Project P – Security Threats and Mitigation

1. Context

This is a sample threat model for a native Android travel app providing account management and travelcard ticket purchase features, with integration to electronic travel cards, third-party ticketing schemes and third-party payment gateways.

2. Threat Analysis

The following table lists threats, assesses a likelihood and impact for each, and identifies mitigations, or limitations for each. The 'Risk' and 'Impact' fields give, respectively, an assessment (High/Medium/Low/None) of the likelihood of the threat, and the impact if it happens.

Risks Relevant to App	Prob	Impact	Action / Mitigation
Attacker sees stuff read from card	L	-	N/A
Attacker sees what’s written to card (logging, etc.)	L	-	
Attacker with card secret key writes to card	L	H	N/A – Out of Scope for Developers. Operator issue.
Attacker sees/key logs payment details	M	H	Use a web view / sdk / tokenisation to mitigate risk.
Attacker sees payment details in memory	L	H	Use a web view / sdk / tokenisation to mitigate risk.
MITM attack sees payment details	L	H	Implement SSL Pinning (done).
MITM attack on travelcard comms	M	H	N/A – Out of Scope for Developers. Card supplier issue.
Card Cloning	L	-	Out of scope – app can’t differentiate.

SAMPLE SECURITY THREATS

Android screenshot of sensitive data (pin/password)	H	H	Prevent screenshot function on screens with sensitive data, e.g. payments.
Another user finds app with payment details/… set up	H	H	Use password mask CVC. OOS with Payment supplier. Check no repeat
Privacy – another user sees journey	M	L	Not in scope
Fraudulent user uses stored payment details to top up a different travel card	L	H	Out of scope – not storing details. Check no repeat
Payment with stolen credit card details (scheme op will block card)	H	H	Out of Scope – Payment Gateway / Operator issue. Terminal Server issue.
Rogue version of app (decompile apk)	M	L	Obfuscate code and have signed version. Google validation of authenticity. App validated by TS, App ID, Version Number, Google Token. Solution makes no assumption that app is genuine.
Test code (with credentials) debug code left in app	M	M	Clean-up app before publishing. Third-party code review.
MITM attack on Terminal Server comms	H	H	SSL Pinning (Done)
Same ticket fraudulently written to 2 cards	L	-	Can't be done. Travelcard security handles this.
Security issues with travelcard supplier library	M	M	Independent app security testing. Forced upgrade implemented.
Security issues with other libraries	M	M	Independent app security testing. Forced upgrade implemented.
Unknowns from O/S (new OS versions)	L	-	Process in place for ongoing product mgt.
Rooted devices	H	L	Unsupported but should be ok.
Logging of sensitive data	M	M	Code reviews and pen testing.
Local storage of sensitive data	L	-	Sensitive data not stored. Security /Pen Testing will validate.

Reading from device RAM	VL	VL	Reading from device RAM is now possible, but very unlikely.
Somebody spoofs a payment token	L	M	Single event unlikely and the terminal server will provide validation of payment vs ticket selection and delivery.
Social Engineering Issues	M	H	To be reviewed with each customer deployment.

Risks Relevant to Server	Prob	Impact	Action / Mitigation
MITM to Travelcard supplier Remote API	L	H	Certificate exchange between servers.
MITM to payment gateway	L	L	Gateway uses token based security. Single token issued by supplier for solution. Look at options for how token can be used – payment only. Check token can't be changed.
Denial of service attacks	H	H	Hosting service to cover this. Ensure hosting contract covers DDOS attacks.
Injection attacks	H	H	Use appropriate libraries. Pen testing feedback.
Malformed data attack	H	M	Use appropriate libraries. Reject invalid data.
Very Long data attack	M	M	Configure framework appropriately. Validate schema, e.g max length
Brute force attack on login passwords	L	M	Two factor authentication validated through third-party pen testing. Review hosting password policy. Don't keep passwords where we don't need them, e.g. live service.
Physical location – accesses machine as operator	L	H	Robust selection of reputable hosting service. Managed service.
Security risk for backup	L	H	Regular data backups by reputable hosting service. Managed service. ITSO trial needs backup (as any live deployment)
User uses same payment token twice	M	H	Can't be done – payment gateway will reject.
Rogue or competitor cloud app using backend	M	L	Make sure paid by server use, not app use / downloads. Using tokens so unlikely.

(Evolvi, transport API)			
Attacker gets passwords	L	H	Delegate to third-party? Operations procedures?
Attacker sees logs	M	M	Review of logging requirements. Pen / security testing & code reviews.
Sensitive information logged	M	H	Obfuscate any personal info in logs. E.g. email addresses.
Test APIs / endpoints left in solution	H	M	Remove any test endpoints on deployment. Need mechanism for turning on/off
Insecure libraries	L	H	Use Reliable Sources. Pen / Security Testing.
Discovered insecurities in OS/frameworks	L	H	Use Reliable Sources. Pen / Security Testing.
Misconfiguration of server leaves security holes	H	H	Pen testing
Clone server	L	L	Security on source code. SSL Pinning, other checks in place, would need client.
[Social engineering attack]	L	L	To review with customer.
Logging or storing personal data (DP Act) - Disclaimers / EULAs - how long keep ? (min/max)	L	H	Email addresses kept on server. Encrypt email addresses. Check requirements on storing logs – assume forever. Check encryption.
Server / database being hosted in the incorrect location / jurisdiction.	L	H	Locate according to requirements.
Attacker changes product list	H	H	Validated during transaction process. Token used as part of purchase process.
Payment with stolen credit card details (scheme op will block card)	H	M	Back office association between payment and smart card ISRN. Transaction ref. ECEBS / Service provider will handle stolen card issues to disable card. How do Developers inform Travelcard Supplier of hotlisted card?
Attacker changing code on server	L	H	Security on hosting server. Risk passed to hosting.
Http Support - MITM	L	L	Switch off http support. Disabled.

SAMPLE SECURITY THREATS

Spoofing a token to call for refunds	L	M	No refund functions at present.
Someone using journey planner service maliciously to make high volume of requests. Cost to Developers.	L	M	Calls authenticated to make sure that it is our app. Google Auth implemented. Accept risk of people using app to make requests.
User changing value of ticket purchase.	M	M	Ticket request type/value and payment must match – server needs to validate.
Handling of updates to scheme data risks offline error handling.	M	M	Establish and test process for handling / updating / validating scheme data.
Security Audit / Report / Log / File. Paper Trail being utilised.	M	M	Use third-party off-the-shelf product for maintaining server logs. Must not contain personal data. Logs contain no personal info.
Database access	M	H	Need two-factor authentication. / forced password changes / dongle. Managed hosting service.
Access to live console by customer.	-	-	Not initially required. Out of scope.
Developer staff member goes rogue	L	H	Only trusteed people see logging. Server admin. Two person authentication an option. Developer leavers removed from systems.