FVU Request System: Third-Party Integration & Security Report

PART 1: NON-TECHNICAL EXPLANATION

Background

The Forensic Video Unit, embedded within the Homicide and Missing Persons Bureau, required specialized request forms that the third-party ticketing system could not accommodate. This led to the development of a custom web-based solution that integrates seamlessly with the existing ticketing infrastructure.

System Overview

The FVU Request System is a web-based form application developed for the Forensic Video Unit within the Homicide and Missing Persons Bureau. Created to address limitations in the third-party ticketing system's field customization capabilities, the application functions as an electronic form system that:

- Runs entirely in the officer's web browser
- Stores no submitted case data
- Sends information directly to the ticketing system
- Uses browser storage only for convenience features (drafts and officer info)
- Provides three specialized forms:
  - **Analysis**: For forensic analysis of recovered video
  - **Upload**: For evidence upload to secure server (FVU as sole conduit)
  - **Recovery**: For on-scene CCTV recovery assistance requests

Data Flow

```
Officer's Browser → Static Forms (via SFTP) → Third-Party Ticketing System
                    (No case data stored)
```

Process:

1. Officer accesses form through third-party ticketing system interface
2. Static form files served from SFTP location
3. Form auto-saves drafts locally while typing
4. Officer clicks submit
5. Application generates:
   - PDF file (for legal disclosure requirements)
   - JSON file (for downstream efficiency applications)
6. Form fields and attachments sent to ticketing system
7. Form clears (draft removed, officer info retained)

Data Storage Locations

During Use:

- Active form data: Browser memory (RAM)
- Draft saves: Browser localStorage (expires after 7 days)
- Officer info: Browser localStorage (name, badge, phone, email only)

After Submission:

- Case data: Only in third-party ticketing system database
- Officer convenience info: Remains in browser for future use
- Application servers: No data stored
- FVU has PDF/JSON via ticketing system for their records

Deployment Options

The application can be deployed in two locations, with the standalone internet connection being the ideal solution for maintaining developer access while ensuring security. See Part 2 for detailed technical specifications.

Key Security Points

- Application created to address third-party system limitations
- No sensitive case data persists after submission
- All storage is browser-based and user-controlled
- Static files only - no server-side execution
- Direct submission eliminates intermediary risks
- SFTP access provided by third-party developer for maintenance

---

PART 2: TECHNICAL SPECIFICATION

Solution Overview

The FVU Request System addresses the third-party ticketing system's inability to provide specialized form fields through a client-side application that generates comprehensive documentation while integrating seamlessly with the existing infrastructure.

Core Innovation: While the ticketing system accepts only basic fields, the FVU application captures all specialized forensic video requirements and packages them into PDF (for disclosure) and JSON (for efficiency tools) attachments.

Architecture

```
┌─────────────────────────────────────────────────────────────┐
│                  Client Browser (Isolated)                  │
├─────────────────────────────────────────────────────────────┤
│  • Form Data (volatile memory)                              │
│  • PDF/JSON Generation (client-side)                        │
│  • localStorage: drafts (7-day expiry) & officer info       │
│  • No submitted case data persistence                       │
```

```
└──────────────────────────────────────────────────────────┘
                  │
                  │ POST (Multipart Form Data)
                  ↓
┌──────────────────────────────────────────────────────────┐
│              Third-Party Ticketing System                │
├──────────────────────────────────────────────────────────┤
│  • Serves static forms from SFTP location                │
│  • PHP Session Validation                                │
│  • Receives form fields + PDF/JSON attachments           │
│  • Manages ticket creation and tracking                  │
└──────────────────────────────────────────────────────────┘
```

Implementation Requirements

Integration Context:

- Developed to address field limitations in third-party ticketing system
- Seamless integration via SFTP and form submission
- Maintains existing ticketing workflow while adding specialized capabilities

Technology Stack:

- Frontend: Vanilla JavaScript (ES6+), HTML, CSS
- File Generation: Client-side (PDFMake for disclosure, JSON for efficiency tools)
- Server: PHP 7.4+ with session support
- Deployment: Static files via SFTP to third-party server

Required Form Fields (Third-Party System): Limited to fields the ticketing system can accept:

- `rName`, `requestingEmail`, `requestingPhone`
- `reqArea`, `fileDetails`, `rfsDetails`
- `occType`, `occDate`
- `fileAttachmentA` (PDF), `fileAttachmentB` (JSON)

Additional specialized fields are captured in the PDF/JSON attachments.

PHP Session Integration:

```php
<?php session_start(); ?>
<input type="hidden" name="session_verify" value="<?php echo session_id(); ?>">
```

Deployment Architecture Options

Based on SFTP location requirements:

1. **Peel Regional Police Corporate Network**
    - Status: Not viable (no external developer access)
2. **Direct on Third-Party Server**

- Limited maintenance flexibility

3. **Standalone Internet Connection** (Ideal Solution): Server on available standalone network
    - **Advantage**: Complete network isolation
    - **Benefit**: Full developer maintenance access
4. **Cloud Hosting with SFTP Bridge**
    - Professional hosting environment
    - Full developer control
    - Additional integration complexity
    - ~$5/month operational cost

Deployment Specifications

File Structure (on third-party server via SFTP):

```
/public_html/
├── index.php        # Landing page for form selection
├── analysis.php     # Forensic analysis request form
├── upload.php       # Evidence upload request form
├── recovery.php     # On-scene recovery request form
└── /assets/
    ├── /css/
    ├── /js/
    └── /images/
```

SFTP Configuration (Provided by Third-Party Developer):

```
# Access provided by ticketing system developer
User: fvu_deploy
Home: /var/www/homicidefvu/
Permissions: Read all, Write to /public_html/
Authentication: SSH key only
IP whitelist required
```

Security Analysis

Data Lifecycle:

1. **Input Phase**: Browser memory only
2. **Generation Phase**: PDF/JSON created in RAM
3. **Transmission**: Direct POST to endpoint
4. **Post-Submission**: Form cleared, drafts removed

Risk Profile:

- **Mitigated**: XSS (input sanitization), Session hijacking (PHP tokens)

- **Not Applicable**: SQL injection (no database), File upload attacks (no server storage)
- **Low Risk**: Client-side only processing, read-only application

Browser Storage Specifications

localStorage Usage (FVU-Controlled):

- **Draft System**: Auto-save with 7-day expiration
- **Officer Data**: Convenience storage for form pre-fill
- **Isolation**: Per-browser, not synchronized
- **Control**: User-clearable via browser settings

The FVU determines what convenience features to implement, independent of third-party system limitations.

Storage Keys:

```
fvu_draft_[formType]     // Draft data
fvu_officer_info         // Officer information
fvu_first_time           // First-use flag
```

Performance Requirements

- Browser: Chrome 80+, Firefox 75+, Safari 13+, Edge 80+
- Network: 1 Mbps minimum
- File sizes: ~100KB total per submission
- Session timeout: 24 minutes

Compliance

- No collection or retention of personal data by the application
- Direct transmission to authorized ticketing system only
- No cross-border data transfer through the application
- Minimal attack surface due to static file architecture
- All data handling complies with law enforcement evidence requirements

Recommended Deployment

Ideal Option: Standalone Internet Connection Deploy on the available standalone internet connection if server space available:

- Complete isolation from corporate network
- Full developer maintenance access via SFTP
- Third-party system retrieves files via SFTP
- No corporate IT dependencies
- Maintains evidence integrity requirements

Maintenance Workflow

1. Develop with .html files locally

2. Test functionality
3. Convert to .php files
4. Add session verification
5. Upload via SFTP to third-party server
6. Third-party system serves forms to end users
7. No database or server-side logic maintenance required

Conclusion

The FVU Request System is architecturally secure through:

- **Zero data persistence** of submitted information
- **Client-side processing** eliminating server vulnerabilities
- **Direct submission** removing intermediary risks
- **Static file deployment** preventing code execution risks
- **Browser storage** limited to convenience features only

The system operates as a secure form processor with no attack surface for data breaches, as no sensitive data is retained post-submission.

Scope Note: This security assessment covers the FVU Request System up to the point of integration with the third-party ticketing system. No affirmations are made regarding data security within the third-party system itself.