958/ to see how long it takes to crack a typical single-character password. The exact time depends on which particular password is chosen, so you will need to feed it about 20 such passwords to get a reasonable average.
2. You can choose a random single-character password by using a 26-letter spinner or a website such as wordwall.net/resource/42874/english/alphabet-wheel
3. As you choose each password, type it into the Scratch program and note how long it takes.
4. After doing this 20 times, record the time taken in a spreadsheet program.
5. Use the data to produce a distribution curve (histogram) of the times taken.
6. Find the average of the times to get a typical value for the time to crack the password.
7. Compare this with results from other groups in the class.

Student activity 8: Cracking a 2-character password

Image source: Pixabay

Obviously, a single-character password is very insecure. The security can be improved a bit by allowing at least 2 characters (so a password might be cf or kt, etc.)

1. Use this program scratch.mit.edu/projects/562855304/ to work through all the 2-letter passwords (starting with aa, ab, ac … then ba, bb, bc … through to za, zb, zc ... zz).
2. Feed each password into the password cracker and see how long it takes to find the password.
3. Repeat this 20 times and produce a histogram.
4. Compare your results with other groups.

Questions: Effect of password length

1. What effect does using 2 letters rather than one have on the time taken for the program to crack the password?
2. Does the extra character make the password harder to crack?
3. What effect would adding a third character have to the average time? Is there a mathematical pattern you can see emerging? Can you predict how long it would take on average to crack an 8-character password?

Extension for students

Single-character password – how hard is it to work out?

If a person decided to use a single lower-case letter as their password (and we knew that they had), it would be very easy to guess their password. We could just try each letter, working our way through the alphabet. We could guarantee to guess it in 26 tries. Depending on which letter they chose and where in the alphabet we start, we might get the password on anywhere from our first try to our 26th try. So, it would take us on average about 13 tries.

Extension activity 1: Write a program to simulate cracking a single-character password

How the one-character password guessing program works

The program is quite simple but involves some useful programming structures. Access the sample program from Activity 7: scratch.mit.edu/projects/562849958/

First, we need to have a collection of the characters we are going to test in our password. This would be the 26 letters of the alphabet. We could store these in a data structure called a list. In the Scratch program above, the list has been named CharactersList.

We would then use a control structure such as a loop to work our way through the list, each time testing the character we read to see if it is the 'secret' password. Of course, we cannot actually test it against a secure website (because we do not have one with such a simple password system) but we can have the secret password stored and test it within the program.

We can keep track of how many guesses our program took to find the password, and how long it took.

We also need to think about how we will stop the program – there is not much point to keep trying further letters once we have found the password. Often a flag is used to mark this – a variable that is set to the value FALSE (or 0) at the start of the program but changed to TRUE (or 1) when the goal of finding the password is achieved. We can then tell our program to keep running until that variable becomes TRUE (or value 1). In the Scratch program above, the flag is named FoundIt.

Here is a sample general-purpose programmed version written in Python: github.com/shabysheik/characterguess/blob/master/charguess1.py

You can use the simple visual program or the general-purpose program and then modify it to reflect the extra possibilities below.

Extension activity 2: Modifying the single-character program

1. What if the single-character password could be either lower case OR upper case (capital)? How many tries would we need to guarantee we would find the password? How many tries on average would it take?
2. Modify your copy of the Scratch or Python program to reflect this.
3. What if the one-character password was chosen from lower- and upper-case letters and also the digits 0–9?
4. Modify your copy of the Scratch or Python program to reflect this.
5. Why not just use a random number approach to randomly guess at the letters, rather than work through all the possibilities methodically?
6. Does that approach have any advantages or disadvantages?
7. Does it matter if we create our list of characters as a, b, ... , x, y, z or z, y, x, ...c, b, a or some other order?
8. Here is a general-purpose Python version of a 2-character password-cracking program: github.com/shabysheik/characterguess/blob/master/charguess2.py Write your own general-purpose language version and see if you can improve on the example program.

How the 2-character password-cracking program works

Obviously, a one-character password system is very, very simple to crack. Real passwords are more complex. Let's see what we need to do if we allow passwords to be 2 characters (each in the range a, b, ... , x, y, z at this stage).

We need an algorithm that describes how we'll approach this. We could randomly pick 2 letters, join them and see if that is the password, then try 2 more at random, and so on.

That approach is not ideal (why not?)

Image source: Wikimedia
Attribution: Sheila Sund

Ransomware

If attackers can find ways to access computer systems (such as those that run business operations in large companies) their actions could potentially lock others out of those systems, or even encrypt the company's data so it cannot be read. Password security (and other forms of security) are critical for those people whose job involves controlling and administering these business systems.

An increasingly common problem is 'ransomware' attacks where a company's data is encrypted so it is no longer readable until the company pays a ransom to have it decrypted. While password cracking is not the usual way for criminals to make ransomware attacks, it is worth knowing what the attacks involve. The following activity could be run as a jigsaw or group activity:

Image source: Pixabay

Student activity 9: Ransomware

1. Read www.cyber.gov.au/ransomware and www.cyber.gov.au/ransomware/protect-yourself-against-ransomware-attacks (Australian Cyber Security Centre advice on ransomware).
2. Read and discuss the case studies at www.cyber.gov.au/ransomware/examples-ransomware-incidents
3. Read and discuss the real-world examples on pages 5 and 6 of www.cyber.gov.au/sites/default/files/2020-10/Ransomware in Australia %28October 2020%29.pdf
4. Identify the most likely businesses in your area to be targets for ransomware. This might include finding the businesses that have a large number of customers, relatively few IT staff, and who could not operate at all without their data.
5. Produce an explainer video and an infographic for one of the identified businesses that would help them verify that they were protected from ransomware attacks.

Man/Monster/Monkey/Machine in the Middle (MITM) attacks

MITM attacks refer to a method of intercepting information sent between a user and the server that hosts whatever online services are being used. The approaches can be used to gain passwords and other data, or more generally to imitate a user's activity for the benefit of the attacker.

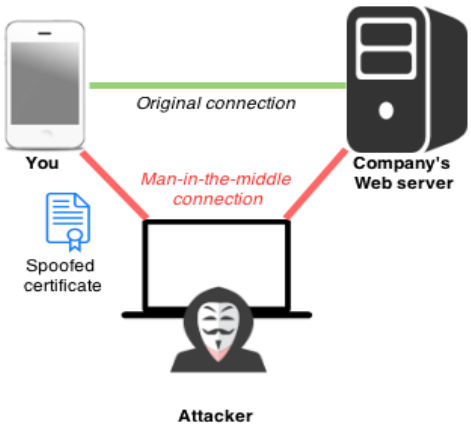

Image source: Wikimedia
Attribution: Nasanbuyn

Students read the following articles and discuss as a class:

1. Man in the middle attack
en.wikipedia.org/wiki/Man-in-the-middle_attack
2. Eavesdropping
en.wikipedia.org/wiki/Eavesdropping
3. Session hijacking
us.norton.com/internetsecurity-id-theft-session-hijacking.html

Student activity 10: Mum in the middle

You and a friend are secretly planning to hold a surprise party for your mother (or your friend's mother). Unfortunately, you are in lockdown, and have no internet or phone connection. The only way you can get messages between you and your friend is via the mother. You do not want the mother to read the messages.

Image source: Pixabay

1. Work out 3 ways that you could send these messages. You both have a computer, a vast library of books on computing, and plenty of paper. Any message you send has to be given to the mother to be conveyed.
2. Can you build in something that will let you know if the message has been intercepted by the mother?
3. Try your solutions, using a classmate from another group as the 'mother'.

Links to the Australian Curriculum

Table 1: Aspects of the Australian Curriculum: Digital Technologies Years 9 and 10 which may be addressed depending on the task

<table>
<tr><td>Digital Technologies
Achievement standard</td><td colspan="3">By the end of Year 10, students develop and modify innovative digital solutions, decompose real-world problems, and critically evaluate alternative solutions against stakeholder elicited user stories. Students acquire, interpret and model complex data with databases and represent documents as content, structure, and presentation. They design and validate algorithms and implement them, including in an object-oriented programming language. Students explain how digital systems manage, control, and secure access to data; and model cyber security threats and explore a vulnerability. They use advanced features of digital tools to create interactive content, and to plan, collaborate on, and manage agile projects. Students apply privacy principles to manage digital footprints.</td></tr>
<tr><td>Strand
Sub-strand</td><td colspan="3">Knowledge and understanding
Digital systems
Processes and production skills
Privacy and security</td></tr>
<tr><td>Content descriptions</td><td colspan="3">develop techniques to acquire, store and validate data from a range of sources using software, including spreadsheets and databases AC9TDI10P01
investigate how hardware and software manage, control and secure access to data in networked digital systems AC9TDI10K01
develop cyber security threat models, and explore a software, user or software supply chain vulnerability AC9TDI10P13
apply the Australian Privacy Principles to critique and manage the digital footprint that existing systems and student solutions collect AC9TDI10P14</td></tr>
<tr><td rowspan="2">Technologies Core concepts</td><td rowspan="2">Systems
Data
Systems Thinking
Computational Thinking
Interactions
Impact</td><td>Digital Technologies Core concepts</td><td>digital systems
data representation
algorithms
privacy and security</td></tr>
<tr><td>General capabilities</td><td>Digital Literacy
Practising digital safety and wellbeing
Manage online safety
Manage digital privacy and identity
Manage digital wellbeing
Critical and Creative Thinking
Personal and Social capability
Ethical Understanding</td></tr>
<tr><td>Cross-curriculum priorities</td><td>Sustainability</td><td>Learning area or subject connections</td><td>Mathematics</td></tr>
</table>

Learning area or subject connections

Mathematics

The Australian Curriculum: Technologies provides contexts within which Mathematics understanding and problem-solving skills can be applied and developed. Digital Technologies and Mathematics share a focus on computational thinking, in particular in data

acquisition and interpretation, models and simulations, and generalising. The Digital Technologies curriculum supports students to apply the knowledge and skills that underpin pattern recognition.

The Digital Technologies curriculum supports students to apply the knowledge and skills that underpin pattern recognition, data acquisition, and interpretation and representation, which form the basis of the Mathematics strand, *Statistics*. Digital Technologies develops students' basic understanding of algorithms in the early years, which Mathematics then builds on. The implementation, design and creation of algorithms form an integral part of a computational approach to learning in Digital Technologies and Mathematics.

Through these classroom ideas, students will have opportunities to:

- represent the distribution of multiple numerical data sets using comparative representations. Compare data distributions with consideration of centre, spread and shape and the effect of outliers on these measures AC9M9ST03 (Year 9)
- choose appropriate forms of display or visualisation for a given type of data. Justify selections and interpret displays for a given context AC9M9ST04 (Year 9)

Teachers could use the data from the 2 password-cracking programs as the basis for data presentation and interpretation in several forms.

- compare data distributions for continuous numerical variables using appropriate data displays including boxplots. Discuss the shapes of these distributions in terms of centre, spread, shape and outliers in the context of the data AC9M10ST02 (Year 10)

Teachers could base the construction and interpretation of boxplots on the data from the 2 password-cracking programs.

Resources

Useful links

- Australian Cyber Security Centre: passwords, PINs and passphrases
  www.cyber.gov.au/acsc/view-all-content/advice/passwords-pins-and-passphrases
  Includes information on:
  - Password tiers
  - Password and PIN hygiene
  - Where to get help
- Australian Cyber Security Centre: creating strong passphrases
  www.cyber.gov.au/acsc/view-all-content/publications/creating-strong-passphrases
- Australian Cyber Security Centre: multi-factor authentication
  www.cyber.gov.au/acsc/view-all-content/advice/multi-factor-authentication
- The Conversation: can I still be hacked with 2-factor authentication enabled?
  theconversation.com/can-i-still-be-hacked-with-2fa-enabled-144682
- Office of the Australian Information Commissioner: preventing data breaches
  www.oaic.gov.au/privacy/notifiable-data-breaches/preventing-data-breaches-advice-from-the-australian-cyber-security-centre/
  Includes information on preventing data breaches, including:
  - Passwords
  - Software systems
- Digital Health: Supporting a positive security culture – passwords
  www.digitalhealth.gov.au/sites/default/files/2020-11/Passwords-Supporting_a_positive_security_culture.pdf
  (downloadable booklet on passwords and information security practices)

- CERTNZ: How to create a good password
  www.cert.govt.nz/individuals/guides/how-to-create-a-good-password/
- CERTNZ: Keep your data safe with a password manager
  www.cert.govt.nz/individuals/guides/keep-your-data-safe-with-a-password-manager/
- ITPro: The top 12 password-cracking techniques used by hackers
  www.itpro.co.uk/security/34616/the-top-password-cracking-techniques-used-by-hackers
- Character guess – 2 sample Python programs to simulate simple password guessing strategies
  github.com/shabysheik/characterguess
- Character guess video tutorial
  https://youtu.be/bhdMDAzbaSQ
- eSafety education
  www.esafety.gov.au/educators
- eSafety education – Keeping your online accounts secure
  www.esafety.gov.au/young-people/keeping-your-online-accounts-secure
- eSafety education – Protecting your identity
  www.esafety.gov.au/young-people/protecting-your-identity
- eSafety education – Consent for sharing photos and videos
  www.esafety.gov.au/young-people/consent-sharing-photos-videos
- eSafety education – Your digital reputation
  www.esafety.gov.au/young-people/your-digital-reputation
- eSafety education – Be Deadly Online
  www.esafety.gov.au/educators/classroom-resources/be-deadly-online
- eSafety education – What's your brand?
  www.esafety.gov.au/educators/classroom-resources/whats-your-brand
- Grok Academy Cyber Security Challenges (Australian students in Year 5–12 have subsidised access to this course.)
  aca.edu.au/projects/cyber-challenges/
- Try Hack me (free with individual account set up)
  tryhackme.com/
- Cyberstart – (12 free programs available by signing in with an individual email account. No credit cards required)
  cyberstart.com/
- Australian Cyber Security Centre – ransomware
  www.cyber.gov.au/ransomware
- Man in the middle attack
  en.wikipedia.org/wiki/Man-in-the-middle_attack
- Eavesdropping
  en.wikipedia.org/wiki/Eavesdropping
- Session hijacking
  us.norton.com/internetsecurity-id-theft-session-hijacking.html

All images in this resource are used with permission.