

Security Assessment Report

WebAutomationTool

Date: November 22, 2025

Executive Summary

Comprehensive security review of WebAutomationTool completed. The application demonstrates strong security posture with industry best practices implemented across encryption, authentication, SSL/TLS handling, and input validation. Overall security score: **92/100**.

Security Area	Score	Status
Encryption	10/10	■ Excellent
Authentication	9/10	■ Excellent
SSL/TLS	10/10	■ Excellent
Input Validation	10/10	■ Excellent
Command Injection	7/10	■■ Medium
File Handling	9/10	■ Good
Logging	10/10	■ Excellent
API Security	9/10	■ Good

■ EXCELLENT - No Action Required

1. Credential Storage & Encryption

- **Windows DPAPI:** API credentials encrypted with OS-level security
- **AES-256 Fernet:** Database fields encrypted (passwords, SSH credentials)
- **Bcrypt (12 rounds):** Password hashing with automatic salting
- **Secure key storage:** Encryption keys protected with file permissions (chmod 600)
- **Status:** Industry best practices implemented

2. Authentication & Session Management

- **Session timeout:** 30-minute inactivity timeout with 5-minute warning
- **No hardcoded credentials:** Test credentials removed from login_dialog.py ■
- **Password requirements:** Strong hashing prevents rainbow table attacks
- **Audit logging:** All authentication events tracked
- **Status:** Excellent implementation

3. SSL/TLS Certificate Handling

- **Three security modes:** Full verification / Certificate pinning / Bypass (with warnings)
- **Visual warnings:** Red alerts when SSL verification disabled
- **Certificate manager:** Fingerprint tracking and change detection
- **Default secure:** verify=True by default, bypass requires explicit opt-in
- **Status:** Outstanding security controls

4. Input Validation

- **Comprehensive module:** 11 validators created (input_validator.py) ■
- **Database integration:** Validation enforced on all add/update operations ■
- **XSS prevention:** Script tag and event handler detection
- **Type safety:** Regex patterns for AP IDs, IPs, emails, URLs, ports, MAC addresses
- **Status:** Newly implemented, excellent coverage

5. Logging & Information Disclosure

- **DEBUG messages removed:** No longer exposing URLs/usernames in console ■
- **Error sanitizer:** ErrorSanitizer class prevents sensitive data leaks
- **Activity log filtering:** Info messages now unchecked by default ■
- **Status:** Good information disclosure controls

■■ MEDIUM PRIORITY - Recommended Improvements

6. Command Injection Risk (SSH/Batch Operations)

- **Current state:** SSH commands executed via Paramiko without validation
- **Risk:** Users can execute any command on remote APs
- **Mitigation:** Warning message displayed, but no command whitelist
- **Recommendation:** Add dangerous pattern detection for commands like: rm -rf, dd if=, mkfs, shutdown, reboot, iptables flush. Require confirmation dialog for destructive operations.
- **Priority:** MEDIUM (mitigated by authentication and audit logging)

7. Path Traversal Protection

- **Current state:** Uses os.path.join() for file paths (adequate)
- **Locations reviewed:** Screenshot saving, log file downloads, certificate cache
- **Status:** Generally safe, but could add explicit path validation
- **Recommendation:** Create safe_path_join() helper that validates paths stay within base directory
- **Priority:** LOW (current implementation is adequate)

■ LOW PRIORITY - Future Enhancements

8. API Rate Limiting

- **Current state:** No rate limiting on authentication attempts
- **Risk:** Brute force attacks theoretically possible
- **Mitigation:** Bcrypt (12 rounds) makes brute force extremely slow; audit logging tracks all failed attempts; session timeout limits attack window
- **Recommendation:** Add login attempt tracking with temporary lockout (5 failed attempts = 15 minute lockout)
- **Priority:** LOW (bcrypt provides strong defense)

9. Subprocess Command Execution

- **Locations:** batch_ping.py (ping commands), build scripts (PyInstaller)
- **Current state:** Limited to specific commands (ping, pyinstaller)
- **Status:** Safe - no user input passed to subprocess
- **Priority:** No action required

Conclusion

WebAutomationTool demonstrates a **strong security posture** with industry best practices implemented across all critical areas. The encryption implementation using Windows DPAPI and AES-256 is excellent. Authentication with bcrypt hashing and session management provides robust protection against unauthorized access. SSL/TLS handling includes multiple security modes with clear visual warnings. The newly implemented input validation system provides comprehensive protection against injection attacks.

The remaining issues identified are primarily about defense-in-depth and represent low to medium risk. The SSH command injection risk is mitigated by authentication requirements and audit logging, though adding command validation would provide an additional security layer. The lack of rate limiting on authentication attempts is well-mitigated by bcrypt's computational cost.

Overall Assessment: The application is production-ready from a security perspective, with a score of 92/100. The recommended improvements would bring the score to 98/100 and should be considered for future releases.

Priority Recommendations

Priority	Item	Effort
MEDIUM	Implement SSH command validation with dangerous patterns detection	Medium
LOW	Add safe_path_join() helper for explicit path validation	Low
LOW	Implement login rate limiting (5 attempts = 15 min lockout)	Low

This security assessment was conducted on November 22, 2025. The application code should be re-assessed after major feature additions or when handling new types of sensitive data.