Example Essential Eight Assessment Test Plan – Maturity Level Two

Mitigation Strategy	Control Description	Test ID	Test Description	Test Methodology	Test Findings
Application control	**Application control is implemented on workstations and internet-facing servers.**	ML2-AC-01	(Workstations & Internet-facing servers) A dedicated application control solution is implemented.	Check whether an in-built or third-party application control solution has been implemented.	
	Application control restricts the execution of executables, software libraries, scripts, installers, compiled HTML, HTML applications and control panel applets **to an organisation-approved set**.	ML2-AC-02	(Workstations & Internet-facing servers) The system is only able to execute approved executables.	Attempt to execute an unapproved, benign executable. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible policy bypass locations when performing this test. E8MVT will attempt to run an exe file within the local AppData temp directory. ACVT will recursively copy an exe file to the file system and attempt to execute it.	
		ML2-AC-03	(Workstations & Internet-facing servers) The system is only able to execute approved software libraries.	Attempt to execute an unapproved, benign software library. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible policy bypass locations when performing this test. E8MVT will attempt to run a dll file within the local AppData temp directory. ACVT will recursively copy a dll file to the file system and attempt to execute it.	
		ML2-AC-04	(Workstations & Internet-facing servers) The system is only able to execute approved scripts.	Attempt to execute an unapproved, benign scripts. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible policy bypass locations when performing this test. E8MVT will attempt to run multiple script files within the local AppData temp directory. ACVT will recursively copy multiple script files to the file system and attempt to execute them.	
		ML2-AC-05	(Workstations & Internet-facing servers) The system is only able to execute approved installers.	Attempt to execute an unapproved benign installer. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible bypass locations when performing this test. E8MVT will attempt to run a MSI file that will install a text file into the local AppData temp directory.	
		ML2-AC-06	(Workstations & Internet-facing servers) The system is only able to execute approved compiled HTML files.	Attempt to execute an unapproved benign compiled HTML file. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible bypass locations when performing this test.	

				The tester should first test that HH.exe is allowed to execute. If HH.exe is unable to execute than further testing may not be required. E8MVT will attempt to run a compiled HTML file within the local AppData temp directory.	
		ML2-AC-07	(Workstations & Internet-facing servers) The system is only able to execute approved HTML applications.	Attempt to execute an unapproved, benign HTML applications. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible bypass locations when performing this test. The tester should first test that MSHTA.exe is allowed to execute. If MSHTA.exe is unable to execute than further testing may not be required. E8MVT will attempt to run an HTML Application file within the local AppData temp directory.	
		ML2-AC-08	(Workstations & Internet-facing servers) The system is only able to execute an approved control panel applets.	Attempt to execute an unapproved benign control panel applet. Depending on the policy, there may be directories or specific versions of software that may be allowed to run, but might not be organisationally approved. The tester will have to consider possible bypass locations when performing this test. The tester should first test that CONTROL.exe is allowed to execute. If CONTROL.exe is unable to execute than further testing may not be required. E8MVT will attempt to run a Control Panel Applet file within the local AppData temp directory.	
	Allowed and blocked execution events on workstations and internet-facing servers are logged.	ML2-AC-09	(Workstations & Internet-facing servers) The system is logging the application control product when it allows and blocks execution.	Verify event logs contain required data (does not require central storage). Ensure all systems are logging.	
Patch applications	A vulnerability scanner is used at least **weekly** to identify missing patches or updates for vulnerabilities in office productivity suites, web browsers and their extensions, email clients, PDF software, and security products.	ML2-PA-01	A vulnerability scanner is run and reviewed at least weekly to scan the organisation's office productivity suites, web browsers, email clients, PDF software and security products.	Confirm that a vulnerability scanner is in place, and it is configured to scan the organisation's applications listed, typically requiring a credentialed scan. Confirm that reports from the vulnerability scanner are reviewed by the responsible staff weekly, and that identified issues have been observed and actioned.	
	A vulnerability scanner is used at least fortnightly to identify missing patches or updates for vulnerabilities in other applications.	ML2-PA-02	A vulnerability scanner is run and reviewed at least fortnightly to scan the organisation's other applications.	Confirm that a vulnerability scanner is in place, and it is configured to scan the organisation's other applications, typically requiring a credentialed scan. Confirm that reports from the vulnerability scanner are reviewed by the responsible staff fortnightly, and that identified issues have been observed and actioned.	

	Patches, updates or other vendor mitigations for vulnerabilities in office productivity suites, web browsers and their extensions, email clients, PDF software, and security products are applied within **two weeks** of release.	ML2-PA-03	The organisation has an effective process for patching office productivity suites, web browsers, email clients, PDF software and security products within two weeks.	Confirm the existence of a list of applications, and where the applications are installed. Ensure a process for identifying vulnerabilities for software in the list is consistently followed. Request evidence of the patching of these applications within two weeks.	
		ML2-PA-04	Office productivity suites, web browsers, email clients, PDF software and security products do not have vulnerabilities older than two weeks.	Use a vulnerability scanner to identify the listed applications within the organisation's environment, and check that they have been patched against a known exploit. Check the date the application was updated and compare to the date the patch was released. Ensure that the gap between is not greater than two weeks. E8MVT will perform basic checks of some Microsoft Office applications based on version numbers and file modification dates to determine if the software has been updated recently.	
	Patches, updates or other vendor mitigations for vulnerabilities in other applications are applied within one month of release.	ML2-PA-05	Other applications that have a vulnerability are patched or mitigated within one month.	Use a vulnerability scanner to identify vulnerable applications within the organisation's environment, and check that they have been patched against a known exploit. Check the date the application was updated and compare to the date the patch was released. Ensure that the gap between is not greater than one month.	
Configure Microsoft Office macro settings	**Microsoft Office macros are blocked from making Win32 API calls.**	ML2-OM-01	Microsoft Office macros in Microsoft Office files are unable to make Win32 API calls.	Open a file that contains a Microsoft Office macro that makes a Win32 API call. Do this for all installed Microsoft Office applications. E8MVT can assist with this test.	
	Allowed and blocked Microsoft Office macro execution events are logged.	ML2-OM-02	Allowed execution of a Microsoft Office macro within a Microsoft Office file is logged.	Check that TrustCenter logging is enabled by checking the Enable Logging registry key at *HKCU:\Software\Microsoft\Office\16.0\Common\TrustCenter*. *Get-ItemProperty -Path "HKCU:\Software\Microsoft\Office\<version>\Common\TrustCenter\" | Select-Object -Property EnableLogging* Example: *Get-ItemProperty -Path "HKCU:\Software\Microsoft\Office\16.0\Common\TrustCenter\" | Select-Object -Property EnableLogging* Request evidence of event logs for allowed Microsoft Office macro execution events.	
		ML2-OM-03	Blocked execution of a Microsoft Office macro within a Microsoft Office file is logged.	Check that TrustCenter logging is enabled by checking the EnableLogging registry key at *HKCU:\Software\Microsoft\Office\16.0\Common\TrustCenter*. *Get-ItemProperty -Path "HKCU:\Software\Microsoft\Office\<version>\Common\TrustCenter\" | Select-Object -Property EnableLogging* Example: *Get-ItemProperty -Path "HKCU:\Software\Microsoft\Office\16.0\Common\TrustCenter\" | Select-Object -Property EnableLogging* Request evidence of event logs for blocked Microsoft Office macro execution events. Microsoft Office macros blocked due to AV can be found in the Event Viewer.	

User application hardening	**ASD or vendor hardening guidance for web browsers is implemented.**	ML2-AH-01	ASD guidance for hardening Microsoft Edge is implemented. OR Microsoft guidance for hardening Microsoft Edge is implemented.	Use the Microsoft Policy Analyzer to validate the effective state of the system against the Microsoft Edge security baseline.	
		ML2-AH-02	The Google guidance for hardening Google Chrome is implemented.	Determine if Google Chrome is configured via group policy settings and if the configured settings are in line with the *Chrome Browser Enterprise Security Configuration Guide* provided by Google at https://support.google.com/chrome/a/answer/9710898?hl=en.	
	Microsoft Office is blocked from creating child processes.	ML2-AH-03	Microsoft Office files cannot create child processes.	Open a file that contains a Microsoft Office macro that will create a child process. Confirm it is unable to do this. Check the ASR rule 'd4f940ab-401b-4efc-aadc-ad5f3c50688a' is configured in block mode, or another solution is in place to prevent creation of child processes. Running E8MVT will confirm if the ASR rule to prevent creation of child processes is enabled, or if child process creation has been blocked through a PowerShell command. Running E8MVT will execute a test that opens a file containing a Microsoft Office macro that creates a child process. *$ASR_Rules = Get-MPPreference \| Select -ExpandProperty AttackSurfaceReductionRules_Ids* *$match = $false* *Foreach($rules in $ASR_Rules) {If ($rules -match "d4f940ab-401b-4efc-aadc-ad5f3c50688a") {$match = $true}}* *If($match -eq $true) {Write-Output("Block all Office applications from creating child processes (d4f940ab-401b-4efc-aadc-ad5f3c50688a) is enabled")} else {Write-Output("Block all Office applications from creating child processes (d4f940ab-401b-4efc-aadc-ad5f3c50688a) is not present or disabled")}*	
	Microsoft Office is blocked from creating executable content.	ML2-AH-04	Microsoft Office files cannot create executable content.	Open a file that contains a Microsoft Office macro that will create executable content. Confirm it is unable to do this. Check the ASR rule '3b576869-a4ec-4529-8536-b80a7769e899' is configured in block mode, or another solution is in place to prevent creation of executable content. Running E8MVT will confirm if the ASR rule to creation of executable content is enabled. Running E8MVT will execute a test that opens a file containing a Microsoft Office macro that creates executable content. *$ASR_Rules = Get-MPPreference \| Select -ExpandProperty AttackSurfaceReductionRules_Ids* *$match = $false* *Foreach($rules in $ASR_Rules) {If ($rules -match "3b576869-a4ec-4529-8536-b80a7769e899") {$match = $true}}* *If($match -eq $true) {Write-Output("Block Office applications from creating executable content (3b576869-a4ec-4529-8536-b80a7769e899) is enabled")} else {Write-Output("Block Office applications from creating executable content (3b576869-a4ec-4529-8536-b80a7769e899) is not present or disabled")}*	

	Microsoft Office is blocked from injecting code into other processes.	ML2-AH-05	Microsoft Office files cannot inject code into other processes.	Open a file that contains a Microsoft Office macro that will inject code into another process. Confirm it is unable to do this. Check the ASR rule '75668C1F-73B5-4CF0-BB93-3ECF5CB7CC84' is configured in block mode, or another solution is in place to prevent code injection. Running E8MVT will confirm if the ASR rule to prevent injection of code into other processes is enabled. Running E8MVT will execute a test that opens a file containing a Microsoft Office macro that will attempt to inject code into the explorer.exe process. *$ASR_Rules = Get-MPPreference \| Select -ExpandProperty AttackSurfaceReductionRules_Ids* *$match = $false* *Foreach($rules in $ASR_Rules) {If ($rules -match "75668C1F-73B5-4CF0-BB93-3ECF5CB7CC84") {$match = $true}}* *If($match -eq $true) {Write-Output("Block Office applications from injecting code into other processes (75668C1F-73B5-4CF0-BB93-3ECF5CB7CC84) is enabled")} else {Write-Output("Block Office applications from injecting code into other processes (75668C1F-73B5-4CF0-BB93-3ECF5CB7CC84) is not present or disabled")}*	
	Microsoft Office is configured to prevent activation of OLE packages.	ML2-AH-06	Microsoft Office files do not execute OLE packages.	Open a file that contains an OLE object. Check the PackagerPrompt registry key within the Trust Center settings is set to 2. E8MVT will check the required registry key. *Get-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\office\<version>\<application>\security\" | Select-Object -Property PackagerPrompt* Example: *Get-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\office\16.0\excel\security\" | Select-Object -Property PackagerPrompt*	
	ASD or vendor hardening guidance for Microsoft Office is implemented.	ML2-AH-07	ASD guidance for hardening Microsoft Office is implemented. OR Microsoft guidance for hardening Microsoft Office is implemented.	Use the Microsoft Policy Analyzer to validate the effective state of the system against the Microsoft Office security baseline.	
	Microsoft Office security settings cannot be changed by users.	ML2-AH-08	Microsoft Office security setting are unable to be modified by a standard user account.	Attempt to modify security settings in Microsoft Office. Check that the vbawarnings registry key is configured via policy and that a user is unable to change the Microsoft Office macro settings within the Trust Center options. E8MVT will check the required registry key. *Get-ItemProperty -Path "HKCU:\SOFTWARE\Policies\Microsoft\office\<version>\<application>\security\" | Select-Object -Property vbawarnings* Example: *Get-ItemProperty -Path "HKCU:\SOFTWARE\Policies\Microsoft\office\16.0\excel\security\" | Select-Object -Property vbawarnings*	

	PDF software is blocked from creating child processes.	ML2-AH-09	PDF software cannot create child processes.	Adobe Reader can be tested by opening the application, selecting Open from the File menu, selecting 'All Files (*.*)' from the dropdown menu in the corner, browsing to system32 folder, find calc.exe. Right click and select Open. Check ASR rule 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c is enabled. Can also run *Get-ProcessMitigation -Name acrobat.exe).ChildProcess.DisallowChildProcessCreation* to check if it has been disabled this way. E8MVT will check that the Adobe child process creation ASR rule is enabled. *$ASR_Rules = Get-MPPreference \| Select -ExpandProperty AttackSurfaceReductionRules_Ids* *$match = $false* *Foreach($rules in $ASR_Rules) {If ($rules -match "7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c") {$match = $true}}* *If($match = $true) {Write-Output("Block Adobe Reader from creating child processes (7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c) is enabled")} else {Write-Output("Block Adobe Reader from creating child processes (7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c) is not present or disabled")}*	
	ASD or vendor hardening guidance for PDF software is implemented.	ML2-AH-10	Vendor guidance for hardening PDF software is implemented.	Determine the PDF software in use, and if the vendor provides hardening guidance for the product. Follow the guidance to determine if the product has been hardened. Adobe Acrobat and Adobe Reader hardening guidance can be found at https://www.adobe.com/devnet-docs/acrobatetk/tools/AppSec/index.html.	
	PDF software security settings cannot be changed by users.	ML2-AH-11	PDF software security settings are unable to be modified by a standard user account.	Attempt to modify security settings within all allowed PDF readers.	
	Blocked PowerShell script execution events are logged.	ML2-AH-12	PowerShell scripts that have been blocked are logged.	Verify event logs contain required data (does not require central storage). Note, if application control is used to restrict the execution of scripts, PowerShell script execution events may already be captured. Ensure all systems are logging.	
Restrict administrative privileges	**Privileged access to systems and applications is automatically disabled after 12 months unless revalidated.**	ML2-RA-01	A process for disabling known privileged accounts exists and is enforced. Users are made aware of this requirement when being provisioned with a privileged account.	Review documented process to disable privileged access after 12 months. Review evidence, such as support tickets, emails, logs, or the automated disabling procedure, to confirm accounts are disabled after 12 months unless revalidated.	
		ML2-RA-02	There are no privileged accounts that have an Active Directory expiry date that is greater than 12 months or do not have an expiry date.	Query Active Directory using PowerShell commands or tools such as ADRecon to identify accounts with distant or no expiry dates. The following PowerShell command returns privileged accounts with no account expiry set. *Get-ADUser -Filter {(admincount -eq 1) -and (enabled -eq $true)} -Properties AccountExpirationDate \| Where-Object {$_.AccountExpirationDate -like ""} | Select @{n='Username'; e={$_.SamAccountName}}, @{n='Account Expiration Date'; e={$_.AccountExpirationDate}}, @{n='Enabled'; e={$_.Enabled}}*	

				The following PowerShell command returns any privileged accounts that have an expiry date greater than 12 months. *Get-ADUser -Filter {(admincount -eq 1) -and (enabled -eq $true)} -Properties AccountExpirationDate \| Where-Object {$_.AccountExpirationDate -gt (Get-Date).AddMonths(12)} | Select @{n='Username'; e={$_.SamAccountName}}, @{n='Account Expiration Date'; e={$_.AccountExpirationDate}}, @{n='Enabled'; e={$_.Enabled}}*	
	Privileged access to systems and applications is automatically disabled after 45 days of inactivity.	ML2-RA-03	A process for disabling privileged accounts that have not been used for 45 days exists and is enforced by the entity. Evidence exists for the usage of the 45 days inactive disabling process, including support tickets or administrative logs that show accounts were disabled.	Review documented process to disable privileged access after 45 days of inactivity. Review evidence, such as support tickets, automated disabling procedure to confirm accounts have been disabled.	
		ML2-RA-04	There are no enabled privileged accounts that have a lastlogondate that is greater than 45 days.	Query Active Directory using PowerShell commands or tools such as ADRecon to identify enabled privileged accounts with a 'lastlogondate' greater than 45 days. *Get-ADUser -Filter {(admincount -eq 1) -and (enabled -eq $true)} -Properties LastLogonDate \| Where-Object {$_.LastLogonDate -lt (Get-Date).AddDays(-45) -and$_.LastLogonDate -ne $null} \| Select @{n='Username'; e={$_.samaccountname}}, @{n='Last Logon Date'; e={$_.LastLogonDate}}, @{n='Enabled'; e={$_.enabled}}*	
	Privileged operating environments are not virtualised within unprivileged operating environments.	ML2-RA-05	Where a privileged environment is virtualised, the virtualised image is not located in an unprivileged environment. This includes virtual machines on a standard unprivileged SOE.	Confirm with the entity where the privileged environment infrastructure is hosted. Look for privileged environments on unprivileged virtual hosts and SOE workstations.	
	Administrative activities are conducted through jump servers.	ML2-RA-06	Servers are configured to not allow remote access traffic or connections from systems that are not jump servers.	Attempt to connect to servers or administrator-only systems from an unprivileged environment. Verify Firewall configuration.	
	Credentials for local administrator accounts and service accounts are long, unique, unpredictable and managed.	ML2-RA-07	The Microsoft Local Administrator Password Solution (LAPS) or a similar solution is implemented on Windows workstations and servers.	Run the following PowerShell commands to retrieve the number of computers with LAPS and compare with the number of computers in Active Directory. Run the following PowerShell command to get number of computers with LAPS. *Get-ADComputer -Filter {ms-Mcs-AdmPwdExpirationTime -like "*"} -Properties ms-Mcs-AdmPwdExpirationTime | measure* Run the following PowerShell command to get the number of enabled computers in Active Directory. *Get-ADComputer -Filter {Enabled -eq $true} | measure*	
		ML2-RA-08	Services account passwords are generated to be long, unique and unpredictable. Service account passwords are stored in a secure location, such as a password manager or a Privileged Access Management solution.	Observe evidence of a password management or privileged access management solution in use for managing service account passwords. Ensure generated passwords are unique, unpredictable, and have a minimum length requirement. Look for accounts with identical passwords.	

				Confirm how passwords are generated for local accounts, and that they are managed. If using LAPS for local accounts, check the following group policy setting *Computer Configuration/Administrative Templates/LAPS/Password Settings*.	
		ML2-RA-09	Passwords should be changed at least once every 12 months.	Query Active Directory using PowerShell commands or tools such as ADRecon to identify service accounts with passwords last set more than 12 months ago. Run the following PowerShell command to get service accounts with passwords older than 12 months. Replace SVC_* with service account naming convention. *$PassLastSetTimeFrame = (Get-Date).AddMonths(-12)* *Get-ADUser -Filter "enabled -eq 'true' -and SamAccountName -like 'SVC_*'" -Properties pwdlastset \| Where-Object{$_.pwdlastset -like '0' -or ([datetime]::FromFileTime($_.pwdLastSet) -lt $PassLastSetTimeFrame)} \| Select-Object SAMAccountName, @{name ="pwdLastSet"; expression={([datetime]::FromFileTime($_.pwdLastSet))}}*	
	Privileged access events are logged.	ML2-RA-10	Successful and failed logins of privileged accounts are logged.	Event logs need to be retained/backed up for a minimum period, so they are available if required. Verify the existence of the following event logs. Event ID 4672 is created when an account with special privileges successfully logs in. Event ID 4625 is created when a logon request fails.	
	Privileged account and group management events are logged.	ML2-RA-11	Changes made to privileged accounts and groups within Active Directory are logged.	Event logs need to be retained/backed up for a minimum period, so they are available if required. Verify the existence of the following event logs. Event ID 4738 is created when a user account is modified in Active Directory. Event ID 4728 is created when a member is added to an Active Directory Security Group. Event ID 4729 is created when a member is removed from an Active Directory Security Group. Event ID 4737 is created when a change is made to an Active Directory Security Group.	
Patch operating systems	A vulnerability scanner is used at least **weekly** to identify missing patches or updates for vulnerabilities in operating systems of workstations, servers and network devices.	ML2-PO-01	A vulnerability scanner is run and reviewed at least weekly to scan the organisation's operating systems.	Confirm that a vulnerability scanner is in place, and it is configured to scan the organisation's operating systems, typically requiring a credentialed scan. Confirm that reports from the vulnerability scanner are reviewed by the responsible staff weekly, and that identified issues have been observed and actioned.	
	Patches, updates or other vendor mitigations for vulnerabilities in operating	ML2-PO-02	The organisation has an effective process for patching operating systems within two weeks.	Confirm the existence of a list of managed operating systems, and where they are located. Ensure a process for identifying vulnerabilities for operating systems in the list is consistently	

	systems of workstations, servers and network devices are applied within **two weeks** of release.			followed. Request evidence of the patching of these systems within two weeks.	
		ML2-PO-03	Operating systems that have a vulnerability are patched or mitigated within two weeks.	Use vulnerability management solution to perform a patch audit of all systems. Retrieve the update history of the system, noting the release date of the patch and the date it was installed. Look for differences greater than two weeks.	
Multi-factor authentication	**Multi-factor authentication is used to authenticate privileged users of systems.**	ML2-MF-01	A privileged user who is performing administrative activities is required to respond to an MFA challenge at some point in the authentication lifecycle. This can be implemented when authenticating to a machine (such as a jump server) or when attempting to raise privileges. The organisation has a list of systems that have privileged users or support privileged functions.	Verify a privileged user is presented with a MFA challenge when authenticating to a machine or attempting to raise privileges. Confirm the organisation has a list of privileged systems and is regularly updated.	
	Multi-factor authentication uses either: something users have and something users know, or something users have that is unlocked by something users know or are.	ML2-MF-02	The organisation requires that internet-facing services use multi-factor authentication that uses either: something users have and something users know, or something users have that is unlocked by something users know or are.	Verify MFA for internet-facing services uses either: Something users have (e.g. look-up secrets, out-of-band devices, single-factor OTP devices, single-factor cryptographic software or single-factor cryptographic devices) AND something users know (e.g. memorised secrets) OR Something users have that is unlocked by something users know or are (e.g. multi-factor OTP devices, multi-factor cryptographic software and multi-factor cryptographic devices)	
		ML2-MF-03	The organisation requires that privileged users utilise multi-factor authentication that uses either: something users have and something users know, or something users have that is unlocked by something users know or are.	Verify MFA for internet-facing services uses either: Something users have (e.g. look-up secrets, out-of-band devices, single-factor OTP devices, single-factor cryptographic software or single-factor cryptographic devices) AND something users know (e.g. memorised secrets) OR Something users have that is unlocked by something users know or are (e.g. multi-factor OTP devices, multi-factor cryptographic software and multi-factor cryptographic devices)	
	Successful and unsuccessful multi-factor authentication events are logged.	ML2-MF-04	The organisation's internet-facing systems log successful MFA attempts.	Verify successful MFA events are logged for organisations internet-facing systems.	
		ML2-MF-05	Administrative access connections log successful MFA attempts.	Verify successful MFA events are logged for administrative access.	
		ML2-MF-06	The organisation's internet-facing systems log unsuccessful MFA attempts.	Verify unsuccessful MFA events are logged for organisations internet-facing systems.	
		ML2-MF-07	Administrative access connections log unsuccessful MFA attempts.	Verify unsuccessful MFA events are logged for administrative access.	

Regular backups	**Privileged accounts (excluding backup administrator accounts) cannot access backups belonging to other accounts.**	ML2-RB-01	Privileged users (excluding backup administrator accounts) are unable to access backups that do not belong to them.	Verify access controls restrict the access of backups to the owner of the backup and backup administrator accounts.	
	Privileged accounts (excluding backup administrator accounts) are prevented from modifying and deleting backups.	ML2-RB-02	Privileged users (excluding backup administrator accounts) are unable to modify and delete backups.	Verify access controls restrict the modification and deletion of backups to backup administrator accounts.	