Assignment 2 Portfolio

Task Overview

You will create a portfolio to demonstrate that you can apply cybersecurity technologies to identify vulnerabilities, protect computer systems and automate common IT processes and tasks. You will be required to:

- PowerShell: Write a PowerShell script to investigate aspects such as Windows processes, packages, services and servers
- Plan: Apply a framework to mature an organisation's cybersecurity management. You will need to frame the organisation's tolerance for risk, create an asset inventory, model the data flows for crucial data, assess the cyber risks and plan the implementation of controls.
- Bash: Write a Bash script that runs on a provided Kali virtual machine to investigate aspects such as Linux processes, packages, services and servers
- Write a Python script that runs on a provided Kali virtual machine to investigate aspects such as computer processes, packages, services and servers

Portfolio

Create a private repository on GitHub and invite your tutor and the unit coordinator.

Due

Weeks 4, 5, 7 and 10.

- PowerShell is due in Week 4
- Plan is due in Week 5
- Bash is due in Week 7
- Python is due in Week 10

Return

Feedback will be provided within 2 weeks of the due date.

Submission Overview

Submit artefacts to both a private Git repository and to the unit website. Submit a link to your private repository to the unit website.

Criteria Overview

You will be marked on aspects such as script functionality, code modularity, avoidance of deprecated functions in scripts, coding style, code documentation, code testing, use of Git, risk tolerance elicitation, quality of asset inventory and data flow models, and the identification and justification of risks and controls.

The PowerShell, Bash and Python tasks contribute 10% each to your grade. The Plan contributes 20% to your grade.

Task 1 PowerShell

Due: Week 4

In this task you will develop a PowerShell script. Create a script called ass2.ps1. Your script should be designed to run on any Windows 10 or later PC. The script should collect information from the Windows environment and produce a report identifying any issues with the system. Each step should remain silent if successful and only produce output if there is an issue that needs to be addressed. Create a new function for each step.

1. List any processes that are hogging the CPU. Provide an explanation of your test in your code's documentation. Submit an example run of your script detecting a CPU hogging process.
2. Check that Windows Defender antimalware toolkit is enabled.
3. Create a software inventory that is a list of approved programs. Check that only approved programs are installed. Submit your list of approved programs. Remove one of the approved programs from your list. Submit an example run of your script detecting an unapproved program.
4. Create a list of services approved for running. Check that only approved services are running. Remove one of the approved services from your list. Submit an example run of your script detecting an unapproved service.
5. Please implement one of the following three tasks: either part a, part b or part c. You do not need to implement all of part a, b and c.
    a. Check the Windows machine has been hardened against DNS poisoning attempts that use failed DNS requests. That is, check the registry property HKLM\SOFTWARE\Policies\Microsoft\Windows NT\DNSClient\EnableMulticast is set to 0. On some computers, the key might instead be HKLM\SOFTWARE\Policies\Microsoft\WindowsNT\DNSClient\EnableMulticast.
    b. Check Cortana has been disabled on the Windows PC. That is, ensure that HKLM\SOFTWARE\Policies\Microsoft\Windows\Windows Search\AllowCortana is set to 0.
    c. Check that the Network Time Protocol (NTP) is enabled. That is, check that NTP Enabled of HKLM\SYSTEM\State\DateTime is set to 1.
6. Check the script is running under your hardcoded username.

Task Submission

Commit a single script called ass2.ps1 and a Word document containing your evidence to your private Git repository. Include a link to your private repository in your Word document. Submit ass2.ps1 and your Word document to the unit website.

Task 2 Plan

Due: Week 5

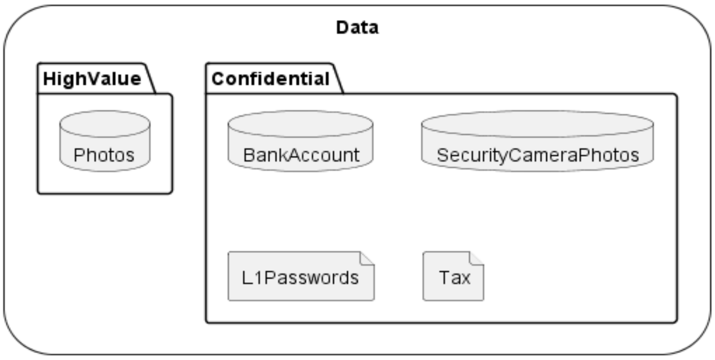
In this task you will help a friend (or family member) to manage their cybersecurity risks. You will help them prioritise their digital assets, identify their cybersecurity risks, visualise their data flows and plan controls to reduce their cybersecurity risks.

Try to select a friend (or family member) who you believe has a higher than usual cybersecurity risk, for example, due to a lower technical ability. Your friend will need to be willing to be part of a 5 minute video in which you both discuss risks.

You will develop a Word document that contains the following sections:

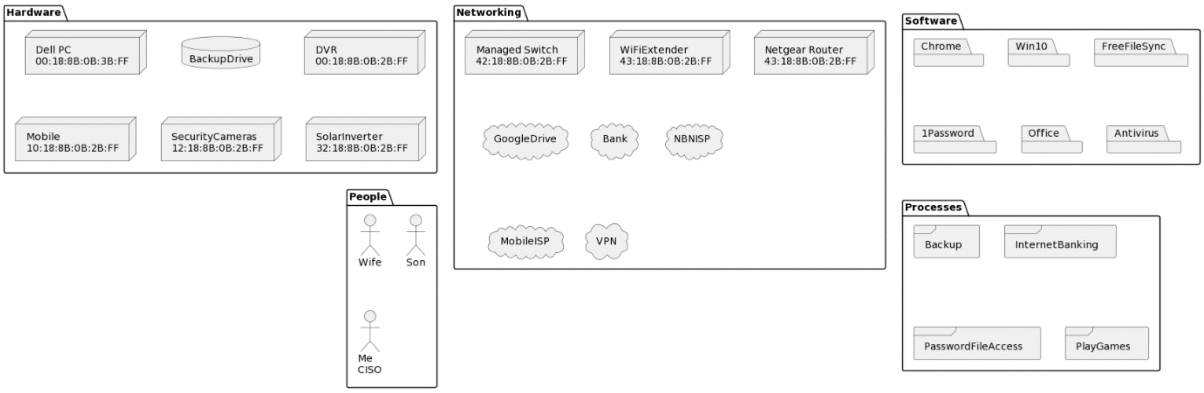
- Ensure your friend (or family member) knows about common cybersecurity risks, such as commercial data spills, identity theft, phishing, malware, ransomware and weak passwords and risk controls such as updating their devices, activating MFA and backing up their devices as detailed, for example, on cyber.gov.au. Cyber Security for Employers is a 15 minute video that provides a nontechnical introduction.
- Frame Risk Tolerance: provide an overview of your friend's (or family member's) cybersecurity risks. Summarise the level of cybersecurity risk that your friend is willing to face. Determine the amount of money and time your friend is willing to spend each week to perform tasks to manage their cybersecurity risk.
- Data Inventory (Top 5): provide a table or visualisation of your friend's top 5 data assets, for example, their photos and financial records. Provide attributes such as their secrecy classification and/or their value.

	Classification/Value
Bank account	Confidential
Photos	High value
…	

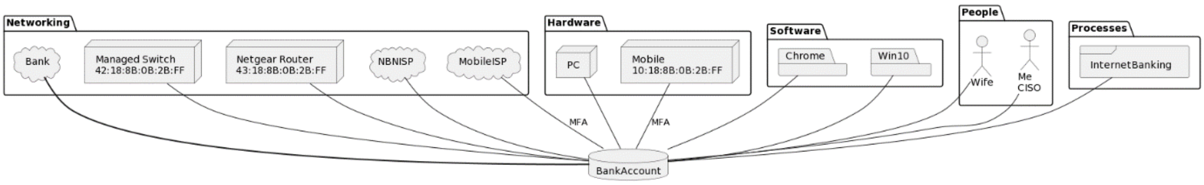


- Non-data Inventory (Top 10): provide table(s) or a visualisation of at least your friend's top ten non-data assets. Non-data assets include networking hardware, other hardware, software, people and their important processes or procedures such as backup or accessing a password file on mobile devices. A thorough inventory would also record attributes such as the asset's MAC, serial number, model and version to help identify vulnerabilities. For privacy reasons, do not include those attributes in your assignment.

Asset Type	Description/Location
Hardware	PC
…	



- Data flows (Top 2): for at least the top two most important data assets, provide table(s) or models to show which assets are involved in storing, processing or transmitting the data asset.



- Risks (Top 10): create a table similar to the following with threat categories in columns across the top and assets in rows down the left hand side. Work with your friend to identify and prioritise their top ten risks helping them to understand the threats, vulnerabilities and consequently the likelihood and impact of the risks. Submit a 5 minute video recording of you working with your friend on this table – the recording does not need to include their face, just your face and both voices.

		Threats											
Type	Asset	IP compromises	Espionage or Trespass	Forces of nature	Human error or failure	Info extortion	Quality-of-service deviations from	Sabotage or vandalism	Software attacks	Technical hardware failures or errors	Technical software failures or errors	Tech. obsolescence	Theft
Data	Bank Account		3		2				1				
	…												
Hardware	Backup Drive					7				6			5
	Mobile									8			9
	…												
Networking	Router			4									
	…												
Software	Chrome												
	…												
People	Me				10								
	…												
Processes	Backup												
	…												

- Risk rationales: develop a table similar to the following to explain the top ten risks.

Risk	Asset	Threat to Asset	Asset vulnerable to	Likelihood	Impact	Risk
1	Bank Account	Software attacks	Malware, e.g. on PC due to poor game download processes	Moderate	High	High
2	…					

- Controls: develop a table similar to the following to identify and justify controls that you and/or your friend are willing and able to implement. Include at least 10 controls. Include at least one control for each NIST step. You will provide an implementation report (with evidence) in Assignment 3.

Risk	Control	NIST step	Explanation/Rationale	Responsible
3	Open bank account that supports MFA to reduce	Protect	Current bank account is vulnerable to password attacks due to a lack of MFA. MFA protects the bank account by reducing unauthorised access.	Friend
6	Develop a PowerShell script to check the integrity of backup files.	Detect	Although backups are automatic (File History), there is no integrity checking. Script will detect integrity violations.	Me
…				

Task Submission

Include a link to your private repository in your Word document. Commit your Word document to your private Git repository. Submit your Word document and your video to the unit website.

Task 3 Bash

Due: Week 7

In this task you will develop a Bash script. Create a Bash script file called ass2.sh. Your script should be designed to run on the Kali virtual image. The script should collect information and produce a report identifying any issues with the system. Each step should remain silent if successful and only produce output if there is an issue that needs to be addressed.

- Create a software inventory that is a list of approved packages. Check that only approved packages are installed. Submit your list of approved programs. Remove one of the approved programs from your list. Submit an example run of your script detecting an unapproved package.
- Create a list of services approved for running. Check that only approved services are running. Submit your list of approved services. Remove one of the approved services from your list. Submit an example run of your script detecting an unapproved service.
- Create a list of authorised local network ports (0..1023). Check that there no other open ports on the localhost. Remove one of the approved ports from your list. Submit an example run of your script detecting the unapproved port.
- Scrape a Bing map of your suburb and save it to a file. You should hardcode a street in your suburb. You should use the Bing map API to obtain your latitude and longitude coordinate and then use that coordinate to obtain a map. Save the map to a file.

- Create a list of approved MAC addresses. Perform an ARP sweep of 172.16.1.1 .. 172.16.1.50 to check only approved NICs are connected. Remove one of the approved MACs from your list. Submit an example run of your script detecting the unapproved MAC.

Task Submission

Commit a single script called ass2.sh and a Word document containing your evidence to your private Git repository. Include a link to your private repository in your Word document. Submit ass2.sh and your Word document to the unit website.

Task 4 Python

Due: Week 11

In this task you will a Python script. Create a Python script called ass2.py. Your script should be designed to run on the Kali virtual image with the ms2 virtual machine also running. The script should collect information and produce a report identifying any issues with the system. Each step should remain silent if successful and only produce output if there is an issue that needs to be addressed.

- Scrape a Bing map of your street and save it to a file.

- Create a list of approved FTP servers on the local network. Check that all and only the approved FTP servers are running. Also report servers which are using vsftpd version 2.3.4.

- Create a list of approved HTTP servers on the local network. Check that only the approved HTTP servers are running. Log into each server and using apache2ctl, report any servers with no idle workers. If the server supports the URL /dvwa, then log into the DVWA server using admin:password. Check the login was successful.

Task Submission

Commit a single script called ass2.py and a Word document containing your evidence to your private Git repository. Include a link to your private repository in your Word document. Submit ass2.py and your Word document to the unit website.

Assessment Criteria

Each of the following marking criteria have equal weighting.

Criteria	Indicative of 100%	75%	50%	25%	0%
PowerShell functionality	All tasks implemented correctly←		→Uses deprecated functionality		
PowerShell code style, documentation & testing	Consistent, reasonable layout← Functions documented appropriately ← Example outputs demonstrate detection of issues ←	→ Scripts not available via Git repository	→Lack of modularity, e.g. poor or no functions		

Bash functionality	All tasks implemented correctly←		→Uses deprecated functionality		
Bash code style, documentation & testing	Consistent, reasonable layout← Functions documented appropriately ← Example outputs demonstrate detection of issues ←	→ Scripts not available via Git repository	→Lack of modularity, e.g. poor or no functions		
Python functionality	All tasks implemented correctly←		→Uses deprecated functionality		
Python code style, documentation & testing	Consistent, reasonable layout← Functions documented appropriately ← Example outputs demonstrate detection of issues ←	→ Scripts not available via Git repository	→Lack of modularity, e.g. poor or no functions		
Plan: Frame Risk Tolerance, Asset Inventory & Data flows	Excellent framing of risk tolerance defining, [in]tolerable risks & weekly time & money resources← Excellent modelling of ≥2 data flows of priority data assets←	→ Plan not available via Git repository → Incomplete data flows	→ Poor summary of risk tolerance, or data flows → Missing video		
Plan: Asset Inventory	≥5 priority data assets identified & classified ← ≥10 priority non-data assets identified ←		→ Missing video		
Plan: Risks	≥10 excellent, priority risks identified & rationalised with threats, vulnerabilities, likelihoods & impacts ← Video shows risk discussions ←		→ Misclassified threats, vulnerabilities or video	→ Missing rationales	
Plan: Controls	≥10 excellent controls that cover all NIST steps ← ≥1 reasonable, scriptable control ←	→ Controls do not cover all NIST steps	→ Missing video		

	Excellent explanations giving context & rationales ←				