[Implementation of security functionalities of the system]

You can only implement a **limited number** of security functionalities:

- Maximum **2** functionalities will be implemented **for sure** (*must-be*).
- Maximum 2 functionalities will be **very likely** implemented (*should-be*).
- Maximum **2** functionalities will be **maybe** implemented (*could*).

Clearly **justify** your choices in a sentence or two.

- 1. click once in cell
- 2. click on arrow for drop-down menu

1 Verify that the chosen access control solution is flexible enough to meet the applications needs. Verify the application uses a single and well-vetted access control applications needs. Verify the application uses a single and well-vetted access control mechanism for a oxid copy and paste or insecure alternative paths. Verify interpresonation resistance against phishing, such as the use of multiflator authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates. Verify preplay resistance through the mandated use of One-time Passwords (OTP) devices, cryptographic authenticators, or lookup codes. Verify intent to authenticate by requiring the entry of an OTP token or user-initiated action such as a button press on a FIDO hardware key. Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage. Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device. Verify bytical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. Verify that users are able to view and (having re-entered login credentials) logo out of any or all currently active sessions and devices. Verify the application allows users to revoke Outh tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or		Security functionality	Implementation	Rationale	
pass through this single mechanism to avoid copy and paste or insecure alternative paths. Verify impersonation resistance against phishing, such as the use of multifactor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates. Verify replay resistance through the mandated use of One-time Passwords (OTP) devices, cryptographic authenticators, or lookup codes. Verify thent to authenticate by requiring the entry of an OTP token or user-initiated action such as a button press on a FIDO hardware key. Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage. Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device. Verify sylvical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of applications and past fraud. Verify that the applications to enforce anti-fraud controls as per the risk of applications to enforce anti-	1	•			
factor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates. 4 Verify replay resistance through the mandated use of One-time Passwords (OTP) devices, cryptographic authenticators, or lookup codes. 5 Verify intent to authenticate by requiring the entry of an OTP token or user-initiated action such as a button press on a FIDO hardware key. Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage. Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device. Verify hysical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated loign (if present), and any relying parties. 10 Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. 11 Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. 12 Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. 13 And failed authentication fologes security relevant events including successful and failed authentication fologes security relevant events including successful and failed authentication events, access contol failures, deserialization failures and input validation failures. 14 Verify that a "last resort" error handler is defined which will catch all unhandled exceptions.	2	mechanism for accessing protected data and resources. All requests must pass through this single mechanism to avoid copy and paste or insecure			
COTP) devices, cryptographic authenticators, or lookup codes.	3	factor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side			
Verify intent to authenticate by requiring the entry of an OTP token or user-initiated action such as a button press on a FIDO hardware key. Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage. Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device. Verify physical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions.	4				
protected, such as by using a hardware security module or secure operating system based key storage. Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device. Verify physical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that acception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. Must-be Should-be Should-be	5	Verify intent to authenticate by requiring the entry of an OTP token or user-initiated action such as a button press on a FIDO hardware key.			
7 validity period, it is logged and rejected with secure notifications being sent to the holder of the device. 8 Verify physical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. 9 Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. 10 Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. 11 Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. 12 Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. 13 Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. 14 Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. 15 Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. 16 Nust-be 0 17 Should-be 0	6	protected, such as by using a hardware security module or secure operating			
sother loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location. Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be Should-be O	7	validity period, it is logged and rejected with secure notifications being sent			
sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. 13 Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. 14 Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be Should-be O	8	other loss. Ensure that revocation is immediately effective across logged in			
Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be O Should-be O	9	sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated			
11 Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications. Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be O Should-be O	10	Verify that users are able to view and (having re-entered login credentials)			
Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be O Should-be O	11	Verify the application allows users to revoke OAuth tokens that form trust			
Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be Should-be 0	12	adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the			
codebase to account for expected and unexpected error conditions. Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. Must-be Should-be 0	13	and failed authentication events, access control failures, deserialization			
Must-be 0 Should-be 0	14	· · · · · · · · · · · · · · · · · · ·			
Should-be 0	15				
	Must-be 0				
Could-be 0		Should-be	0		
		Could-be	0		