



AUTOMATED SECURITY ASSESSMENT

TestChecks

Application: TestChecks
Dates Assessed: June 2, 2020 through June 12, 2020
Report Date: June 15, 2020
Resource: Denis Kucinic (dkucinic@packetlabs.net)

Table of Contents

- 1 RISK LEVEL DESCRIPTIONS 3**
- 2 EXECUTIVE SUMMARY 4**
- 3 APPROACH..... 5**
 - 3.1 SCOPE..... 5
 - 3.1.1 *Application*..... 5
 - 3.2 CONSTRAINTS AND LIMITATIONS..... 5
- 4 METHODOLOGY 6**
- 5 TECHNICAL FINDINGS 10**

1 Risk Level Descriptions



Critical-risk finding pose an imminent risk to assets and sensitive information. Exploitation and discovery of these findings typically require minimal skill and often result in high-privileged access to the affected systems or information. Remediation of critical-risk findings are of high precedence and should not be left unaddressed under any circumstances.



High-risk findings pose an immediate risk to corporate assets or sensitive information. Exploitation of these items can directly lead to the compromise of systems, services or sensitive information. Exploitation is often possible with minimal effort and exploit code is likely to be publicly available or not required. It is recommended that these items be actioned as soon as possible.



Medium-risk findings may lead to a compromise of the environment or disclosure of sensitive information, but may require a significant amount of effort, time and complexity to successfully exploit. Medium risk findings should be actioned in a timely manner.



Low-risk findings have a small impact on the environment and a low likelihood of being exploited. It is generally recommended to address these risks at the lowest priority, occasionally the risk of these findings may be accepted and not actioned due to the limited impact and/or complexity to remediate.



Informational findings are observations made during the assessment which can be addressed with a lower priority. Informational findings typically do not pose a risk to the environment. This may include benign behavior such as bugs and broken functionality.



Remediated findings are findings where the identified vulnerabilities have been determined as fixed with no outstanding risk. Remediated findings do not pose a risk to the environment.



Automated Security Testing



2 Executive Summary

Packetlabs was engaged to perform an automated security assessment of TestChecks. The core objectives of this assessment were to evaluate the security of the web application through an automated web application scan.

Testing began on June 2, 2020 and completed on June 12, 2020. Overall, the environment was found to be at a **low** risk for compromise.

COMPONENT	KEY FINDINGS	RISK LEVEL
 WEB APPLICATION	No findings were identified during the automated scan.	 Low

The following recommendations outline strategic changes to address the root cause of the findings identified during this engagement:

- Conduct a maturity assessment to identify gaps in the TestChecks security program.
- Conduct regular table-top exercises to assess response capabilities to incidents. Ensure all third-party resources are included.
- Conduct annual application security assessments to provide continuous monitoring of the application.

3 Approach

Automated testing was performed on each of the in-scope systems and applications using the methodology outlined in section 4.

3.1 Scope

The scope of this assessment was an application security assessment of TestChecks in order to validate external risk effectively.

3.1.1 Application

For this assessment, the following URLs were considered in scope:

- <https://test.testchecks.com>

3.2 Constraints and Limitations









Our objective in this Penetration Test was to identify publicly known vulnerabilities residing in systems, applications and infrastructure components. While we have performed extensive testing and analysis, there is no assurance that all vulnerabilities were identified.





Prior to the execution of our testing, we have taken measures to ensure that all of our tools are up to date and are running with the latest feed updates and plugins. This report represents the state of the systems tested on a particular point in time.









4 Methodology

Our security testing methodology is derived from the OWASP Top 10:2017 and has been enhanced with current threats and our overall experience in the industry. Our methodology is comprehensive and has been broken up based on which areas can be tested with automation and those which require extensive manual testing.

Phase	Tasks Completed	Manual	Automated
Recon & Mapping	✓ Conduct search engine discovery and reconnaissance for information leakage		
	✓ Fingerprint web server		
	✓ Review web server metafiles for information leakage		
	✓ Enumerate applications on web servers		
	✓ Review webpage comments and metadata for information leakage		
	✓ Identify application entry points		
	✓ Identify technologies (e.g., web applications, frameworks or CMS platforms) used		
	✓ Map visible content and perform automated spidering of referenced content		
	✓ Test for debug parameters		
	✓ Discover hidden & default content		
Discovery	Configuration and Deploy Management Testing		
	✓ Test network/infrastructure configuration		
	✓ Test application platform configuration		
	✓ Test file extensions handling for sensitive information		
	✓ Analyze backup and unreferenced files for sensitive information		
	✓ Enumerate Infrastructure and application admin interfaces		
	✓ Test HTTP methods		
	✓ Test HTTP strict transport security		
	✓ Test RIA cross-domain policy		
	✓ Test for web server vulnerabilities		
✓ Testing for vulnerabilities in third-party applications (e.g. WordPress, Joomla, Drupal, SharePoint)			
Identity Management Testing	✓ Test role definitions		
	✓ Test user registration process		
	✓ Test account provisioning process		
	✓ Testing for account enumeration and guessable user account		
	✓ Testing for weak or unenforced username policy		
	✓ Test permissions of guest/training accounts		
✓ Test account suspension/resumption Process			

Phase	Tasks Completed	Manual	Automated
	Authentication Testing <ul style="list-style-type: none"> ✓ Testing for credentials transported over an encrypted channel ✓ Testing for default credentials ✓ Testing for a weak lockout mechanism ✓ Testing for bypassing authentication schema ✓ Test remember password functionality ✓ Testing for browser cache weakness ✓ Testing for weak password policy ✓ Testing for weak security question/answer ✓ Testing for weak password change or reset functionalities ✓ Testing for weaker authentication in alternative channel 		
	Authorization Testing <ul style="list-style-type: none"> ✓ Testing directory traversal/file include ✓ Testing for bypassing authorization schema ✓ Testing for privilege escalation ✓ Testing for insecure direct object references 		
	Session Management Testing <ul style="list-style-type: none"> ✓ Testing for bypassing session management schema ✓ Analyze cookies attributes (e.g., HttpOnly, Secure flags and scope) ✓ Testing for session fixation ✓ Testing for cross-site request forgery ✓ Testing for logout functionality ✓ Test session timeout ✓ Testing for session puzzling ✓ Persistent cookies ✓ Test tokens for predictability ✓ Check for insecure transmission of session tokens 		
	Input Validation Testing <ul style="list-style-type: none"> ✓ Fuzz all input parameters ✓ Testing for reflected cross-site scripting ✓ Testing for stored cross-site scripting ✓ Testing for HTTP verb tampering ✓ Testing for HTTP parameter pollution ✓ Testing for HTTP splitting/smuggling ✓ Testing for SQL injection (Oracle, MySQL, MsSQL, PostgreSQL, Microsoft Access, NoSQL) ✓ Testing for LDAP injection ✓ Testing for ORM injection ✓ Testing for XML injection ✓ Testing for SSI injection ✓ Testing for XPath injection 		

Phase	Tasks Completed	Manual	Automated
	<ul style="list-style-type: none"> ✓ Testing for IMAP/SMTP injection ✓ Testing for code injection ✓ Testing for local file inclusion ✓ Testing for remote file inclusion ✓ Testing for command injection ✓ Testing for native software flaws (buffer overflow, integer bugs, format strings) ✓ Testing for incubated vulnerabilities ✓ Testing for open redirection ✓ Testing for SOAP injection 		
	Error Handling <ul style="list-style-type: none"> ✓ Analysis of error codes ✓ Analysis of stack traces 		
	Cryptography <ul style="list-style-type: none"> ✓ Testing for weak SSL/TLS ciphers, insufficient transport layer protection ✓ Testing for padding oracle ✓ Testing for sensitive information sent via unencrypted channels ✓ Testing for CBC bit flipping ✓ Testing for hash length extension 		
	Business Logic Testing <ul style="list-style-type: none"> ✓ Identify the logic attack surface ✓ Test business logic data validation ✓ Test the ability to forge requests ✓ Test integrity checks ✓ Test for process timing (race conditions, TOCTOU) ✓ Testing for the circumvention of workflows ✓ Test defenses against application misuse ✓ Test upload of unexpected file types ✓ Test upload of malicious files ✓ Analyze SSL responses for caching of sensitive content ✓ Analyze content for sensitive data in URL parameters ✓ Testing for reliance on client-side input validation ✓ Testing of trust boundaries 		
	Client Side Testing <ul style="list-style-type: none"> ✓ Testing for DOM-based cross-site scripting ✓ Testing for JavaScript execution ✓ Testing for HTML injection ✓ Testing for client-side open redirection ✓ Testing for CSS injection ✓ Testing for client-side resource manipulation ✓ Test cross-origin resource sharing 		

Phase	Tasks Completed	Manual	Automated
	<ul style="list-style-type: none"> ✓ Testing for cross-site flashing ✓ Testing for clickjacking ✓ Testing WebSockets ✓ Test web messaging ✓ Test local storage ✓ Testing of thick-client components (Java, ActiveX, Flash) 		
	<p>Audit: WordPress</p> <ul style="list-style-type: none"> ✓ Test for exposed admin portal ✓ Analyze plugins and themes ✓ Test for XMLRPC exposure ✓ Username Enumeration ✓ Password policy and multifactor authentication settings ✓ Missing Patches ✓ Various WordPress security checks 		
	<p>Audit: JavaScript</p> <ul style="list-style-type: none"> ✓ Test for overly permissive Content Security Policy (CSP) ✓ Test for sub resource integrity checks ✓ Testing for linking to third-party Code ✓ Testing for advertisement and analytics on critical flows ✓ Testing for critical flows isolation 		
<p>Exploitation **</p>	<ul style="list-style-type: none"> ✓ Leverage findings from previous phases in order to expand foothold in the environment. ✓ Execute a number of exploits focusing on: <ul style="list-style-type: none"> ○ bypass attacks ○ injection attacks ○ session attacks ✓ Attempt to escalate privileges and/or gain unauthorized access ✓ Attempt to pivot from compromised systems to other internal systems. 		
<p>Reporting</p>	<ul style="list-style-type: none"> ✓ A draft detailed report outlining findings coupled with control recommendations including an executive summary outlining the overall state of the application. ✓ Document steps to reproduce findings to ensure application developers can validate remediation efforts prior to retesting. ✓ Conduct root cause analysis of findings outlining common themes observed with recommendations to improve security within the environment. 		



Findings



5 Technical Findings

Overall findings and risk-level has been outlined in the following table with each component detailed in the section below. The application is at a **low**-risk for compromise given the lack of findings.

Findings Breakdown

COMPONENT	FINDINGS	RISK LEVEL
 WEB APPLICATION	<ul style="list-style-type: none">No findings identified.	 Low