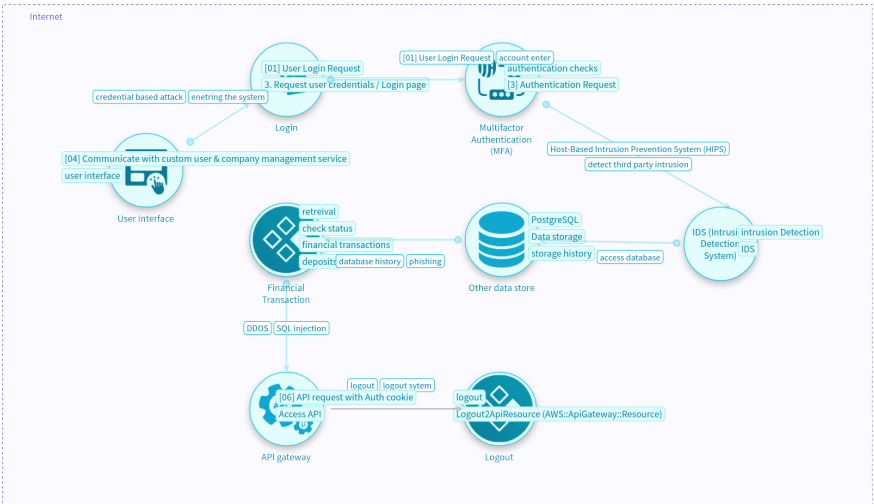


Threat-Modeling_LAB_03

PCI-DSS-v3.2.1 - Compliance report

Tue Feb 11 2025 11:46:13 GMT+0000 (Coordinated Universal Time)

Project description: No description
Unique ID: threat-modeling_lab_03
Owner: [ayela israr]
Workflow state: Draft
Tags: No tags



Compliance summary

See the countermeasure state data categorized by standards:

PCI-DSS-v3.2.1: 1.1.1			PCI-DSS-v3.2.1: 1.1.2			PCI-DSS-v3.2.1: 1.1.3		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	1
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚪ Non-compliant	<div><div></div></div>	1	⚪ Non-compliant	<div><div></div></div>	1	⚪ Non-compliant	<div><div></div></div>	1
PCI-DSS-v3.2.1: 1.1.5			PCI-DSS-v3.2.1: 10.6.1			PCI-DSS-v3.2.1: 10.6.2		
✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	2	✔ Implemented	<div><div></div></div>	2
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	3	🔵 Recommended	<div><div></div></div>	3
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚪ Non-compliant	<div><div></div></div>	1	⚪ Non-compliant	<div><div></div></div>	3	⚪ Non-compliant	<div><div></div></div>	3
PCI-DSS-v3.2.1: 10.6.3			PCI-DSS-v3.2.1: 11.5.1			PCI-DSS-v3.2.1: 12.10.1		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	3	🔵 Recommended	<div><div></div></div>	3	🔵 Recommended	<div><div></div></div>	1
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚪ Non-compliant	<div><div></div></div>	3	⚪ Non-compliant	<div><div></div></div>	3	⚪ Non-compliant	<div><div></div></div>	1
PCI-DSS-v3.2.1: 12.10.5			PCI-DSS-v3.2.1: 12.10.6			PCI-DSS-v3.2.1: 12.3.10		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0	⚪ Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	3	🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	1
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚪ Non-compliant	<div><div></div></div>	3	⚪ Non-compliant	<div><div></div></div>	1	⚪ Non-compliant	<div><div></div></div>	1

PCI-DSS-v3.2.1: 12.3.8			PCI-DSS-v3.2.1: 12.3.9			PCI-DSS-v3.2.1: 12.5.2		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	3
❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	3
PCI-DSS-v3.2.1: 12.8.2			PCI-DSS-v3.2.1: 6.4.2			PCI-DSS-v3.2.1: 7.1.4		
✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	0
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	4	🔵 Recommended	<div><div></div></div>	2
❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	4	⚠ Non-compliant	<div><div></div></div>	2
PCI-DSS-v3.2.1: 8.1.5			PCI-DSS-v3.2.1: 8.2.1			PCI-DSS-v3.2.1: 8.2.2		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	0
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	2	🔵 Recommended	<div><div></div></div>	0	🔵 Recommended	<div><div></div></div>	2
❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	2	⚠ Non-compliant	<div><div></div></div>	0	⚠ Non-compliant	<div><div></div></div>	2
PCI-DSS-v3.2.1: 8.5.1			PCI-DSS-v3.2.1: 9.1.1			PCI-DSS-v3.2.1: 9.6.1		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	3	🔵 Recommended	<div><div></div></div>	0
❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	3	⚠ Non-compliant	<div><div></div></div>	0
PCI-DSS-v3.2.1: 9.9.3			PCI-DSS-v3.2.1: 1			PCI-DSS-v3.2.1: 1.1		
✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	3	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0	🛑 Not-applicable	<div><div></div></div>	0
🔵 Recommended	<div><div></div></div>	1	🔵 Recommended	<div><div></div></div>	2	🔵 Recommended	<div><div></div></div>	1
❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0	❌ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	2	⚠ Non-compliant	<div><div></div></div>	1

PCI-DSS-v3.2.1: 1.2			PCI-DSS-v3.2.1: 1.3			PCI-DSS-v3.2.1: 1.5		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0
🛡 Recommended	<div><div></div></div>	1	🛡 Recommended	<div><div></div></div>	1	🛡 Recommended	<div><div></div></div>	0
❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	0
PCI-DSS-v3.2.1: 2			PCI-DSS-v3.2.1: 2.1			PCI-DSS-v3.2.1: 2.2		
✔ Implemented	<div><div></div></div>	3	✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	1
🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0
🛡 Recommended	<div><div></div></div>	2	🛡 Recommended	<div><div></div></div>	1	🛡 Recommended	<div><div></div></div>	2
❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	2	⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	2
PCI-DSS-v3.2.1: 2.3			PCI-DSS-v3.2.1: 2.5			PCI-DSS-v3.2.1: 3.7		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0
🛡 Recommended	<div><div></div></div>	1	🛡 Recommended	<div><div></div></div>	0	🛡 Recommended	<div><div></div></div>	0
❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	0	⚠ Non-compliant	<div><div></div></div>	0
PCI-DSS-v3.2.1: 4			PCI-DSS-v3.2.1: 4.3			PCI-DSS-v3.2.1: 5		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	0
🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0
🛡 Recommended	<div><div></div></div>	0	🛡 Recommended	<div><div></div></div>	0	🛡 Recommended	<div><div></div></div>	1
❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	0	⚠ Non-compliant	<div><div></div></div>	0	⚠ Non-compliant	<div><div></div></div>	1
PCI-DSS-v3.2.1: 5.4			PCI-DSS-v3.2.1: 6.1			PCI-DSS-v3.2.1: 6.2		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	2
🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0	🔑 Required	<div><div></div></div>	0
❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0	❌ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0	⚡ Not-applicable	<div><div></div></div>	0
🛡 Recommended	<div><div></div></div>	0	🛡 Recommended	<div><div></div></div>	2	🛡 Recommended	<div><div></div></div>	2
❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0	❗ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	0	⚠ Non-compliant	<div><div></div></div>	2	⚠ Non-compliant	<div><div></div></div>	2

PCI-DSS-v3.2.1: 6.5			PCI-DSS-v3.2.1: 6.7			PCI-DSS-v3.2.1: 7.1		
✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0
🛡 Recommended	<div></div>	1	🛡 Recommended	<div></div>	1	🛡 Recommended	<div></div>	5
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚠ Non-compliant	<div></div>	1	⚠ Non-compliant	<div></div>	1	⚠ Non-compliant	<div></div>	5
PCI-DSS-v3.2.1: 7.2			PCI-DSS-v3.2.1: 7.3			PCI-DSS-v3.2.1: 8.1		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0
🛡 Recommended	<div></div>	5	🛡 Recommended	<div></div>	1	🛡 Recommended	<div></div>	2
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚠ Non-compliant	<div></div>	5	⚠ Non-compliant	<div></div>	1	⚠ Non-compliant	<div></div>	2
PCI-DSS-v3.2.1: 8.2			PCI-DSS-v3.2.1: 8.3			PCI-DSS-v3.2.1: 8.4		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0
🛡 Recommended	<div></div>	2	🛡 Recommended	<div></div>	2	🛡 Recommended	<div></div>	1
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚠ Non-compliant	<div></div>	2	⚠ Non-compliant	<div></div>	2	⚠ Non-compliant	<div></div>	1
PCI-DSS-v3.2.1: 8.5			PCI-DSS-v3.2.1: 8.6			PCI-DSS-v3.2.1: 8.7		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0
🛡 Recommended	<div></div>	1	🛡 Recommended	<div></div>	1	🛡 Recommended	<div></div>	4
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚠ Non-compliant	<div></div>	1	⚠ Non-compliant	<div></div>	1	⚠ Non-compliant	<div></div>	4
PCI-DSS-v3.2.1: 8.8			PCI-DSS-v3.2.1: 9.10			PCI-DSS-v3.2.1: 9.3		
✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0	⚡ Not-applicable	<div></div>	0
🛡 Recommended	<div></div>	0	🛡 Recommended	<div></div>	0	🛡 Recommended	<div></div>	4
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚠ Non-compliant	<div></div>	0	⚠ Non-compliant	<div></div>	0	⚠ Non-compliant	<div></div>	4

PCI-DSS-v3.2.1: 10.1			PCI-DSS-v3.2.1: 10.2			PCI-DSS-v3.2.1: 10.3		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0
🔵 Recommended	<div></div>	3	🔵 Recommended	<div></div>	2	🔵 Recommended	<div></div>	2
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚪ Non-compliant	<div></div>	3	⚪ Non-compliant	<div></div>	2	⚪ Non-compliant	<div></div>	2
PCI-DSS-v3.2.1: 10.4			PCI-DSS-v3.2.1: 10.5			PCI-DSS-v3.2.1: 10.6		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	2
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0
🔵 Recommended	<div></div>	2	🔵 Recommended	<div></div>	2	🔵 Recommended	<div></div>	6
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚪ Non-compliant	<div></div>	2	⚪ Non-compliant	<div></div>	2	⚪ Non-compliant	<div></div>	6
PCI-DSS-v3.2.1: 10.7			PCI-DSS-v3.2.1: 10.8			PCI-DSS-v3.2.1: 10.9		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	2	✔ Implemented	<div></div>	0
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0
🔵 Recommended	<div></div>	2	🔵 Recommended	<div></div>	3	🔵 Recommended	<div></div>	1
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚪ Non-compliant	<div></div>	2	⚪ Non-compliant	<div></div>	3	⚪ Non-compliant	<div></div>	1
PCI-DSS-v3.2.1: 11.1			PCI-DSS-v3.2.1: 11.2			PCI-DSS-v3.2.1: 11.3		
✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	1	✔ Implemented	<div></div>	2
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0
🔵 Recommended	<div></div>	3	🔵 Recommended	<div></div>	2	🔵 Recommended	<div></div>	3
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚪ Non-compliant	<div></div>	3	⚪ Non-compliant	<div></div>	2	⚪ Non-compliant	<div></div>	3
PCI-DSS-v3.2.1: 11.4			PCI-DSS-v3.2.1: 11.5			PCI-DSS-v3.2.1: 11.6		
✔ Implemented	<div></div>	2	✔ Implemented	<div></div>	0	✔ Implemented	<div></div>	1
⚠ Required	<div></div>	0	⚠ Required	<div></div>	0	⚠ Required	<div></div>	0
❌ Failed	<div></div>	0	❌ Failed	<div></div>	0	❌ Failed	<div></div>	0
🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0	🔍 Verified	<div></div>	0
⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0	⚪ Not-applicable	<div></div>	0
🔵 Recommended	<div></div>	3	🔵 Recommended	<div></div>	3	🔵 Recommended	<div></div>	0
❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0	❌ Rejected	<div></div>	0
⚪ Non-compliant	<div></div>	3	⚪ Non-compliant	<div></div>	3	⚪ Non-compliant	<div></div>	0

PCI-DSS-v3.2.1: 12.1			PCI-DSS-v3.2.1: 12.10			PCI-DSS-v3.2.1: 12.2		
✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	1
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛡 Not-applicable	<div><div></div></div>	0	🛡 Not-applicable	<div><div></div></div>	0	🛡 Not-applicable	<div><div></div></div>	0
💡 Recommended	<div><div></div></div>	0	💡 Recommended	<div><div></div></div>	3	💡 Recommended	<div><div></div></div>	2
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	0	⚠ Non-compliant	<div><div></div></div>	3	⚠ Non-compliant	<div><div></div></div>	2
PCI-DSS-v3.2.1: 12.3			PCI-DSS-v3.2.1: 12.4			PCI-DSS-v3.2.1: 12.5		
✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	0
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛡 Not-applicable	<div><div></div></div>	0	🛡 Not-applicable	<div><div></div></div>	0	🛡 Not-applicable	<div><div></div></div>	0
💡 Recommended	<div><div></div></div>	1	💡 Recommended	<div><div></div></div>	1	💡 Recommended	<div><div></div></div>	1
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	1
PCI-DSS-v3.2.1: 12.6			PCI-DSS-v3.2.1: 12.8			PCI-DSS-v3.2.1: 12.9		
✔ Implemented	<div><div></div></div>	0	✔ Implemented	<div><div></div></div>	1	✔ Implemented	<div><div></div></div>	0
⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0	⚠ Required	<div><div></div></div>	0
✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0	✖ Failed	<div><div></div></div>	0
🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0	🔍 Verified	<div><div></div></div>	0
🛡 Not-applicable	<div><div></div></div>	0	🛡 Not-applicable	<div><div></div></div>	0	🛡 Not-applicable	<div><div></div></div>	0
💡 Recommended	<div><div></div></div>	1	💡 Recommended	<div><div></div></div>	2	💡 Recommended	<div><div></div></div>	1
✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0	✖ Rejected	<div><div></div></div>	0
⚠ Non-compliant	<div><div></div></div>	1	⚠ Non-compliant	<div><div></div></div>	2	⚠ Non-compliant	<div><div></div></div>	1

Content menu

- PCI-DSS-v3.2.1: 1
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 2
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 4
 - Implemented
- PCI-DSS-v3.2.1: 5
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 6.4.2
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 2.2
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 7.1
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 7.2
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 8.7
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 9.3
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 10.6.1
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 10.6.2
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 11.4
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 10.6.3
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 11.5.1
 - Implemented
 - Non-compliant
 - Recommended
- PCI-DSS-v3.2.1: 12.10.5
 - Implemented

	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 12.5.2	Implemented	
	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 9.1.1	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.1	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.2	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.3	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.4	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.5	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.6	Implemented	
	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.7	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 10.8	Implemented	
	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 11.1	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 11.5	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 12.10	Implemented	
	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 12.8	Implemented	
	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 12.10.1	Implemented	
	Non-compliant	
	Recommended	
PCI-DSS-v3.2.1: 12.10.6	Implemented	
	Non-compliant	

Recommended
PCI-DSS-v3.2.1: 9.6.1
Implemented
PCI-DSS-v3.2.1: 1.5
Implemented
PCI-DSS-v3.2.1: 2.5
Implemented
PCI-DSS-v3.2.1: 3.7
Implemented
PCI-DSS-v3.2.1: 4.3
Implemented
PCI-DSS-v3.2.1: 5.4
Implemented
PCI-DSS-v3.2.1: 6.1
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 6.2
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 6.5
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 6.7
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 7.3
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.8
Implemented
PCI-DSS-v3.2.1: 9.10
Implemented
PCI-DSS-v3.2.1: 11.2
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 11.3
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 11.6
Implemented
PCI-DSS-v3.2.1: 12.1
Implemented
PCI-DSS-v3.2.1: 12.2
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 1.1.1

Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 1.1.2
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 1.1.3
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 12.3.10
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 12.3.8
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 12.3.9
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.1.5
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.2.1
Implemented
PCI-DSS-v3.2.1: 8.5.1
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 2.3
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.3
Implemented
Non-compliant
Recommended
PCI-DSS-v3.2.1: 7.1.4
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.2.2
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.1
Non-compliant
Recommended
PCI-DSS-v3.2.1: 8.2
Non-compliant
Recommended
PCI-DSS-v3.2.1: 10.9
Non-compliant
Recommended

PCI-DSS-v3.2.1: 1.1

- Implemented
- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 1.2

- Implemented
- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 1.3

- Implemented
- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 1.1.5

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 12.8.2

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 9.9.3

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 8.4

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 12.4

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 12.5

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 12.6

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 12.9

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 2.1

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 8.5

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 8.6

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 12.3

- Non-compliant
- Recommended

PCI-DSS-v3.2.1: 1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: Financial Transaction

Imp1. Use of a DDoS protection service for Payment System C-DDOS-PROTECTION Low

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Invalidate session after logout C-LOGOUT-V2-CNT-01 High Not tested

- State: Recommended
- Description:
 - Logout Trigger:** When a user initiates a logout, trigger a secure and verified logout process.
 - Session Identifier Destruction:** Immediately invalidate and destroy the user's session identifier associated with the logout action.
 - Clear Session Data:** Remove any user-specific data stored on the client side (e.g., cookies, local storage) related to the session upon logout.
 - Server-Side Session Termination:** Ensure that the server-side session is properly terminated, and any related data is wiped out to prevent further access.
 - Implement CSRF Protection:** Employ anti-CSRF (Cross-Site Request Forgery) measures to prevent malicious actors from forcing a user to perform unintended logouts.
 - Use Secure Communication:** Perform the logout operation over a secure communication channel (HTTPS) to prevent interception or tampering of the logout request.
 - Clear Authentication Tokens:** If the authentication process involves tokens (e.g., JWT), make sure to invalidate or blacklist these tokens during the logout process.
 - Double-Check Redirections:** Confirm that after logout, users are not redirected to sensitive pages without proper authentication. Avoid open redirects.
 - Audit Logout Events:** Log logout events for auditing purposes. Include details like user, timestamp, and reason for logout, if applicable.
 - Session Expiry Handling:** Configure session expiration policies to automatically invalidate sessions after a certain period of inactivity or elapsed time.
 - User Confirmation:** For sensitive actions like logout, consider implementing a confirmation mechanism to ensure that the user indeed intends to log out.
 - Clear Server-Side Caches:** If your system involves server-side caching, clear any cached user-related data to prevent potential information leakage.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

- State: Recommended
- Description:
 - Implement short session expiration times and require re-authentication for critical actions.**
 - Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.**
 - Monitor session activity and automatically log out users when unusual behavior is detected.**
 - Educate users on avoiding insecure networks and using VPNs for secure connections.**

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Invalidate session after logout C-LOGOUT-V2-CNT-01 High

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

PCI-DSS-v3.2.1: 2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: Financial Transaction

Imp1. Use of a DDoS protection service for Payment System C-DDOS-PROTECTION Low

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low


Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures


Rec1. Invalidate session after logout C-LOGOUT-V2-CNT-01 High Not tested

- State:  Recommended
- Description:
 - **Logout Trigger:** When a user initiates a logout, trigger a secure and verified logout process.
 - **Session Identifier Destruction:** Immediately invalidate and destroy the user's session identifier associated with the logout action.
 - **Clear Session Data:** Remove any user-specific data stored on the client side (e.g., cookies, local storage) related to the session upon logout.
 - **Server-Side Session Termination:** Ensure that the server-side session is properly terminated, and any related data is wiped out to prevent further access.
 - **Implement CSRF Protection:** Employ anti-CSRF (Cross-Site Request Forgery) measures to prevent malicious actors from forcing a user to perform unintended logouts.
 - **Use Secure Communication:** Perform the logout operation over a secure communication channel (HTTPS) to prevent interception or tampering of the logout request.
 - **Clear Authentication Tokens:** If the authentication process involves tokens (e.g., JWT), make sure to invalidate or blacklist these tokens during the logout process.
 - **Double-Check Redirects:** Confirm that after logout, users are not redirected to sensitive pages without proper authentication. Avoid open redirects.
 - **Audit Logout Events:** Log logout events for auditing purposes. Include details like user, timestamp, and reason for logout, if applicable.
 - **Session Expiry Handling:** Configure session expiration policies to automatically invalidate sessions after a certain period of inactivity or elapsed time.
 - **User Confirmation:** For sensitive actions like logout, consider implementing a confirmation mechanism to ensure that the user indeed intends to log out.
 - **Clear Server-Side Caches:** If your system involves server-side caching, clear any cached user-related data to prevent potential information leakage.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02  High  Not tested

- State:  Recommended
- Description:

Implement short session expiration times and require re-authentication for critical actions.
Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.
Monitor session activity and automatically log out users when unusual behavior is detected.
Educate users on avoiding insecure networks and using VPNs for secure connections.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Invalidate session after logout C-LOGOUT-V2-CNT-01  High

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02  High

PCI-DSS-v3.2.1: 4

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03  Low

PCI-DSS-v3.2.1: 5


Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03  High  Not tested

- State:  Recommended
- Description:
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

PCI-DSS-v3.2.1: 6.4.2

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA)

C-LOGIN-CM1

Medium

Not tested

State: Recommended

Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:
Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).
Integrate MFA into Your Login Flow:

- Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
- Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

- Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

- For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
- For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

- Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

- During registration or first login, prompt users to set up MFA.
- Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

- Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
- Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
- Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: Multifactor Authentication (MFA)

2 Recommended countermeasures

Rec1. Enhanced MFA resilience

C-MFA-01

High

Not tested

State: Recommended

Description:

Implement strict monitoring of MFA usage patterns to detect anomalies.
Ensure MFA tokens or devices are encrypted and tied to a specific user and device.
Regularly update and patch MFA software to fix vulnerabilities.
Educate users on recognizing phishing attempts and other social engineering attacks.

Rec2. Secure MFA fallbacks

C-MFA-03

High

Not tested

State: Recommended

Description:

Use secure fallback options, such as hardware tokens or biometrics, instead of SMS or email.
Limit the number of MFA bypass attempts and alert administrators on suspicious activity.
Ensure that all fallback mechanisms require re-authentication with a different factor than what was compromised.

Component: Financial Transaction

1 Recommended countermeasures

Rec1. Implement an access control system

C-ACCESS-CONTROL-SYSTEM

Medium

Not tested

State: Recommended

Description:

Implementing an access control system within a financial transaction environment requires careful planning and execution to ensure that sensitive data and operations are securely managed. Follow these actionable steps, oriented towards developers with or without previous security experience, to establish a robust access control system:
Define Security Requirements and Objectives:

- Identify the assets that require protection, including data, applications, and systems.
- Determine the security requirements for each asset based on its sensitivity and the potential impact of unauthorized access.

Classify Users and Assets:

- Classify users into roles based on their job functions and responsibilities within the organization.
- Assign sensitivity levels to assets (e.g., public, internal, confidential, highly confidential) to facilitate appropriate access control measures.

Select an Access Control Model:

- Choose an access control model that best fits your organizational needs and security objectives. Common models include:
 - Discretionary Access Control (DAC): Access is based on the identity of the users and/or the groups to which they belong.
 - Role-Based Access Control (RBAC): Access permissions are based on roles assigned to users. Users can have multiple roles, and roles can have multiple users.
 - Attribute-Based Access Control (ABAC): Access decisions are based on attributes (user attributes, resource attributes, and environment conditions).

Implement Authentication Mechanisms:

- Deploy strong authentication methods to verify the identity of users attempting to access the system. Consider multi-factor authentication (MFA) to add an extra layer of security.

Design and Implement Authorization Mechanisms:

- Once authenticated, users should be granted access only to the resources necessary for their roles. Implement permissions and access controls based on the selected access control model.

- Ensure that the principle of least privilege is applied, providing users with the minimum levels of access or permissions needed to perform their tasks.
- Develop Secure Administration Procedures:**
- Establish secure procedures for administering access controls, including how roles are assigned, changed, and revoked.
 - Ensure that changes to access control policies and configurations are logged and auditable.
- Ensure Scalability and Flexibility:**
- Design the access control system to be scalable and flexible, allowing for the addition of new users, roles, and resources without compromising security.
- Monitor and Audit Access:**
- Implement monitoring and logging to detect and record access attempts, both successful and unsuccessful. This helps in identifying potential security incidents and ensuring compliance with policies.
 - Regularly audit access controls and logs to ensure that only authorized users are accessing sensitive resources and that the access control system is functioning as intended.
- Educate Users and Enforce Policies:**
- Train users on the importance of security policies and the proper handling of sensitive information.
 - Enforce security policies and access controls consistently across the organization.
- Review and Update Access Controls:**
- Regularly review and update access controls in response to changes in the organizational structure, user roles, or security requirements.
 - Adjust policies and permissions as necessary to address new threats, vulnerabilities, and changes in regulatory requirements.
- Implementing an access control system is a dynamic process that requires ongoing attention and maintenance. By following these steps, you can create a secure and efficient environment that protects sensitive financial transactions and data from unauthorized access while facilitating legitimate business activities.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: Multifactor Authentication (MFA)

Rec1. Enhanced MFA resilience C-MFA-01 High

Rec2. Secure MFA fallbacks C-MFA-03 High

Component: Financial Transaction

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium

PCI-DSS-v3.2.1: 2.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

- State: Recommended
- Description:
Implement short session expiration times and require re-authentication for critical actions.
Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.
Monitor session activity and automatically log out users when unusual behavior is detected.
Educate users on avoiding insecure networks and using VPNs for secure connections.

Component: Financial Transaction

1 Recommended countermeasures

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium Not tested

- State: Recommended
- Description:
Implementing an access control system within a financial transaction environment requires careful planning and execution to ensure that sensitive data and operations are securely managed. Follow these actionable steps, oriented towards developers with or without previous security experience, to establish a robust access control system:
Define Security Requirements and Objectives:
 - Identify the assets that require protection, including data, applications, and systems.
 - Determine the security requirements for each asset based on its sensitivity and the potential impact of unauthorized access.**Classify Users and Assets:**
 - Classify users into roles based on their job functions and responsibilities within the organization.
 - Assign sensitivity levels to assets (e.g., public, internal, confidential, highly confidential) to facilitate appropriate access control measures.**Select an Access Control Model:**
 - Choose an access control model that best fits your organizational needs and security objectives. Common models include:
 - Discretionary Access Control (DAC): Access is based on the identity of the users and/or the groups to which they belong.
 - Role-Based Access Control (RBAC): Access permissions are based on roles assigned to users. Users can have multiple roles, and roles can have multiple users.
 - Attribute-Based Access Control (ABAC): Access decisions are based on attributes (user attributes, resource attributes, and environment conditions).**Implement Authentication Mechanisms:**
 - Deploy strong authentication methods to verify the identity of users attempting to access the system. Consider multi-factor authentication (MFA) to add an extra layer of security.**Design and Implement Authorization Mechanisms:**
 - Once authenticated, users should be granted access only to the resources necessary for their roles. Implement permissions and access controls based on the selected access control model.
 - Ensure that the principle of least privilege is applied, providing users with the minimum levels of access or permissions needed to perform their tasks.

Develop Secure Administration Procedures:

- Establish secure procedures for administering access controls, including how roles are assigned, changed, and revoked.
- Ensure that changes to access control policies and configurations are logged and auditable.

Ensure Scalability and Flexibility:

- Design the access control system to be scalable and flexible, allowing for the addition of new users, roles, and resources without compromising security.

Monitor and Audit Access:

- Implement monitoring and logging to detect and record access attempts, both successful and unsuccessful. This helps in identifying potential security incidents and ensuring compliance with policies.
- Regularly audit access controls and logs to ensure that only authorized users are accessing sensitive resources and that the access control system is functioning as intended.

Educate Users and Enforce Policies:

- Train users on the importance of security policies and the proper handling of sensitive information.
- Enforce security policies and access controls consistently across the organization.

Review and Update Access Controls:

- Regularly review and update access controls in response to changes in the organizational structure, user roles, or security requirements.
- Adjust policies and permissions as necessary to address new threats, vulnerabilities, and changes in regulatory requirements.

Implementing an access control system is a dynamic process that requires ongoing attention and maintenance. By following these steps, you can create a secure and efficient environment that protects sensitive financial transactions and data from unauthorized access while facilitating legitimate business activities.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

Component: Financial Transaction

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium

PCI-DSS-v3.2.1: 7.1

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: Multifactor Authentication (MFA)

3 Recommended countermeasures

Rec1. Enhanced MFA resilience C-MFA-01 High Not tested

- State: Recommended
- Description:

Implement strict monitoring of MFA usage patterns to detect anomalies.

Ensure MFA tokens or devices are encrypted and tied to a specific user and device.

Regularly update and patch MFA software to fix vulnerabilities.

Educate users on recognizing phishing attempts and other social engineering attacks.

Rec2. Secure MFA fallbacks C-MFA-03 High Not tested

- State: Recommended
- Description:

Use secure fallback options, such as hardware tokens or biometrics, instead of SMS or email.
Limit the number of MFA bypass attempts and alert administrators on suspicious activity.
Ensure that all fallback mechanisms require re-authentication with a different factor than what was compromised.

Rec3. User training and awareness programs C-MFA-05 Very high Not tested

- State: Recommended
- Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Component: Financial Transaction

1 Recommended countermeasures

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium Not tested

- State: Recommended
- Description:
Implementing an access control system within a financial transaction environment requires careful planning and execution to ensure that sensitive data and operations are securely managed. Follow these actionable steps, oriented towards developers with or without previous security experience, to establish a robust access control system:
Define Security Requirements and Objectives:
 - Identify the assets that require protection, including data, applications, and systems.
 - Determine the security requirements for each asset based on its sensitivity and the potential impact of unauthorized access.**Classify Users and Assets:**
 - Classify users into roles based on their job functions and responsibilities within the organization.
 - Assign sensitivity levels to assets (e.g., public, internal, confidential, highly confidential) to facilitate appropriate access control measures.**Select an Access Control Model:**
 - Choose an access control model that best fits your organizational needs and security objectives. Common models include:
 - Discretionary Access Control (DAC): Access is based on the identity of the users and/or the groups to which they belong.
 - Role-Based Access Control (RBAC): Access permissions are based on roles assigned to users. Users can have multiple roles, and roles can have multiple users.
 - Attribute-Based Access Control (ABAC): Access decisions are based on attributes (user attributes, resource attributes, and environment conditions).**Implement Authentication Mechanisms:**
 - Deploy strong authentication methods to verify the identity of users attempting to access the system. Consider multi-factor authentication (MFA) to add an extra layer of security.**Design and Implement Authorization Mechanisms:**
 - Once authenticated, users should be granted access only to the resources necessary for their roles. Implement permissions and access controls based on the selected access control model.
 - Ensure that the principle of least privilege is applied, providing users with the minimum levels of access or permissions needed to perform their tasks.**Develop Secure Administration Procedures:**
 - Establish secure procedures for administering access controls, including how roles are assigned, changed, and revoked.
 - Ensure that changes to access control policies and configurations are logged and auditable.**Ensure Scalability and Flexibility:**
 - Design the access control system to be scalable and flexible, allowing for the addition of new users, roles, and resources without compromising security.**Monitor and Audit Access:**
 - Implement monitoring and logging to detect and record access attempts, both successful and unsuccessful. This helps in identifying potential security incidents and ensuring compliance with policies.
 - Regularly audit access controls and logs to ensure that only authorized users are accessing sensitive resources and that the access control system is functioning as intended.**Educate Users and Enforce Policies:**
 - Train users on the importance of security policies and the proper handling of sensitive information.
 - Enforce security policies and access controls consistently across the organization.**Review and Update Access Controls:**
 - Regularly review and update access controls in response to changes in the organizational structure, user roles, or security requirements.
 - Adjust policies and permissions as necessary to address new threats, vulnerabilities, and changes in regulatory requirements.Implementing an access control system is a dynamic process that requires ongoing attention and maintenance. By following these steps, you can create a secure and efficient environment that protects sensitive financial transactions and data from unauthorized access while facilitating legitimate business activities.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: Multifactor Authentication (MFA)

Rec1. Enhanced MFA resilience C-MFA-01 High

Rec2. Secure MFA fallbacks C-MFA-03 High

Rec3. User training and awareness programs C-MFA-05 Very high

Component: Financial Transaction

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium

PCI-DSS-v3.2.1: 7.2

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:
Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

- Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
- Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

- Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

- For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
- For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

- Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

- During registration or first login, prompt users to set up MFA.
- Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

- Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
- Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
- Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: Multifactor Authentication (MFA)

3 Recommended countermeasures

Rec1. Enhanced MFA resilience C-MFA-01 High Not tested

- State: Recommended
- Description:
Implement strict monitoring of MFA usage patterns to detect anomalies.
Ensure MFA tokens or devices are encrypted and tied to a specific user and device.
Regularly update and patch MFA software to fix vulnerabilities.
Educate users on recognizing phishing attempts and other social engineering attacks.

Rec2. Secure MFA fallbacks C-MFA-03 High Not tested

- State: Recommended
- Description:
Use secure fallback options, such as hardware tokens or biometrics, instead of SMS or email.
Limit the number of MFA bypass attempts and alert administrators on suspicious activity.
Ensure that all fallback mechanisms require re-authentication with a different factor than what was compromised.

Rec3. User training and awareness programs C-MFA-05 Very high Not tested

- State: Recommended
- Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Component: Financial Transaction

1 Recommended countermeasures

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium Not tested

- State: Recommended
- Description:
Implementing an access control system within a financial transaction environment requires careful planning and execution to ensure that sensitive data and operations are securely managed. Follow these actionable steps, oriented towards developers with or without previous security experience, to establish a robust access control system:
Define Security Requirements and Objectives:
 - Identify the assets that require protection, including data, applications, and systems.
 - Determine the security requirements for each asset based on its sensitivity and the potential impact of unauthorized access.**Classify Users and Assets:**
 - Classify users into roles based on their job functions and responsibilities within the organization.
 - Assign sensitivity levels to assets (e.g., public, internal, confidential, highly confidential) to facilitate appropriate access control measures.**Select an Access Control Model:**
 - Choose an access control model that best fits your organizational needs and security objectives. Common models include:
 - Discretionary Access Control (DAC): Access is based on the identity of the users and/or the groups to which they belong.
 - Role-Based Access Control (RBAC): Access permissions are based on roles assigned to users. Users can have multiple roles, and roles can have multiple users.
 - Attribute-Based Access Control (ABAC): Access decisions are based on attributes (user attributes, resource attributes, and environment conditions).**Implement Authentication Mechanisms:**
 - Deploy strong authentication methods to verify the identity of users attempting to access the system. Consider multi-factor authentication (MFA) to add an extra layer of security.**Design and Implement Authorization Mechanisms:**
 - Once authenticated, users should be granted access only to the resources necessary for their roles. Implement permissions and access controls based on the selected access control model.
 - Ensure that the principle of least privilege is applied, providing users with the minimum levels of access or permissions needed to perform their tasks.**Develop Secure Administration Procedures:**
 - Establish secure procedures for administering access controls, including how roles are assigned, changed, and revoked.
 - Ensure that changes to access control policies and configurations are logged and auditable.**Ensure Scalability and Flexibility:**
 - Design the access control system to be scalable and flexible, allowing for the addition of new users, roles, and resources without compromising security.**Monitor and Audit Access:**
 - Implement monitoring and logging to detect and record access attempts, both successful and unsuccessful. This helps in identifying potential security incidents and ensuring compliance with policies.
 - Regularly audit access controls and logs to ensure that only authorized users are accessing sensitive resources and that the access control system is functioning as intended.**Educate Users and Enforce Policies:**
 - Train users on the importance of security policies and the proper handling of sensitive information.
 - Enforce security policies and access controls consistently across the organization.**Review and Update Access Controls:**
 - Regularly review and update access controls in response to changes in the organizational structure, user roles, or security requirements.

- Adjust policies and permissions as necessary to address new threats, vulnerabilities, and changes in regulatory requirements.
- Implementing an access control system is a dynamic process that requires ongoing attention and maintenance. By following these steps, you can create a secure and efficient environment that protects sensitive financial transactions and data from unauthorized access while facilitating legitimate business activities.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: Multifactor Authentication (MFA)

Rec1. Enhanced MFA resilience C-MFA-01 High

Rec2. Secure MFA fallbacks C-MFA-03 High

Rec3. User training and awareness programs C-MFA-05 Very high

Component: Financial Transaction

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium

PCI-DSS-v3.2.1: 8.7

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: Multifactor Authentication (MFA)

2 Recommended countermeasures

Rec1. Enhanced MFA resilience C-MFA-01 High Not tested

- State: Recommended
- Description:

Implement strict monitoring of MFA usage patterns to detect anomalies.

Ensure MFA tokens or devices are encrypted and tied to a specific user and device.

Regularly update and patch MFA software to fix vulnerabilities.

Educate users on recognizing phishing attempts and other social engineering attacks.

Rec2. Secure MFA fallbacks C-MFA-03 High Not tested

- State: Recommended
- Description:

Use secure fallback options, such as hardware tokens or biometrics, instead of SMS or email.

Limit the number of MFA bypass attempts and alert administrators on suspicious activity.

Ensure that all fallback mechanisms require re-authentication with a different factor than what was compromised.

Component: Financial Transaction

1 Recommended countermeasures

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium ☐ Not tested

- State: ☒ Recommended
- Description:

Implementing an access control system within a financial transaction environment requires careful planning and execution to ensure that sensitive data and operations are securely managed. Follow these actionable steps, oriented towards developers with or without previous security experience, to establish a robust access control system:

Define Security Requirements and Objectives:

 - Identify the assets that require protection, including data, applications, and systems.
 - Determine the security requirements for each asset based on its sensitivity and the potential impact of unauthorized access.

Classify Users and Assets:

 - Classify users into roles based on their job functions and responsibilities within the organization.
 - Assign sensitivity levels to assets (e.g., public, internal, confidential, highly confidential) to facilitate appropriate access control measures.

Select an Access Control Model:

 - Choose an access control model that best fits your organizational needs and security objectives. Common models include:
 - Discretionary Access Control (DAC): Access is based on the identity of the users and/or the groups to which they belong.
 - Role-Based Access Control (RBAC): Access permissions are based on roles assigned to users. Users can have multiple roles, and roles can have multiple users.
 - Attribute-Based Access Control (ABAC): Access decisions are based on attributes (user attributes, resource attributes, and environment conditions).

Implement Authentication Mechanisms:

 - Deploy strong authentication methods to verify the identity of users attempting to access the system. Consider multi-factor authentication (MFA) to add an extra layer of security.

Design and Implement Authorization Mechanisms:

 - Once authenticated, users should be granted access only to the resources necessary for their roles. Implement permissions and access controls based on the selected access control model.
 - Ensure that the principle of least privilege is applied, providing users with the minimum levels of access or permissions needed to perform their tasks.

Develop Secure Administration Procedures:

 - Establish secure procedures for administering access controls, including how roles are assigned, changed, and revoked.
 - Ensure that changes to access control policies and configurations are logged and auditable.

Ensure Scalability and Flexibility:

 - Design the access control system to be scalable and flexible, allowing for the addition of new users, roles, and resources without compromising security.

Monitor and Audit Access:

 - Implement monitoring and logging to detect and record access attempts, both successful and unsuccessful. This helps in identifying potential security incidents and ensuring compliance with policies.
 - Regularly audit access controls and logs to ensure that only authorized users are accessing sensitive resources and that the access control system is functioning as intended.

Educate Users and Enforce Policies:

 - Train users on the importance of security policies and the proper handling of sensitive information.
 - Enforce security policies and access controls consistently across the organization.

Review and Update Access Controls:

 - Regularly review and update access controls in response to changes in the organizational structure, user roles, or security requirements.
 - Adjust policies and permissions as necessary to address new threats, vulnerabilities, and changes in regulatory requirements.

Implementing an access control system is a dynamic process that requires ongoing attention and maintenance. By following these steps, you can create a secure and efficient environment that protects sensitive financial transactions and data from unauthorized access while facilitating legitimate business activities.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: Multifactor Authentication (MFA)

Rec1. Enhanced MFA resilience C-MFA-01 High

Rec2. Secure MFA fallbacks C-MFA-03 High

Component: Financial Transaction

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium

PCI-DSS-v3.2.1: 9.3

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium ☐ Not tested

- State: ☒ Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.

- Monitor and review authentication logs for any unusual activities or failed login attempts.
- Educate Users:**
- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.
- Compliance and Best Practices:**
- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.
- Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: Multifactor Authentication (MFA)

2 Recommended countermeasures

Rec1. Enhanced MFA resilience C-MFA-01 High Not tested

- State: Recommended
- Description:
Implement strict monitoring of MFA usage patterns to detect anomalies.
Ensure MFA tokens or devices are encrypted and tied to a specific user and device.
Regularly update and patch MFA software to fix vulnerabilities.
Educate users on recognizing phishing attempts and other social engineering attacks.

Rec2. Secure MFA fallbacks C-MFA-03 High Not tested

- State: Recommended
- Description:
Use secure fallback options, such as hardware tokens or biometrics, instead of SMS or email.
Limit the number of MFA bypass attempts and alert administrators on suspicious activity.
Ensure that all fallback mechanisms require re-authentication with a different factor than what was compromised.

Component: Financial Transaction

1 Recommended countermeasures

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium Not tested

- State: Recommended
- Description:
Implementing an access control system within a financial transaction environment requires careful planning and execution to ensure that sensitive data and operations are securely managed. Follow these actionable steps, oriented towards developers with or without previous security experience, to establish a robust access control system:
Define Security Requirements and Objectives:
 - Identify the assets that require protection, including data, applications, and systems.
 - Determine the security requirements for each asset based on its sensitivity and the potential impact of unauthorized access.**Classify Users and Assets:**
 - Classify users into roles based on their job functions and responsibilities within the organization.
 - Assign sensitivity levels to assets (e.g., public, internal, confidential, highly confidential) to facilitate appropriate access control measures.**Select an Access Control Model:**
 - Choose an access control model that best fits your organizational needs and security objectives. Common models include:
 - Discretionary Access Control (DAC): Access is based on the identity of the users and/or the groups to which they belong.
 - Role-Based Access Control (RBAC): Access permissions are based on roles assigned to users. Users can have multiple roles, and roles can have multiple users.
 - Attribute-Based Access Control (ABAC): Access decisions are based on attributes (user attributes, resource attributes, and environment conditions).**Implement Authentication Mechanisms:**
 - Deploy strong authentication methods to verify the identity of users attempting to access the system. Consider multi-factor authentication (MFA) to add an extra layer of security.**Design and Implement Authorization Mechanisms:**
 - Once authenticated, users should be granted access only to the resources necessary for their roles. Implement permissions and access controls based on the selected access control model.
 - Ensure that the principle of least privilege is applied, providing users with the minimum levels of access or permissions needed to perform their tasks.**Develop Secure Administration Procedures:**
 - Establish secure procedures for administering access controls, including how roles are assigned, changed, and revoked.
 - Ensure that changes to access control policies and configurations are logged and auditable.**Ensure Scalability and Flexibility:**
 - Design the access control system to be scalable and flexible, allowing for the addition of new users, roles, and resources without compromising security.**Monitor and Audit Access:**
 - Implement monitoring and logging to detect and record access attempts, both successful and unsuccessful. This helps in identifying potential security incidents and ensuring compliance with policies.
 - Regularly audit access controls and logs to ensure that only authorized users are accessing sensitive resources and that the access control system is functioning as intended.**Educate Users and Enforce Policies:**
 - Train users on the importance of security policies and the proper handling of sensitive information.
 - Enforce security policies and access controls consistently across the organization.**Review and Update Access Controls:**
 - Regularly review and update access controls in response to changes in the organizational structure, user roles, or security requirements.
 - Adjust policies and permissions as necessary to address new threats, vulnerabilities, and changes in regulatory requirements.
Implementing an access control system is a dynamic process that requires ongoing attention and maintenance. By following these steps, you can create a secure and efficient environment that protects sensitive financial transactions and data from unauthorized access while facilitating legitimate business activities.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: Multifactor Authentication (MFA)

Rec1. Enhanced MFA resilience C-MFA-01 High

Rec2. Secure MFA fallbacks C-MFA-03 High

Component: Financial Transaction

Rec1. Implement an access control system C-ACCESS-CONTROL-SYSTEM Medium

PCI-DSS-v3.2.1: 10.6.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: Financial Transaction

Imp1. Use of a DDoS protection service for Payment System C-DDOS-PROTECTION Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.

- **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- **State:** Recommended
- **Description:**

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.6.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: Financial Transaction

Imp1. Use of a DDoS protection service for Payment System C-DDOS-PROTECTION Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- **State:** Recommended
- **Description:**

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
```

```
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

- Use Indicators for System Status**
- Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`
- Provide Contextual Help and Tooltips**
- Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`
- Test for Clarity and Effectiveness**
- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.
- Ensure Accessibility**
- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.
- Additional Considerations:**
- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- **State:** Recommended
- **Description:**
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- **State:** Recommended
- **Description:**

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 11.4

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: Financial Transaction

Imp1. Use of a DDoS protection service for Payment System C-DDOS-PROTECTION Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description: Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.

- **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
- **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
- **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
- **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.6.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
```

```
margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

- Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
- Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
- Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

- Integrate tooltips or help icons next to complex features or inputs that require explanation.
- Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
- Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
- Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
- Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
- Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High ☐ Not tested

- State: Recommended
- Description:
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium ☐ Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 11.5.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 - Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 - Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 - Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 - Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 - Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 - Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 12.10.5

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;

  margin: 4px 2px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
.button:hover {
  background-color: #45a049;
```

```
}

Use Indicators for System Status
  • Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
  • Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
  • Example in HTML: <div id="status" class="status-offline">Offline</div>

Provide Contextual Help and Tooltips
  • Integrate tooltips or help icons next to complex features or inputs that require explanation.
  • Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
  • Example HTML for a tooltip: <a href="#" data-tooltip="Enter your email here. We do not share your email with anyone.">Email:</a>

Test for Clarity and Effectiveness
  • Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
  • Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility
  • Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
  • Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:
  • Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
  • Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.
```

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

State: Recommended

Description:

- Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
- Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
- Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
- Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
- Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
- Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
- User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
- Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
- Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
- Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
- Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
- Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
- Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

State: Recommended

Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

- Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
- Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
- Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
- Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
- Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
- Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 12.5.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 - Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 - Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 - Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 - Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 - Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 - Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 9.1.1

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.
- Additional Considerations:**
- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.
- By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.1

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.2

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:
Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.
Design Clear and Intuitive Cues
 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.**Implement Feedback for Interactive Elements**
 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```**Use Indicators for System Status**
 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`**Provide Contextual Help and Tooltips**
 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`**Test for Clarity and Effectiveness**
 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.**Ensure Accessibility**
 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.**Additional Considerations:**
 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 - Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 - Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 - Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 - Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 - Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 - Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.3

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;

  margin: 4px 2px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
.button:hover {
  background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
- Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
- Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
- Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

• State: Recommended

• Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.4

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

• State: Recommended

• Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

- Develop a consistent visual language that aligns with your application's design guidelines.
- Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
- Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

- Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
- Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
- Example CSS for a button:

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
```

```
margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
```

```
.button:hover {
  background-color: #45a049;
```

```
}

Use Indicators for System Status
  • Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
  • Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
  • Example in HTML: <div id="status" class="status-offline">Offline</div>

Provide Contextual Help and Tooltips
  • Integrate tooltips or help icons next to complex features or inputs that require explanation.
  • Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
  • Example HTML for a tooltip: <a href="#" data-tooltip="Enter your email here. We do not share your email with anyone.">Email:</a>

Test for Clarity and Effectiveness
  • Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
  • Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility
  • Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
  • Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:
  • Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
  • Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.
```

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

 - Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 - Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 - Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 - Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 - Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 - Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.5

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
```



```
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

- Use Indicators for System Status**
- Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`
- Provide Contextual Help and Tooltips**
- Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`
- Test for Clarity and Effectiveness**
- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.
- Ensure Accessibility**
- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.
- Additional Considerations:**
- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.6

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high ☐ Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;

  margin: 4px 2px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.button:hover {
  background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High ☐ Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium ☐ Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

- Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).
- User Registration and Recovery:**
- During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.
- Test and Deploy:**
- Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.
- Educate Users:**
- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.
- Compliance and Best Practices:**
- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.
- Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Component: Multifactor Authentication (MFA)

2 Recommended countermeasures

Rec1. Enhanced MFA resilience C-MFA-01 High Not tested

- State: Recommended
- Description:
Implement strict monitoring of MFA usage patterns to detect anomalies.
Ensure MFA tokens or devices are encrypted and tied to a specific user and device.
Regularly update and patch MFA software to fix vulnerabilities.
Educate users on recognizing phishing attempts and other social engineering attacks.

Rec2. Secure MFA fallbacks C-MFA-03 High Not tested

- State: Recommended
- Description:
Use secure fallback options, such as hardware tokens or biometrics, instead of SMS or email.
Limit the number of MFA bypass attempts and alert administrators on suspicious activity.
Ensure that all fallback mechanisms require re-authentication with a different factor than what was compromised.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:
Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.
Follow these steps to implement this countermeasure:
 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.
By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

Component: Multifactor Authentication (MFA)

Rec1. Enhanced MFA resilience C-MFA-01 High

Rec2. Secure MFA fallbacks C-MFA-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.7

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

| | | | |
|---|---|-----------|---|
| Component: User Interface | | | |
| 1 Recommended countermeasures | | | |
| Rec1. Implement visual cues and indicators | C-USER-INTERFACE-VISUAL-CUES | Very high | <input checked="" type="radio"/> Not tested |
| <div><div>• State: Recommended</div><div>• Description:
Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.
Design Clear and Intuitive Cues<ul style="list-style-type: none">Develop a consistent visual language that aligns with your application's design guidelines.Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.Example: Use a red color for warnings, green for successful actions, and yellow for caution.Implement Feedback for Interactive Elements<ul style="list-style-type: none">Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.Example CSS for a button:<pre>.button { background-color: #4CAF50; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; margin: 4px 2px; cursor: pointer; transition: background-color 0.3s ease; } .button:hover { background-color: #45a049; }</pre>Use Indicators for System Status<ul style="list-style-type: none">Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).Example in HTML: <code><div id="status" class="status-offline">Offline</div></code>Provide Contextual Help and Tooltips<ul style="list-style-type: none">Integrate tooltips or help icons next to complex features or inputs that require explanation.Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.Example HTML for a tooltip: <code>Email:</code>Test for Clarity and Effectiveness<ul style="list-style-type: none">Conduct usability testing to gauge the effectiveness of your visual cues and indicators.Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.Ensure Accessibility<ul style="list-style-type: none">Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.Use text labels or ARIA labels in addition to color-based cues.Additional Considerations:<ul style="list-style-type: none">Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.Regularly update visual elements to stay in line with modern UI/UX trends and standards.By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.</div></div> | | | |
| Component: IDS (Intrusion Detection System) | | | |
| 1 Recommended countermeasures | | | |
| Rec1. Configure the IDS to send alerts to a central location | C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 | Medium | <input type="radio"/> Not tested |
| <div><div>• State: Recommended</div><div>• Description:
Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.
Follow these steps to implement this countermeasure:<ol style="list-style-type: none">Select a Central Monitoring Solution: Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.Network Configuration: Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.Configure IDS Alert Transmission: Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).Test Alert Transmission: Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.Define Alert Prioritization Criteria: Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.Monitor and Adjust: Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.</div></div> | | | |

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

| | | | |
|--|------------------------------|-----------|--|
| Component: User Interface | | | |
| Rec1. Implement visual cues and indicators | C-USER-INTERFACE-VISUAL-CUES | Very high | |

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 10.8

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
 - Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

 - Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

 - Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

 - Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

- State: Recommended
- Description:

Implement short session expiration times and require re-authentication for critical actions.

Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.

Monitor session activity and automatically log out users when unusual behavior is detected.

Educate users on avoiding insecure networks and using VPNs for secure connections.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 - Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 - Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 - Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 - Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 - Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 - Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 11.1

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

 - Develop a consistent visual language that aligns with your application's design guidelines.
 - Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
 - Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

 - Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
 - Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
 - Example CSS for a button:

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;

  margin: 4px 2px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
.button:hover {
  background-color: #45a049;
}
```

Use Indicators for System Status

 - Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
 - Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
 - Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

 - Integrate tooltips or help icons next to complex features or inputs that require explanation.
 - Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.

- Example HTML for a tooltip: `Email:`
- Test for Clarity and Effectiveness**
- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
 - Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.
- Ensure Accessibility**
- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
 - Use text labels or ARIA labels in addition to color-based cues.
- Additional Considerations:**
- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
 - Regularly update visual elements to stay in line with modern UI/UX trends and standards.
- By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- **State:** Recommended
- **Description:**
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- **State:** Recommended
- **Description:**

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

 1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 11.5

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

State: Recommended

Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

- Develop a consistent visual language that aligns with your application's design guidelines.
- Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
- Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

- Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
- Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
- Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

- Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
- Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
- Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

- Integrate tooltips or help icons next to complex features or inputs that require explanation.
- Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
- Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
- Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
- Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
- Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

State: Recommended

Description:

- Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
- Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
- Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
- Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
- Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
- Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
- User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
- Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
- Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
- Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
- Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
- Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
- Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

State: Recommended

Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner.

Follow these steps to implement this countermeasure:

- Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
- Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
- Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
- Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
- Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.

6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 12.10

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high ☐ Not tested

• State: ☒ Recommended

• Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

- Develop a consistent visual language that aligns with your application's design guidelines.
- Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
- Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

- Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
- Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
- Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

- Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
- Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
- Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

- Integrate tooltips or help icons next to complex features or inputs that require explanation.
- Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
- Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
- Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
- Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
- Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

- State: Recommended
- Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

 - Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
 - Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
 - Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
 - Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
 - Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
 - Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 12.8

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high Not tested

- State: Recommended
- Description:

Visual cues and indicators enhance the usability and security of your application by providing feedback or warnings to users about their interactions or system states. These can include changes in button colors when active, spinner icons during loading phases, or warning messages for potentially insecure actions.

Design Clear and Intuitive Cues

- Develop a consistent visual language that aligns with your application's design guidelines.
- Use color, shape, animation, and positioning strategically to draw attention without overwhelming the user.
- Example: Use a red color for warnings, green for successful actions, and yellow for caution.

Implement Feedback for Interactive Elements

- Provide immediate feedback for user actions. For example, change the appearance of a button when hovered or clicked.
- Use animations to indicate loading or processing states, which helps users understand that the application is active and responsive.
- Example CSS for a button:

```
.button {
background-color: #4CAF50; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;

margin: 4px 2px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.button:hover {
background-color: #45a049;
}
```

Use Indicators for System Status

- Indicate system or process statuses clearly, such as online/offline modes, connection issues, or security states.
- Implement a visual indicator for secure sessions (e.g., a padlock icon when a connection is secured with SSL/TLS).
- Example in HTML: `<div id="status" class="status-offline">Offline</div>`

Provide Contextual Help and Tooltips

- Integrate tooltips or help icons next to complex features or inputs that require explanation.
- Ensure tooltips are accessible on all devices, including mobile and those used by individuals with disabilities.
- Example HTML for a tooltip: `Email:`

Test for Clarity and Effectiveness

- Conduct usability testing to gauge the effectiveness of your visual cues and indicators.
- Gather user feedback to refine and adjust the visual elements to ensure they are understood and appreciated by users.

Ensure Accessibility

- Make sure that all visual cues are accessible, including to those with visual impairments or color blindness.
- Use text labels or ARIA labels in addition to color-based cues.

Additional Considerations:

- Keep in mind that not all users interpret colors and icons the same way; cultural differences can influence user perception.
- Regularly update visual elements to stay in line with modern UI/UX trends and standards.

By implementing visual cues and indicators as described, you can significantly enhance user interaction and security awareness within your application, providing a more engaging and safe experience.

Component: IDS (Intrusion Detection System)

1 Recommended countermeasures

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium Not tested

• State: Recommended

• Description:

Ensuring prompt detection and response to security threats is crucial for maintaining the integrity and availability of your network. To achieve this, configure your Intrusion Detection System (IDS) to transmit alerts to a centralized monitoring location. This setup will enable security analysts to view, prioritize, and respond to potential threats in a timely manner. Follow these steps to implement this countermeasure:

1. **Select a Central Monitoring Solution:** Identify a centralized monitoring platform (e.g., a Security Information and Event Management system) that will serve as the repository for all IDS alerts. Ensure the solution is scalable, secure, and allows for detailed analysis and reporting.
2. **Network Configuration:** Establish a secure communication channel between the IDS and the central monitoring location. Depending on your network architecture, this could involve configuring VLANs, setting up secure tunnels (e.g., VPNs), or using encrypted protocols for data transmission.
3. **Configure IDS Alert Transmission:** Access the IDS's configuration settings and specify the IP address or hostname of the central monitoring location. Set the appropriate network ports if required, and choose a reliable transmission protocol (such as Syslog, SNMP, or an API-based approach).
4. **Test Alert Transmission:** Initiate a series of test alerts to verify that the IDS successfully sends notifications to the central monitoring location. Ensure that the communication process is reliable and that the alerts are received, processed, and logged accurately.
5. **Define Alert Prioritization Criteria:** Within the central monitoring solution, establish criteria to prioritize alerts based on severity, source, and type. Setting thresholds and automated responses for critical alerts can streamline the incident response process.
6. **Monitor and Adjust:** Regularly review the performance of the alert transmission configuration. Adjust settings based on the evolving threat landscape and organizational needs to maintain an effective security posture.

By properly setting up your IDS to send alerts to a central monitoring location, your organization can enhance its ability to promptly detect and respond to security incidents, thereby mitigating potential risks and safeguarding critical assets.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Implement visual cues and indicators C-USER-INTERFACE-VISUAL-CUES Very high

Component: IDS (Intrusion Detection System)

Rec1. Configure the IDS to send alerts to a central location C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-02 Medium

PCI-DSS-v3.2.1: 12.10.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

PCI-DSS-v3.2.1: 12.10.6

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

PCI-DSS-v3.2.1: 9.6.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 1.5

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 2.5

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 3.7

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 4.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 5.4

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 6.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: Login

1 Recommended countermeasures

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High Not tested

- State: Recommended
- Description:

Conducting regular security audits and reviews of your login system is a critical step in ensuring its ongoing security and integrity. These audits help identify vulnerabilities, assess the effectiveness of current security measures, and ensure compliance with best practices and regulations. Here's a step-by-step guide on how to implement this countermeasure effectively:

 - Establish an Audit Schedule**
 - Define Frequency:** Determine how often security audits and reviews should be conducted. The frequency can depend on various factors, including the sensitivity of the data handled by the login system, regulatory requirements, and the system's complexity.
 - Plan for Regular Reviews:** In addition to full audits, plan for more frequent, less formal security reviews to quickly catch and address potential issues.
 - Outline Audit Scope**
 - Identify Components:** List all components of the login system to be audited. This includes the authentication mechanism, database storage of credentials, session management, and any multi-factor authentication (MFA) integrations.
 - Determine Audit Criteria:** Define what standards, regulations, and best practices the audit will use as benchmarks for evaluation. Common references include OWASP Top 10, ISO/IEC 27001, and specific compliance mandates like GDPR or HIPAA.
 - Conduct the Security Audit**
 - Review Code:** Perform a thorough code review focusing on authentication flows, data validation, and session management. Look for common vulnerabilities like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).
 - Test Authentication Mechanisms:** Assess the strength and implementation of password policies, MFA, and session management practices. Use both automated tools and manual testing techniques.
 - Evaluate Configuration and Deployment:** Check the configuration of servers, databases, and any third-party services used in the login process. Ensure that only necessary services are exposed and securely configured.
 - Assess Incident Response Mechanisms:** Review how the system detects, logs, and responds to security incidents. Ensure that adequate logging is in place and that alerts are configured for suspicious activities.
 - Document Findings and Recommendations**
 - Compile a Report:** Document all findings from the audit, including vulnerabilities discovered, areas for improvement, and adherence to best practices and compliance requirements.
 - Prioritize Issues:** Rank the identified issues based on their potential impact and the effort required to address them. High-risk vulnerabilities should be prioritized for immediate remediation.
 - Implement Recommendations**
 - Develop a Remediation Plan:** For each identified issue, outline a plan for remediation. Assign responsibilities and set deadlines for addressing the vulnerabilities.
 - Monitor Progress:** Track the implementation of the remediation plan, ensuring that all issues are addressed in a timely manner.
 - Review and Iterate**
 - Post-Implementation Review:** After implementing the recommendations, conduct a follow-up review to ensure that the changes have effectively addressed the vulnerabilities.
 - Continuous Improvement:** Use the insights gained from each audit to refine the audit process and improve the security of the login system continuously.
 - Train and Educate**
 - Educate Developers:** Share the findings and lessons learned from the audit with the development team. Use this as an opportunity to improve secure coding practices.
 - Awareness for All Stakeholders:** Ensure that all stakeholders understand the importance of the security audit process and their role in maintaining the security of the login system.

Conducting regular security audits and reviews is an essential practice for maintaining the security of your login system. It helps identify vulnerabilities, ensures compliance with security standards, and fosters a culture of continuous improvement in security practices.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: Login

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High

PCI-DSS-v3.2.1: 6.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Conduct regular security audits and reviewsC-LOGIN-CM5

High

Not tested

State: Recommended

Description:

Conducting regular security audits and reviews of your login system is a critical step in ensuring its ongoing security and integrity. These audits help identify vulnerabilities, assess the effectiveness of current security measures, and ensure compliance with best practices and regulations. Here's a step-by-step guide on how to implement this countermeasure effectively:

1. Establish an Audit Schedule

Define Frequency: Determine how often security audits and reviews should be conducted. The frequency can depend on various factors, including the sensitivity of the data handled by the login system, regulatory requirements, and the system's complexity.

Plan for Regular Reviews: In addition to full audits, plan for more frequent, less formal security reviews to quickly catch and address potential issues.

2. Outline Audit Scope

Identify Components: List all components of the login system to be audited. This includes the authentication mechanism, database storage of credentials, session management, and any multi-factor authentication (MFA) integrations.

Determine Audit Criteria: Define what standards, regulations, and best practices the audit will use as benchmarks for evaluation. Common references include OWASP Top 10, ISO/IEC 27001, and specific compliance mandates like GDPR or HIPAA.

3. Conduct the Security Audit

Review Code: Perform a thorough code review focusing on authentication flows, data validation, and session management. Look for common vulnerabilities like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).

Test Authentication Mechanisms: Assess the strength and implementation of password policies, MFA, and session management practices. Use both automated tools and manual testing techniques.

Evaluate Configuration and Deployment: Check the configuration of servers, databases, and any third-party services used in the login process. Ensure that only necessary services are exposed and securely configured.

Assess Incident Response Mechanisms: Review how the system detects, logs, and responds to security incidents. Ensure that adequate logging is in place and that alerts are configured for suspicious activities.

4. Document Findings and Recommendations

Compile a Report: Document all findings from the audit, including vulnerabilities discovered, areas for improvement, and adherence to best practices and compliance requirements.

Prioritize Issues: Rank the identified issues based on their potential impact and the effort required to address them. High-risk vulnerabilities should be prioritized for immediate remediation.

5. Implement Recommendations

Develop a Remediation Plan: For each identified issue, outline a plan for remediation. Assign responsibilities and set deadlines for addressing the vulnerabilities.

Monitor Progress: Track the implementation of the remediation plan, ensuring that all issues are addressed in a timely manner.

6. Review and Iterate

Post-Implementation Review: After implementing the recommendations, conduct a follow-up review to ensure that the changes have effectively addressed the vulnerabilities.

Continuous Improvement: Use the insights gained from each audit to refine the audit process and improve the security of the login system continuously.

7. Train and Educate

Educate Developers: Share the findings and lessons learned from the audit with the development team. Use this as an opportunity to improve secure coding practices.

Awareness for All Stakeholders: Ensure that all stakeholders understand the importance of the security audit process and their role in maintaining the security of the login system.

Conducting regular security audits and reviews is an essential practice for maintaining the security of your login system. It helps identify vulnerabilities, ensures compliance with security standards, and fosters a culture of continuous improvement in security practices.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancementC-MFA-02

High

Not tested

State: Recommended

Description:

Implement short session expiration times and require re-authentication for critical actions. Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens. Monitor session activity and automatically log out users when unusual behavior is detected. Educate users on avoiding insecure networks and using VPNs for secure connections.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Conduct regular security audits and reviewsC-LOGIN-CM5

High

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancementC-MFA-02

High

PCI-DSS-v3.2.1: 6.5

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularlyC-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01

Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Report: Threat-Modeling_LAB_03

Compliance report - 2025-02-11T11:46:13.61536482Z

55 of 80

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High Not tested

- State: Recommended
- Description:

Conducting regular security audits and reviews of your login system is a critical step in ensuring its ongoing security and integrity. These audits help identify vulnerabilities, assess the effectiveness of current security measures, and ensure compliance with best practices and regulations. Here's a step-by-step guide on how to implement this countermeasure effectively:

 - Establish an Audit Schedule**
 - Define Frequency:** Determine how often security audits and reviews should be conducted. The frequency can depend on various factors, including the sensitivity of the data handled by the login system, regulatory requirements, and the system's complexity.
 - Plan for Regular Reviews:** In addition to full audits, plan for more frequent, less formal security reviews to quickly catch and address potential issues.
 - Outline Audit Scope**
 - Identify Components:** List all components of the login system to be audited. This includes the authentication mechanism, database storage of credentials, session management, and any multi-factor authentication (MFA) integrations.
 - Determine Audit Criteria:** Define what standards, regulations, and best practices the audit will use as benchmarks for evaluation. Common references include OWASP Top 10, ISO/IEC 27001, and specific compliance mandates like GDPR or HIPAA.
 - Conduct the Security Audit**
 - Review Code:** Perform a thorough code review focusing on authentication flows, data validation, and session management. Look for common vulnerabilities like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).
 - Test Authentication Mechanisms:** Assess the strength and implementation of password policies, MFA, and session management practices. Use both automated tools and manual testing techniques.
 - Evaluate Configuration and Deployment:** Check the configuration of servers, databases, and any third-party services used in the login process. Ensure that only necessary services are exposed and securely configured.
 - Assess Incident Response Mechanisms:** Review how the system detects, logs, and responds to security incidents. Ensure that adequate logging is in place and that alerts are configured for suspicious activities.
 - Document Findings and Recommendations**
 - Compile a Report:** Document all findings from the audit, including vulnerabilities discovered, areas for improvement, and adherence to best practices and compliance requirements.
 - Prioritize Issues:** Rank the identified issues based on their potential impact and the effort required to address them. High-risk vulnerabilities should be prioritized for immediate remediation.
 - Implement Recommendations**
 - Develop a Remediation Plan:** For each identified issue, outline a plan for remediation. Assign responsibilities and set deadlines for addressing the vulnerabilities.
 - Monitor Progress:** Track the implementation of the remediation plan, ensuring that all issues are addressed in a timely manner.
 - Review and Iterate**
 - Post-Implementation Review:** After implementing the recommendations, conduct a follow-up review to ensure that the changes have effectively addressed the vulnerabilities.
 - Continuous Improvement:** Use the insights gained from each audit to refine the audit process and improve the security of the login system continuously.
 - Train and Educate**
 - Educate Developers:** Share the findings and lessons learned from the audit with the development team. Use this as an opportunity to improve secure coding practices.
 - Awareness for All Stakeholders:** Ensure that all stakeholders understand the importance of the security audit process and their role in maintaining the security of the login system.

Conducting regular security audits and reviews is an essential practice for maintaining the security of your login system. It helps identify vulnerabilities, ensures compliance with security standards, and fosters a culture of continuous improvement in security practices.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High

PCI-DSS-v3.2.1: 6.7

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

- State: Recommended
- Description:

Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks. Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication. Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 7.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

- State: Recommended
- Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 8.8

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 9.10

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 11.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.

- **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
- **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: Login

1 Recommended countermeasures

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High Not tested

- State: Recommended
- Description:

Conducting regular security audits and reviews of your login system is a critical step in ensuring its ongoing security and integrity. These audits help identify vulnerabilities, assess the effectiveness of current security measures, and ensure compliance with best practices and regulations. Here's a step-by-step guide on how to implement this countermeasure effectively.

 1. **Establish an Audit Schedule**
 - **Define Frequency:** Determine how often security audits and reviews should be conducted. The frequency can depend on various factors, including the sensitivity of the data handled by the login system, regulatory requirements, and the system's complexity.
 - **Plan for Regular Reviews:** In addition to full audits, plan for more frequent, less formal security reviews to quickly catch and address potential issues.
 2. **Outline Audit Scope**
 - **Identify Components:** List all components of the login system to be audited. This includes the authentication mechanism, database storage of credentials, session management, and any multi-factor authentication (MFA) integrations.
 - **Determine Audit Criteria:** Define what standards, regulations, and best practices the audit will use as benchmarks for evaluation. Common references include OWASP Top 10, ISO/IEC 27001, and specific compliance mandates like GDPR or HIPAA.
 3. **Conduct the Security Audit**
 - **Review Code:** Perform a thorough code review focusing on authentication flows, data validation, and session management. Look for common vulnerabilities like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).
 - **Test Authentication Mechanisms:** Assess the strength and implementation of password policies, MFA, and session management practices. Use both automated tools and manual testing techniques.
 - **Evaluate Configuration and Deployment:** Check the configuration of servers, databases, and any third-party services used in the login process. Ensure that only necessary services are exposed and securely configured.
 - **Assess Incident Response Mechanisms:** Review how the system detects, logs, and responds to security incidents. Ensure that adequate logging is in place and that alerts are configured for suspicious activities.
 4. **Document Findings and Recommendations**
 - **Compile a Report:** Document all findings from the audit, including vulnerabilities discovered, areas for improvement, and adherence to best practices and compliance requirements.
 - **Prioritize Issues:** Rank the identified issues based on their potential impact and the effort required to address them. High-risk vulnerabilities should be prioritized for immediate remediation.
 5. **Implement Recommendations**
 - **Develop a Remediation Plan:** For each identified issue, outline a plan for remediation. Assign responsibilities and set deadlines for addressing the vulnerabilities.
 - **Monitor Progress:** Track the implementation of the remediation plan, ensuring that all issues are addressed in a timely manner.
 6. **Review and Iterate**
 - **Post-Implementation Review:** After implementing the recommendations, conduct a follow-up review to ensure that the changes have effectively addressed the vulnerabilities.
 - **Continuous Improvement:** Use the insights gained from each audit to refine the audit process and improve the security of the login system continuously.
 7. **Train and Educate**
 - **Educate Developers:** Share the findings and lessons learned from the audit with the development team. Use this as an opportunity to improve secure coding practices.
 - **Awareness for All Stakeholders:** Ensure that all stakeholders understand the importance of the security audit process and their role in maintaining the security of the login system.

Conducting regular security audits and reviews is an essential practice for maintaining the security of your login system. It helps identify vulnerabilities, ensures compliance with security standards, and fosters a culture of continuous improvement in security practices.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: Login

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High

PCI-DSS-v3.2.1: 11.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.

- **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
- **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
- **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
- **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
- **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
- **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
- **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
- **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: Login

1 Recommended countermeasures

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High Not tested

- State: Recommended
- Description:

Conducting regular security audits and reviews of your login system is a critical step in ensuring its ongoing security and integrity. These audits help identify vulnerabilities, assess the effectiveness of current security measures, and ensure compliance with best practices and regulations. Here's a step-by-step guide on how to implement this countermeasure effectively:

 - 1. Establish an Audit Schedule**
 - **Define Frequency:** Determine how often security audits and reviews should be conducted. The frequency can depend on various factors, including the sensitivity of the data handled by the login system, regulatory requirements, and the system's complexity.
 - **Plan for Regular Reviews:** In addition to full audits, plan for more frequent, less formal security reviews to quickly catch and address potential issues.
 - 2. Outline Audit Scope**
 - **Identify Components:** List all components of the login system to be audited. This includes the authentication mechanism, database storage of credentials, session management, and any multi-factor authentication (MFA) integrations.
 - **Determine Audit Criteria:** Define what standards, regulations, and best practices the audit will use as benchmarks for evaluation. Common references include OWASP Top 10, ISO/IEC 27001, and specific compliance mandates like GDPR or HIPAA.
 - 3. Conduct the Security Audit**
 - **Review Code:** Perform a thorough code review focusing on authentication flows, data validation, and session management. Look for common vulnerabilities like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).
 - **Test Authentication Mechanisms:** Assess the strength and implementation of password policies, MFA, and session management practices. Use both automated tools and manual testing techniques.
 - **Evaluate Configuration and Deployment:** Check the configuration of servers, databases, and any third-party services used in the login process. Ensure that only necessary services are exposed and securely configured.
 - **Assess Incident Response Mechanisms:** Review how the system detects, logs, and responds to security incidents. Ensure that adequate logging is in place and that alerts are configured for suspicious activities.
 - 4. Document Findings and Recommendations**
 - **Compile a Report:** Document all findings from the audit, including vulnerabilities discovered, areas for improvement, and adherence to best practices and compliance requirements.
 - **Prioritize Issues:** Rank the identified issues based on their potential impact and the effort required to address them. High-risk vulnerabilities should be prioritized for immediate remediation.
 - 5. Implement Recommendations**
 - **Develop a Remediation Plan:** For each identified issue, outline a plan for remediation. Assign responsibilities and set deadlines for addressing the vulnerabilities.
 - **Monitor Progress:** Track the implementation of the remediation plan, ensuring that all issues are addressed in a timely manner.
 - 6. Review and Iterate**
 - **Post-Implementation Review:** After implementing the recommendations, conduct a follow-up review to ensure that the changes have effectively addressed the vulnerabilities.
 - **Continuous Improvement:** Use the insights gained from each audit to refine the audit process and improve the security of the login system continuously.
 - 7. Train and Educate**
 - **Educate Developers:** Share the findings and lessons learned from the audit with the development team. Use this as an opportunity to improve secure coding practices.
 - **Awareness for All Stakeholders:** Ensure that all stakeholders understand the importance of the security audit process and their role in maintaining the security of the login system.

Conducting regular security audits and reviews is an essential practice for maintaining the security of your login system. It helps identify vulnerabilities, ensures compliance with security standards, and fosters a culture of continuous improvement in security practices.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

- State: Recommended
- Description:

Implement short session expiration times and require re-authentication for critical actions. Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens. Monitor session activity and automatically log out users when unusual behavior is detected. Educate users on avoiding insecure networks and using VPNs for secure connections.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: Login

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

PCI-DSS-v3.2.1: 11.6

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 12.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

PCI-DSS-v3.2.1: 12.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Update IDS regularly C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-01 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: Login

1 Recommended countermeasures

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High Not tested

- State: Recommended
- Description:

Conducting regular security audits and reviews of your login system is a critical step in ensuring its ongoing security and integrity. These audits help identify vulnerabilities, assess the effectiveness of current security measures, and ensure compliance with best practices and regulations. Here's a step-by-step guide on how to implement this countermeasure effectively:

 - Establish an Audit Schedule**
 - Define Frequency:** Determine how often security audits and reviews should be conducted. The frequency can depend on various factors, including the sensitivity of the data handled by the login system, regulatory requirements, and the system's complexity.
 - Plan for Regular Reviews:** In addition to full audits, plan for more frequent, less formal security reviews to quickly catch and address potential issues.
 - Outline Audit Scope**
 - Identify Components:** List all components of the login system to be audited. This includes the authentication mechanism, database storage of credentials, session management, and any multi-factor authentication (MFA) integrations.
 - Determine Audit Criteria:** Define what standards, regulations, and best practices the audit will use as benchmarks for evaluation. Common references include OWASP Top 10, ISO/IEC 27001, and specific compliance mandates like GDPR or HIPAA.
 - Conduct the Security Audit**
 - Review Code:** Perform a thorough code review focusing on authentication flows, data validation, and session management. Look for common vulnerabilities like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).
 - Test Authentication Mechanisms:** Assess the strength and implementation of password policies, MFA, and session management practices. Use both automated tools and manual testing techniques.
 - Evaluate Configuration and Deployment:** Check the configuration of servers, databases, and any third-party services used in the login process. Ensure that only necessary services are exposed and securely configured.
 - Assess Incident Response Mechanisms:** Review how the system detects, logs, and responds to security incidents. Ensure that adequate logging is in place and that alerts are configured for suspicious activities.
 - Document Findings and Recommendations**
 - Compile a Report:** Document all findings from the audit, including vulnerabilities discovered, areas for improvement, and adherence to best practices and compliance requirements.
 - Prioritize Issues:** Rank the identified issues based on their potential impact and the effort required to address them. High-risk vulnerabilities should be prioritized for immediate remediation.
 - Implement Recommendations**
 - Develop a Remediation Plan:** For each identified issue, outline a plan for remediation. Assign responsibilities and set deadlines for addressing the vulnerabilities.
 - Monitor Progress:** Track the implementation of the remediation plan, ensuring that all issues are addressed in a timely manner.
 - Review and Iterate**
 - Post-Implementation Review:** After implementing the recommendations, conduct a follow-up review to ensure that the changes have effectively addressed the vulnerabilities.
 - Continuous Improvement:** Use the insights gained from each audit to refine the audit process and improve the security of the login system continuously.
 - Train and Educate**
 - Educate Developers:** Share the findings and lessons learned from the audit with the development team. Use this as an opportunity to improve secure coding practices.
 - Awareness for All Stakeholders:** Ensure that all stakeholders understand the importance of the security audit process and their role in maintaining the security of the login system.

Conducting regular security audits and reviews is an essential practice for maintaining the security of your login system. It helps identify vulnerabilities, ensures compliance with security standards, and fosters a culture of continuous improvement in security practices.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: Login

Rec1. Conduct regular security audits and reviews C-LOGIN-CM5 High

PCI-DSS-v3.2.1: 1.1.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

PCI-DSS-v3.2.1: 1.1.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.

- **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
- **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
- **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
- **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
- **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
- **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
- **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
- **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
- **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
- **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
- **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
- **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

PCI-DSS-v3.2.1: 1.1.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

PCI-DSS-v3.2.1: 12.3.10

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 12.3.8

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your

implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 12.3.9

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 8.1.5

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High Not tested

- State: Recommended
- Description:
 - Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 8.2.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

PCI-DSS-v3.2.1: 8.5.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1

Medium

Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1

Medium

PCI-DSS-v3.2.1: 2.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03

Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1

Medium

Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

- Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.
- Educate Users:**
- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.
- Compliance and Best Practices:**
- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.
- Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 8.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: IDS (Intrusion Detection System)

Imp1. Use an out-of-band management connection for IDS C-IDS-INTRUSION-DETECTION-SYSTEM-CNT-03 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high ☐ Not tested

- State: Recommended
- Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 1. **Something you know** (e.g., password or PIN)
 2. **Something you have** (e.g., security token, smartphone app, or smart card)
 3. **Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

 - Keep the MFA solution updated to defend against new vulnerabilities.
 - Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

 - Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
 - Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium ☐ Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

- Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

- For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
- For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

- Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

- During registration or first login, prompt users to set up MFA.
- Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

- Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
- Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
- Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 7.1.4

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high Not tested

- State: Recommended
- Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 - Something you know** (e.g., password or PIN)
 - Something you have** (e.g., security token, smartphone app, or smart card)
 - Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

 - Keep the MFA solution updated to defend against new vulnerabilities.
 - Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

 - Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
 - Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

- Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
- Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

- Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

- For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
- For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

- Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

- During registration or first login, prompt users to set up MFA.
- Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

- Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
- Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

- Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

- Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
- Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 8.2.2


Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high ☐ Not tested

- **State:**  Recommended
- **Description:**

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 1. **Something you know** (e.g., password or PIN)
 2. **Something you have** (e.g., security token, smartphone app, or smart card)
 3. **Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

 - Keep the MFA solution updated to defend against new vulnerabilities.
 - Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

 - Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
 - Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 8.1

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high Not tested

- State: Recommended
- Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 - Something you know** (e.g., password or PIN)
 - Something you have** (e.g., security token, smartphone app, or smart card)
 - Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

 - Keep the MFA solution updated to defend against new vulnerabilities.
 - Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

 - Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
 - Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium ☐ Not tested

- State: ☒ Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 8.2

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high ☐ Not tested

- State: ☒ Recommended
- Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 - Something you know** (e.g., password or PIN)
 - Something you have** (e.g., security token, smartphone app, or smart card)
 - Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

- Keep the MFA solution updated to defend against new vulnerabilities.
- Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

- Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
- Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Component: Login

1 Recommended countermeasures

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium ☐ Not tested

- State: Recommended
- Description:

Implementing Multi-Factor Authentication (MFA) significantly enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as a login system. This countermeasure makes it harder for attackers to gain unauthorized access, as they would need to compromise multiple authentication mechanisms. Here's how to implement it:

Choose an MFA Method: Decide on the types of factors you will use. Common types include something you know (password or PIN), something you have (a mobile device or security token), and something you are (biometrics such as fingerprints or facial recognition).

Integrate MFA into Your Login Flow:

 - Use a reputable MFA library or service that aligns with your development stack. For web applications, consider OAuth 2.0 or OpenID Connect with support for MFA.
 - Ensure the library or service is well-documented and actively maintained.

Set Up the Primary Authentication Factor:

 - Continue using passwords as the first factor but enforce strong password policies (e.g., minimum length, complexity requirements).

Implement the Secondary Authentication Factor:

 - For something you have: Send a one-time passcode (OTP) to the user's phone via SMS or an authentication app.
 - For something you are: Integrate biometric authentication if the platform supports it.

Fallback Mechanisms:

 - Provide options for users to authenticate through another method if their primary MFA method is unavailable (e.g., using backup codes).

User Registration and Recovery:

 - During registration or first login, prompt users to set up MFA.
 - Offer a clear, secure process for users to recover access to their account if they lose their MFA device.

Test and Deploy:

 - Rigorously test the MFA implementation to ensure it works smoothly across different devices and scenarios.
 - Monitor and review authentication logs for any unusual activities or failed login attempts.

Educate Users:

 - Provide guidance and training for users on setting up and using MFA. Explain the benefits and the process clearly to encourage adoption.

Compliance and Best Practices:

 - Adhere to relevant security standards and regulations for your industry that may mandate the use of MFA.
 - Regularly review and update your MFA implementation to align with emerging threats and new best practices.

Implementing MFA is a crucial step in securing access to sensitive systems and data. While it adds an extra step for users, the added security layer significantly outweighs the minor inconvenience, especially in environments susceptible to phishing attacks or where sensitive data is accessed. Always stay informed about the latest in MFA technology and security practices to ensure your implementation remains effective against evolving threats.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

Component: Login

Rec1. Implement Multi-Factor Authentication (MFA) C-LOGIN-CM1 Medium

PCI-DSS-v3.2.1: 10.9

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Logout

1 Recommended countermeasures

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High ☐ Not tested

- State: Recommended
- Description:
 - **Define Logging Standards:** Establish clear standards for logging, outlining the format, content, and level of detail required for security-related logs.
 - **Log Critical Events:** Log critical security events such as authentication attempts, authorization failures, and significant configuration changes.
 - **Centralized Logging:** Implement centralized logging to a secure, dedicated server or log management system for efficient monitoring and analysis.
 - **Regular Log Reviews:** Regularly review logs to identify and investigate any abnormal or suspicious activities, ensuring timely detection of potential security incidents.
 - **Automated Alerts:** Set up automated alerts based on predefined security thresholds to notify relevant personnel in real-time when suspicious activities are detected.
 - **Incident Response Plan:** Integrate security logging with an incident response plan, outlining the steps to be taken in the event of a security incident.
 - **User Activity Monitoring:** Monitor user activities and privilege escalations to detect and respond to unauthorized or suspicious behavior.
 - **Network Traffic Analysis:** Analyze network traffic logs to identify patterns indicative of security threats, such as unusual spikes or unexpected data transfers.
 - **Application-Level Logging:** Implement detailed application-level logging to capture events like failed login attempts, input validation failures, and other security-relevant actions.
 - **Regularly Update Log Sources:** Keep the list of log sources up-to-date to account for changes in the infrastructure, applications, and security landscape.
 - **Store Logs Securely:** Ensure that logs are stored securely, with restricted access, encryption, and integrity checks, to prevent tampering or unauthorized access.
 - **Logging in Compliance:** Align logging practices with relevant compliance standards and regulations to meet legal and industry-specific requirements.
 - **Security Information and Event Management (SIEM):** Consider implementing a SIEM solution to streamline log management, correlation, and analysis for more effective threat detection and response.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Logout

Rec1. Security Logging and Monitoring C-LOGOUT-V2-CNT-03 High

PCI-DSS-v3.2.1: 1.1

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

State: Recommended

Description:
Implement short session expiration times and require re-authentication for critical actions.
Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.
Monitor session activity and automatically log out users when unusual behavior is detected.
Educate users on avoiding insecure networks and using VPNs for secure connections.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

PCI-DSS-v3.2.1: 1.2

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

State: Recommended

Description:
Implement short session expiration times and require re-authentication for critical actions.
Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.
Monitor session activity and automatically log out users when unusual behavior is detected.
Educate users on avoiding insecure networks and using VPNs for secure connections.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

PCI-DSS-v3.2.1: 1.3

Implemented countermeasures

Below are the implemented countermeasures ("Imp") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Imp1. Rate limiting and throttling for MFA C-MFA-04 Low

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. Session security enhancement C-MFA-02 High Not tested

State: Recommended

Description:
Implement short session expiration times and require re-authentication for critical actions.
Use secure, encrypted cookies with HttpOnly and Secure flags to protect session tokens.
Monitor session activity and automatically log out users when unusual behavior is detected.
Educate users on avoiding insecure networks and using VPNs for secure connections.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. Session security enhancement C-MFA-02 High

PCI-DSS-v3.2.1: 1.1.5

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

State: Recommended

Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 12.8.2

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

State: Recommended

Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 9.9.3

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high ☐ Not tested

- State: ☒ Recommended
- Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 8.4

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high ☐ Not tested

- State: ☒ Recommended
- Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 12.4

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high ☐ Not tested

- State: ☒ Recommended
- Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 12.5

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

State: Recommended

Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 12.6

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

State: Recommended

Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 12.9

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: Multifactor Authentication (MFA)

1 Recommended countermeasures

Rec1. User training and awareness programs C-MFA-05 Very high Not tested

State: Recommended

Description:
Conduct regular training sessions on identifying and reporting phishing attempts and other social engineering attacks.
Implement a verification process for any requests to disable or bypass MFA, requiring multiple forms of authentication.
Use tools to simulate phishing attacks and measure user awareness and readiness.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: Multifactor Authentication (MFA)

Rec1. User training and awareness programs C-MFA-05 Very high

PCI-DSS-v3.2.1: 2.1

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication

C-USER-INTERFACE-MFA

Very high

☒ Not tested

State: Recommended

Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

- Select the types of authentication factors you will use. Common factors include:
 - Something you know (e.g., password or PIN)
 - Something you have (e.g., security token, smartphone app, or smart card)
 - Something you are (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

- Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
- Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

- Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
- Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

- Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
- Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

- Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
- Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

- Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
- Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

- Keep the MFA solution updated to defend against new vulnerabilities.
- Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

- Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
- Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

PCI-DSS-v3.2.1: 8.5

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication

C-USER-INTERFACE-MFA

Very high

☐ Not tested

State: Recommended

Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

- Select the types of authentication factors you will use. Common factors include:
 - Something you know (e.g., password or PIN)
 - Something you have (e.g., security token, smartphone app, or smart card)
 - Something you are (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

- Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
- Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

- Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
- Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

- Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
- Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

- Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
- Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

- Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
- Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

- Keep the MFA solution updated to defend against new vulnerabilities.
- Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

- Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
- Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

PCI-DSS-v3.2.1: 8.6

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqP") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high Not tested

- State: Recommended
- Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 1. **Something you know** (e.g., password or PIN)
 2. **Something you have** (e.g., security token, smartphone app, or smart card)
 3. **Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

 - Keep the MFA solution updated to defend against new vulnerabilities.
 - Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

 - Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
 - Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication C-USER-INTERFACE-MFA Very high

PCI-DSS-v3.2.1: 12.3

Non-compliant countermeasures

"Non-compliant" are those countermeasures that are recommended ("Rec"), rejected ("Rej") or not applicable ("N/A") in the model, but required by the standard. Required ("ReqF") countermeasures which tests have failed are also shown as non-compliant ones.

Component: User Interface

1 Recommended countermeasures

Rec1. Use multi-factor authentication

C-USER-INTERFACE-MFA

Very high

☒ Recommended ☐ Not tested

- State: Recommended
- Description:

Multi-factor authentication enhances security by requiring users to provide two or more verification factors to gain access to a resource, such as an application, online account, or a VPN. This significantly reduces the risk of unauthorized access resulting from compromised credentials.

Determine Appropriate Factors

 - Select the types of authentication factors you will use. Common factors include:
 - Something you know** (e.g., password or PIN)
 - Something you have** (e.g., security token, smartphone app, or smart card)
 - Something you are** (e.g., biometrics like fingerprints or facial recognition)

Choose an MFA Solution

 - Evaluate and choose an MFA solution that fits your security requirements and budget. Consider solutions from vendors like Google Authenticator, Authy, Microsoft Authenticator, or hardware-based tokens like YubiKey.
 - Ensure the solution integrates well with your existing security infrastructure and user management systems.

Implement MFA

 - Integrate MFA into your authentication process. This often involves modifying login workflows to include additional authentication steps.
 - Example setup for an online service:

Upon entering their username and password, users are prompted to enter a code generated by their authentication app or sent via SMS.

Enforce MFA Policies

 - Define and enforce policies around the use of MFA. Consider making MFA mandatory for accessing sensitive systems or data.
 - Set up policies to handle lost or unavailable authentication factors, such as backup codes or administrative resets.

Educate Users

 - Provide training and documentation to users on why MFA is important and how to use the chosen MFA method.
 - Offer support for users who might have difficulties setting up or using MFA.

Monitor and Review

 - Regularly review the effectiveness of your MFA implementation. Look for any security incidents that may indicate bypasses or failures in your MFA setup.
 - Monitor user compliance with MFA policies and address any areas where adherence may be lacking.

Maintain Security Standards

 - Keep the MFA solution updated to defend against new vulnerabilities.
 - Regularly evaluate new authentication methods and technologies to enhance security as needed.

Additional Considerations:

 - Be mindful of the user experience. Implementing MFA should balance security with usability. Overly complicated MFA processes can lead to user frustration or bypass attempts.
 - Consider the implications of relying on SMS-based authentication, as it can be less secure than other methods due to vulnerabilities in mobile networks.

By implementing MFA, you add a critical layer of security that protects against the most common threats to user accounts, particularly credential theft. This makes unauthorized access significantly more difficult, safeguarding your systems and data effectively.

Recommended countermeasures

Below are the recommended countermeasures ("Rec") by component and threat for standard reference.

Component: User Interface

Rec1. Use multi-factor authentication

C-USER-INTERFACE-MFA

Very high

End of Compliance report