

# SECURITY

## ASSESSMENT

**App Name**  
**CompanyName**

**Document Name:**  
**Version:**  
**Date:**  
**Classification:**

**VT\_DASA\_DesktopAPP\_V1.0**  
**1.0**  
**[Date]**  
**Strictly Confidential**

## Document History

#	Date	Purpose of Version	Author	Reviewer
1.0	DD/MM/YYYY	Initial Security Assessment	Tester	David NG

## Distribution List – Customer Name

Name	Role
xxx	Requester
xxx	Requester

Veiron has been engaged to assess the xxx desktop application of COMPANY NAME as of [Date]. The mitigation of audit findings is the responsibility of COMPANY NAME.

Accepted and agreed to:

COMPANY NAME

COMPANY CONTACT

.....

(Authorized Signature & Date)

Accepted and agreed to:

**Veiron**

David Ng

.....

(Authorized Signature & Date)

## Contents

1	Foreword.....	3
2	Executive Summary .....	3
2.1	Scope.....	3
2.2	Methodology .....	4
2.3	Scoring System .....	4
2.4	Vulnerability Review .....	5
3	Vulnerabilities and Remediation .....	6
3.1	Legend .....	6
3.2	Vulnerability Summary Table.....	7
4	Desktop App Detailed Report .....	8
4.1	Information Gathering and Static Analysis.....	8
4.2	Dynamic Analysis.....	11
4.3	Runtime Analysis .....	14
4.4	Data Storage .....	16
5	WebAPI/Backend Detailed Report.....	18
5.1	Information Gathering / Information Leakage .....	18
5.1.1	Open Ports from the Internet .....	21
5.2	Encryption.....	22
5.2.1	Certificate Details .....	24
5.3	Authentication.....	25
5.4	Authorization .....	27
5.5	Credential Management .....	28
5.6	Data Validation .....	29
5.7	Web App & Server Security Misconfiguration .....	32
6	OWASP Desktop Top 10 – 2016 .....	33
7	Glossary .....	34

## 1 Foreword

Veilron was requested to assess the security posture of the Customer's name's applications by performing a desktop and web application penetration test that uses automated and manual IT security testing methodologies.

Veilron conducted the security analysis starting [Date] until [Date\_End] through the Internet. This report summarizes the results of tests performed on the target systems, including the vulnerability assessment and recommended remediation actions.

## 2 Executive Summary

### 2.1 Scope

The following apps/binaries were tested during the security assessment

App Name	Platform	Executable name
Desktop	Windows and Linux	Desktop.app/

The app is downloadable from [source] and accessible on all Windows and Linux devices.

The end-user interfaces with a backend system via the desktop application. The penetration test included the assessment of the backend and was conducted from the Internet with a greybox approach on a test environment (replica of the production server).

The specific hosts and subnets that were tested are indicated below:

IP address	Hostname/Service
x.x.x.x	www.myservice.com

This report summarizes the results of the retest of previously discovered vulnerabilities during the initial security assessment. As per the rules of engagement, vulnerabilities that have been properly closed have been removed from this report."

The following user accounts were provided to conduct the penetration test:

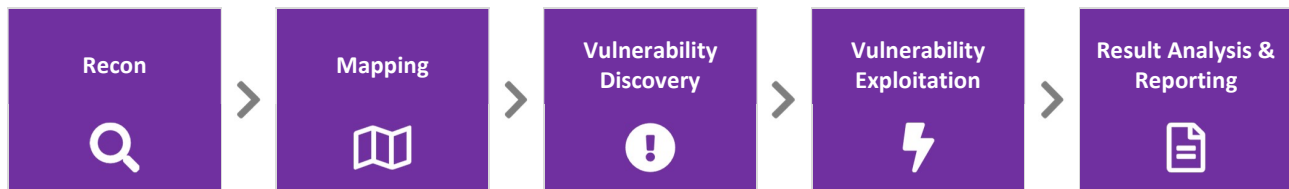
Username	Role / Description
Veilron1	Standard user (from Company ABC)
Veilron2	Standard user (from Company DEF)
AdminVEILRON1	Administrator user (from Company ABC)
AdminVEILRON2	Administrator user (from Company DEF)

Since Veilron did not receive two user accounts per user role, it was not possible to test fully authorization mechanism of the application.

As per the rules of engagement, the following attacks were not in scope of the penetration test: Denial of Service, Password Bruteforce, Social Engineering, and Man-in-the-Middle attacks.




## 2.2 Methodology

At Veilron, our penetration testing methodology builds on the approach outlined in the OWASP Testing Guide, Open Source Security Testing Methodology Manual (OSSTMM) and Penetration Testing Execution Standard (PTES).



## 2.3 Scoring System

Veilron uses the Common Vulnerability Scoring System v3.0 to rate identified weaknesses and vulnerabilities:

Severity Rating	CVSS Score	Description
<b>Info</b>	0.0	There is no direct impact on the target. The weakness usually expands the attacking surface or helps the attacker to increase the foothold in the system.
<b>Low</b>	0.1 - 3.9	This rating is given to all other issues that have a security impact. These are the types of vulnerabilities that are believed to require unlikely circumstances to be able to be exploited, or where a successful exploit would give minimal consequences.
 <b>Medium</b>	4.0 - 6.9	This rating is given to flaws that may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances. These are the types of vulnerabilities that could have had a critical impact but are less easily exploited based on a technical evaluation of the flaw, or affect unlikely configurations.
 <b>High</b>	7.0 - 8.9	This rating is given to flaws that can easily compromise the confidentiality, integrity, or availability of resources. These are the types of vulnerabilities that allow local users to gain privileges, allow unauthenticated remote users to view resources that should otherwise be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow remote users to cause a denial of service.
 <b>Critical</b>	9.0 - 10	This rating is given to flaws that could be easily exploited by a remote unauthenticated attacker and lead to system compromise (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms.

The severity rating is calculated based on several factors such as:

### Exploitability Metrics:

- Attack Vector (AV)
- Attack Complexity (AC)
- Privileges Required (PR)
- User Interaction (UI)

### Scope:

- Unchanged (U)
- Changed (C)

### Impact Metrics:

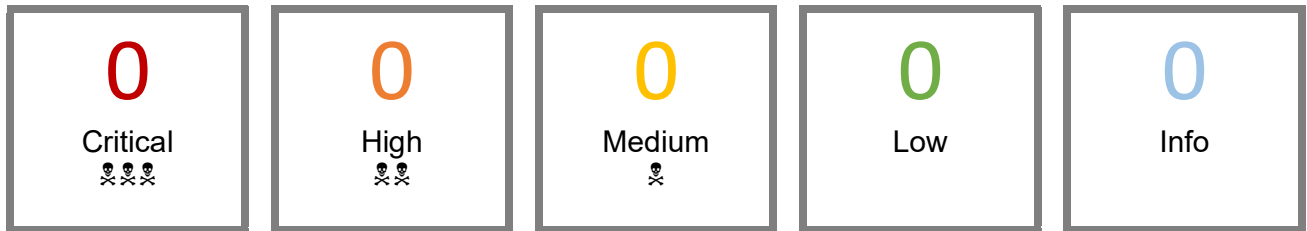
- Confidentiality Impact (C)
- Integrity Impact (I)
- Availability Impact (A)



For more information about CVSS Scoring System: <https://www.first.org/cvss/specification-document>.

## 2.4 Vulnerability Review

The following table displays the number of open vulnerabilities found on the system per severity (see chapter 3.2 for further details):



This is where you write the exec summary.

### Recommendations:






It is recommended that all vulnerabilities rated as high severity level be fixed in the short term, while medium or low rated threats can be fixed in the medium term. The following mitigation steps should be considered:

Click or tap here to enter text.

The detailed technical mitigation recommendations are to be found in Chapter 3.2

## 3 Vulnerabilities and Remediation

### 3.1 Legend

No.	Severity	Reference	Weakness	Threat	Solution	Comments
Issue number	<div>  <b>Critical</b> </div> <div>  <b>High</b> </div> <div>  <b>Medium</b> </div> <div>  <b>Low</b> </div> <div>  <b>Info</b> </div>	Reference of:  CWE CVE OWASP Detailed Section	Explains the vulnerability or weakness found during testing.	Explains what Veilron could do with the vulnerability if exploitable or what could happen if the weakness is exploited by an attacker.	Recommendation on how to correct the vulnerability.	This section contains:  status of the vulnerability (Open or Closed)  CVSS Score  Fix difficulty Level:  <b>Fix Difficulty:</b> <span>Complex</span>  <b>Fix Difficulty:</b> <span>Medium</span>  <b>Fix Difficulty:</b> <span>Quick Win</span>

**Complex** Requires major changes to the web application's code and architecture or requires the implementation of another technology (e.g. Two Factor Authentication)




**Medium** Requires a small structural change in the architecture, but it is of easy implementation and affordable by small development teams.

**Quick Win** Requires minor changes to the web application's code or to the web server's configuration

**Note:** Items that are marked as "Info" have been tested as per web application best practice configuration. Even if marked as "Info", Veilron recommends on implementing our recommendations to increase the security posture of the application.

## 3.2 Vulnerability Summary Table

The following table shows the vulnerabilities found on the system (see [chapter 4](#) and [chapter 5](#) for further details) during the security assessment:







#	Severity	Ref.	Weakness	Threat	Solution	Comments
1.						
2.						
3.						
4.						
5.						
















## 4 Desktop App Detailed Report







### 4.1 Information Gathering and Static Analysis

Tools	<ul style="list-style-type: none"> <li>• Nmap</li> <li>• Burp Suite</li> <li>• Nikto</li> <li>• Google Search/Newsgroups</li> <li>• Strings</li> <li>• CFF Explorer</li> <li>• PEid</li> <li>• DLLSpy</li> <li>• Robber</li> </ul>
-------	--

#	Description	Expectations	Results	Pass  Fail 
1.	Is it possible to find security-relevant information about the application on the Internet? <ul style="list-style-type: none"> <li>• Newsgroups</li> <li>• Social Networks</li> <li>• Search Engines</li> <li>• Etc.</li> </ul>	No	<b>Windows:</b> <b>Linux:</b>	 
2.	Is it possible to identify the product and version of libraries and/or framework in use?	Eventually the product but not the version.	<b>Windows:</b> <b>Linux:</b>	 

3.	If product's version is identifiable, is it up-to-date?	Product is up-to-date.	Windows: Linux:	 
4.	If product's version is identifiable, are there any known vulnerabilities?	No known vulnerabilities or known patch available.	Windows: Linux:	 
5.	Are there any identifiable hard-coded secrets within the application such as API keys, credentials, or proprietary business logic?	No	Windows: Linux:	 
6.	Are there any easy to identify misconfigurations within the application found within the configuration files?	No	Windows: Linux:	 
7.	For what architectures was the app compiled for?	N/A	Windows: Linux:	 

8.	Was it possible to retrieve credentials for the app by observing the dump?	No	Windows: Linux:	 
9.	Has the binary been compiled as a Position Independent Executable?	Yes	Windows: Linux:	 
10.	Has the binary a Stack Smashing Protection in place?	Yes	Windows: Linux:	 
11.	<p>Is it possible to identify valid usernames by checking how the application reacts when providing valid and invalid usernames?</p> <ul style="list-style-type: none"> <li>Login Error Messages</li> <li>URL-based Errors</li> <li>Page differences</li> </ul> <p>Harvesting Areas:</p> <ul style="list-style-type: none"> <li>Password Reset</li> <li>Password Recovery</li> <li>User Sign-up</li> <li>Messaging systems</li> </ul>	No	Windows: Linux:	 













12.	What is the minimum supported OS version?	For Linux: 24 or later For Windows:	<b>Windows:</b> <b>Linux:</b>	 
13.	What are the libraries utilized by the application?		<b>Windows:</b> <b>Linux:</b>	 
14.	Any custom libraries detected? Are they vulnerable to potential hijacking?		<b>Windows:</b> <b>Linux:</b>	 



## 4.2 Dynamic Analysis

### Tools

- Browser
- Burp Suite
- Nikto
- Regshot
- Process Explorer / Monitor











#	Description	Expectations	Results	Pass  Fail 
1.	Does the App have Internet transactions?	N/A	<b>Windows:</b> <b>Linux:</b>	 











2.	Do error messages disclose too many information about the App?	No	Windows: Linux:	 
3.	Does the App disclose sensitive information when communicating with the server?	No sensitive communication is sent to the server without proper prior authentication.	Windows: Linux:	 
4.	Does the App utilize trusted source only for the user input?	Yes, always.	Windows: Linux:	 
5.	Does the application send push notification to the user?	N/A	Windows: Linux:	 
6.	Do push notifications allow users to modify/delete data directly from the alert message?	No	Windows: Linux:	 
7.	Are sensitive information stored in plain text?	No	Windows: Linux:	 

8.	Is it possible to identify business logic flaws?	No	<b>Windows:</b> <b>Linux:</b>	 
----	--	----	----------------------------------	---

## 4.3 Runtime Analysis

<b>Tools</b>	<ul style="list-style-type: none"> <li>• Ghidra</li> <li>• Ida Pro</li> <li>• Native Logger</li> <li>• Regshot</li> <li>• Process Explorer / Monitor</li> </ul>
--------------	---











#	Description	Expectations	Results	Pass  Fail 
1.	Is it possible to identify buffer overflows?	No	Windows: Linux:	 
2.	Is it possible to identify client-side injections?	No	Windows: Linux:	 
3.	Is the memory correctly managed at runtime?	Yes	Windows: Linux:	 
4.	Is it possible to perform runtime injections?	No	Windows: Linux:	 







5.	Is it possible to replicate the homepage of the App in an external frame?	No	<b>Windows:</b> <b>Linux:</b>	 
6.	Is it possible to create a method swizzling hook in order to create a "stealer" app?	No	<b>Windows:</b> <b>Linux:</b>	 
7.	Is it possible to decompile and easily read the source code of the App?	No, the code must be obfuscated.	<b>Windows:</b> <b>Linux:</b>	 
8.	For apps with sensitive transactions, is there an SSL pinning mechanism in place?	Apps with sensitive transaction must have SSL pinning.	<b>Windows:</b> <b>Linux:</b>	 
9.	Are there any changes in the registry during the application runtime?		<b>Windows:</b> <b>Linux:</b>	 



## 4.4 Data Storage

<b>Tools</b>	<ul style="list-style-type: none"> <li>• Browser</li> <li>• Burp Suite</li> <li>• Nikto</li> </ul>
--------------	--













#	Description	Expectations	Results	Pass  Fail 
1.	Does the application perform logging of sensitive operations?	Yes	<b>Windows:</b> <b>Linux:</b>	 
2.	Are log files securely stored?	Yes	<b>Windows:</b> <b>Linux:</b>	 
3.	Are log sent over the internet or stored locally?	If logs are sent to an external service, user must be informed, and has to be able to choose whether to disable such feature, for its own privacy.	<b>Windows:</b> <b>Linux:</b>	 
4.	Is it possible to identify a Database?	N/A	<b>Windows:</b> <b>Linux:</b>	 







5.	Does the App create files to be stored locally?	N/A	<b>Windows:</b> <b>Linux:</b>	 
6.	Do locally created files disclose sensitive information about the App in clear text?	No, if sensitive information need to be stored they must be encrypted first	<b>Windows:</b> <b>Linux:</b>	 
7.	Are sensitive information cached locally?	No	<b>Windows:</b> <b>Linux:</b>	 





## 5 WebAPI/Backend Detailed Report

### 5.1 Information Gathering / Information Leakage

<b>Tools</b>	<ul style="list-style-type: none"> <li>• Nmap</li> <li>• Burp Suite</li> <li>• Nikto</li> <li>• HTTPPrint</li> <li>• Google Search/Newsgroups</li> <li>• CeWL (Custom Word List Generator)</li> </ul>
--------------	---

#	Description	Expectations	Results	Pass  Fail 
1.	Is it possible to identify the product and version?	Eventually the product but not the version.		 
2.	If product's version is identifiable, is it up-to-date?	Product is up-to-date.		  <b>INFO</b>
3.	If product's version is identifiable, are there any known vulnerabilities?	No known vulnerabilities or known patch available.		  <b>INFO</b>
4.	Is it possible to gather information about the application owner via publicly available tools (e.g. WHOIS)?	No		 
5.	Is it possible to find information about the application on the Internet? <ul style="list-style-type: none"> <li>• Newsgroups</li> <li>• Social Networks</li> <li>• Search Engines</li> <li>• Etc.</li> </ul>	No		 

#	Description	Expectations	Results	Pass  Fail 
6.	Is it possible to access default pages (e.g. Welcome page from Apache)?	No		 
7.	Are other ports than HTTP (tcp-80) or HTTPS (tcp-443) opened on the server?	No		 
8.	Is it possible to identify a load balancer in-line of the Web Application? If yes, is it possible to identify the load balancer's vendor?  <u>Various Methods:</u> <ul style="list-style-type: none"> <li>• URL Analysis</li> <li>• Timestamp Analysis</li> <li>• Last modified values comparison</li> <li>• Load Balancer cookie detection</li> <li>• HTTPS differences</li> <li>• HTML source code discrepancies</li> </ul>	It is strongly recommended to hide the load balancer's vendor.		<b>INFO</b>
9.	Is it possible to identify a WAF in-line of the Web Application? If yes, is it possible to identify the WAF's vendor?  <u>Various Methods:</u> <ul style="list-style-type: none"> <li>• URL Analysis</li> <li>• Error messages fingerprinting</li> <li>• WAF cookie detection</li> </ul>	It is strongly recommended to hide the WAF's vendor.		<b>INFO</b>

#	Description	Expectations	Results	Pass  Fail 
10.	Is it possible to find sensitive data while spidering the website? <ul style="list-style-type: none"> <li>• Potential Security weaknesses in code or comments</li> <li>• Email addresses, name, phone number, etc.</li> <li>• List of keywords for password-guessing attacks</li> <li>• Confidential data</li> <li>• Disabled functionality</li> <li>• Backup files</li> </ul>	No		 

### 5.1.1 Open Ports from the Internet



















The server has only the ports 80 (http) and 443 (https) open. No unnecessary services have been discovered on the target server.









PORT	STATE	SERVICE
80/tcp	open	http
443/tcp	open	ssl/http

## 5.2 Encryption

### Tools

- Burp Suite
- Browser
- OpenSSL
- SSLThing
- THC SSL Check

#	Description	Expectations	Results	Pass  Fail 
1.	Encryption (SSL/TLS) is enabled?	Yes		 
2.	Valid certificate information?	<b>Yes</b>		 
3.	Valid Issuer?	<b>Yes</b>		 
4.	Valid Expiration date?	Yes		 
5.	Valid Domain name?	Yes		 
6.	Valid SSL Certificate strength?	Minimum requirements: <b>Public Key:</b> > 2048 Bits <b>Exponent:</b> 0x10001 <b>Signature:</b> sha256withRSAEncryption		 
7.	Export and Cipher verification?	Strong ciphers are: Symmetric: >116 bit Asymmetric: >1024 bit		 
8.	SSLv2 or earlier Supported?	No		 

#	Description	Expectations	Results	Pass  Fail 
9.	SSLv3 / TLSv1.0 Supported?	Not recommended.		 
10.	TLSv1.1 and 1.2 Supported?	Yes		 
11.	Weak ciphers are supported?	Only strong ciphers supported. Export-, weak and null ciphers are disabled.		 



































### 5.2.1 Certificate Details

Certificate Content	
SHA1 Fingerprint	Click or tap here to enter text.
Common Name	Click or tap here to enter text.
Issuer	Click or tap here to enter text.
Serial Number	Click or tap here to enter text.
Not Before	Click or tap here to enter text.
Not After	Click or tap here to enter text.
Signature Algorithm	Click or tap here to enter text.
Public Key Algorithm & Key Size	Click or tap here to enter text.
Deflate Compression	
Compression Status	Click or tap here to enter text.
Session Renegotiation	
Client-initiated Renegotiations	Click or tap here to enter text.
Secure Renegotiation	Click or tap here to enter text.
OpenSSL Heartbleed	
Vulnerable?	Click or tap here to enter text.
ROBOT Attack	
Vulnerable?	Click or tap here to enter text.
Downgrade Attack	
Vulnerable?	Click or tap here to enter text.
OpenSSL CCS Injection	
Vulnerable?	Click or tap here to enter text.

## 5.3 Authentication

<b>Tools</b>	<ul style="list-style-type: none"> <li>• Browser</li> <li>• Burp Suite</li> <li>• Nikto</li> </ul>
--------------	--











#	Description	Expectations	Results	Pass  Fail 
1.	Authentication only available using encrypted connection?	Yes		 
2.	How many authentication factors are available?	<u>Confidential Data stored:</u> At least two-factor authentication.  <u>Internal Data stored:</u> At least username + Password.	x	 
3.	Basic Authentication used?	No		 
4.	Digest Authentication used?	No		 
5.	Which HTTP Method is used to send Login data?	<b>POST:</b> Yes <b>GET:</b> No		 

#	Description	Expectations	Results	Pass  Fail 
6.	Is it possible to identify valid usernames by checking how the application reacts when providing valid and invalid usernames?  <ul style="list-style-type: none"> <li>Login Error Messages</li> <li>URL-based Errors</li> <li>Page differences</li> </ul> Harvesting Areas: <ul style="list-style-type: none"> <li>Password Reset</li> <li>Password Recovery</li> <li>User Sign-up</li> </ul> Messaging systems	No	<<>>	 
7.	What happens after entering several false passwords for the same account?	Timeout or Lockout		 
8.	Is it possible to login to the application using default credentials?	No		 
9.	Is it possible to login using guessable user accounts?	No		 
10.	Does the application have logout mechanism?	Yes		 
11.	Which HTTP Method is used to Logout?	POST: Yes GET: No		 
12.	For financial/banking App: Is there a session inactivity timeout?	Yes, set to max of 15 minutes.		 
13.	What happens during a timeout?	The user must re-authenticate. Furthermore, all session content must be immediately deleted.		 
14.	Is it possible to access internal data without being authenticated?	No		 

## 5.4 Authorization

















### Tools

- **Browser**
- **Burp Suite**

#	Description	Expectations	Results	Pass  Fail 
1.	Is vertical privilege escalation possible (obtain a higher level of access)?	No		 
2.	Is horizontal privilege escalation possible (User A is able to access User B's data)?	No		 
3.	Is it possible to have multiple sessions opened (login multiple times)?	No		 
4.	Is the application vulnerable to Path traversal? Is it possible to leave the web root?  Examples: <ul style="list-style-type: none"> <li>• vulnsite /../../etc/passwd</li> <li>• vulnsite/../../etc/passwd%00</li> <li>• vulnsite/../../htpasswd</li> <li>• vulnsite/..%u2216..%u2216etc/passwd</li> </ul>	No		 









## 5.5 Credential Management











<b>Tools</b>	<ul style="list-style-type: none"> <li>• Nmap</li> <li>• Burp Suite</li> <li>• Nikto</li> <li>• HTTPPrint</li> <li>• Google Search/Newsgroups</li> <li>• CeWL (Custom Word List Generator)</li> </ul>
--------------	---







#	Description	Expectations	Results	Pass  Fail 
1.	What is the password policy?	Passwords should be at least eight (8) characters long.  Password characters should be a combination of alphanumeric characters.		 
2.	Is it possible to use various alphanumeric characters in the password (e.g. & / + -)?	Yes		 
3.	What is the maximal length allowed of a password?	Typical maximum length is 128 characters.		 
4.	Is it possible to change the password?	Yes		 
5.	Observe the error message if password couldn't be changed successfully?	The form fields are empty.		 
6.	Is it possible to reset a forgotten password?	N/A		 
7.	How long is the token valid to reset a password?	Maximum 20 minutes.		 

## 5.6 Data Validation

<b>Tools</b>	<ul style="list-style-type: none"> <li>• Burp Suite</li> <li>• Nikto</li> <li>• W3AF</li> <li>• Sqlmap</li> <li>• Flare</li> <li>• BeEF</li> </ul>
--------------	--

#	Description	Expectations	Results	Pass  Fail 
1.	Are the user's inputs properly parsed or encoded for output?	Yes		 
2.	Where is the user input parsed?	<p>The user input is parsed on the server's side.</p> <p>Client-side parsing can be implemented as long as server-side parsing is implemented too.</p>		 
3.	Does the application validate length for input?	The server side should have length restriction for input.		 















#	Description	Expectations	Results	Pass  Fail 
4.	<p>Is the application vulnerable to Cross-Site Scripting (XSS)?</p> <ul style="list-style-type: none"> <li>• <code>"*())/&amp;&lt;&gt;%</code></li> <li>• <code>&lt;script&gt;alert(test)&lt;/script&gt;</code></li> <li>• <code>&lt;IMG onmouseover="alert('test')"&gt;</code></li> <li>• <code>'&gt;&lt;script&gt;alert(test)&lt;/script&gt;</code></li> <li>• <code>&lt;IMG SRC="javascript:alert('test');"&gt;</code></li> <li>• <code>&lt;IMG SRC=javascript:alert('test')&gt;</code></li> <li>• <code>&lt;IMG SRC=JaVaScRiPt:alert('test')&gt;</code></li> <li>• <code>&lt;IMG ""'"&gt;&lt;SCRIPT&gt;alert("test")&lt;/SCRIPT&gt;"&gt;</code></li> </ul> <p>Full list can be found at: <a href="https://owasp.org">OWASP.ORG</a></p>	<p>The application should validate input to prevent XSS.</p> <p>Preferred filtering solution is whitelists:</p> <ul style="list-style-type: none"> <li>• Filtering based on “known goods”</li> <li>• Better security</li> </ul> <p>Another filtering solution is blacklists:</p> <ul style="list-style-type: none"> <li>• Filtering based on “known bads”</li> <li>• Easier to bypass</li> </ul> <p>The application should also encode output of characters that causes XSS:</p> <ul style="list-style-type: none"> <li>• <code>&amp;</code> <code>&amp;amp;</code></li> <li>• <code>"</code> <code>&amp;quot;</code></li> <li>• <code>'</code> <code>&amp;apos;</code></li> <li>• <code>&lt;</code> <code>&amp;lt;</code></li> <li>• <code>&gt;</code> <code>&amp;gt;</code></li> </ul>		 
5.	<p>Is the application vulnerable to SQL injection (SQLi)?</p> <ul style="list-style-type: none"> <li>• <code>'</code></li> <li>• <code>' OR 1=1 #</code></li> <li>• <code>' OR 1=1 --</code></li> <li>• <code>' OR 2=2 --</code></li> <li>• <code>' OR dummy=dummy --</code></li> <li>• <code>' OR 1=1 FOR XML RAW; --</code></li> <li>• <code>' OR 1='1</code></li> <li>• <code>%C0%A7+OR+1%3D1+%C0%AD%C0%AD+</code></li> </ul> <p>Check all inputs including cookies &amp; headers</p>	<p>Primary Defenses:</p> <ul style="list-style-type: none"> <li>• Option #1: Use of prepared statements (Parameterized Queries).</li> <li>• Option #2: Use of stored procedures.</li> <li>• Option #3: Escaping all user supplied input.</li> </ul> <p>Additional Defenses:</p> <ul style="list-style-type: none"> <li>• Also enforce least privilege.</li> </ul> <p>Also Perform:</p> <ul style="list-style-type: none"> <li>• Whitelist input validation.</li> </ul>		 
6.	Is the application vulnerable to XPath Injection?	No		 
7.	Is the application vulnerable to XML Injection?	No		 

#	Description	Expectations	Results	Pass  Fail 												
8.	Is the application vulnerable to OS Command Injection? Examples: <ul style="list-style-type: none"><li>• ; ls /etc</li><li>•   cat /etc/passwd</li></ul> ; ping Y.o.u.r.IP	Not vulnerable		 												
9.	Is the application vulnerable to LDAP injection? Examples: <ul style="list-style-type: none"><li>• search?user=john</li><li>• default.aspx?user=*</li><li>• value)( (homedirectory=*)</li></ul>	No. The application escapes the following characters:  Used in DN - Requires \ escape <table><tr><td>&amp;</td><td>,</td></tr><tr><td>!</td><td>+</td></tr><tr><td> </td><td>-</td></tr><tr><td>=</td><td>"</td></tr><tr><td>&lt;</td><td>'</td></tr><tr><td>&gt;</td><td>;</td></tr></table> Used in Filter- Requires {\ASCII} escape <ul style="list-style-type: none"><li>• ( {\28}</li><li>• ) {\29}</li><li>• \ {\5c}</li><li>• * {\2a}</li><li>• / {\2f}</li><li>• NUL {\0}</li></ul>	&	,	!	+		-	=	"	<	'	>	;		 
&	,															
!	+															
	-															
=	"															
<	'															
>	;															



## 5.7 Web App & Server Security Misconfiguration

<b>Tools</b>	<ul style="list-style-type: none"> <li>• Netcat</li> <li>• Browser</li> <li>• Burp Suite</li> <li>• Nmap</li> </ul>
--------------	---

#	Description	Expectations	Results	Pass  Fail 
1.	Which HTTP Request Methods are supported by the target infrastructure?	The following methods shouldn't be accessible: <ul style="list-style-type: none"> <li>• TRACE</li> <li>• CONNECT</li> <li>• PUT</li> <li>• DELETE</li> </ul>		 
2.	Does the target allow Directory listing/browsing?	No		 
3.	Does the target has robots.txt file? If yes, does it contain sensitive directories?	Robots.txt if implemented, shouldn't contain sensitive directories.		 
4.	Is the file crossdomain.xml available?	No		 
5.	Is the Database listener accepting connections?	No		 
6.	Does the server running the Web Application also has FTP listener running?	No		 

## 6 OWASP Desktop Top 10 – 2021

Source: <https://owasp.org/www-project-desktop-app-security-top-10/>

Name	Description
<b>DA1 - Injections</b>	SQLi, LDAP, XML, OS Command, etc
<b>DA2 - Broken Authentication &amp; Session Management</b>	OS / DesktopApp account Authentication & Session Management, Auth. for Import / Export with external Drive, Auth. for Network Shared Drives or other Peripheral devices.
<b>DA3 - Sensitive Data Exposure</b>	Data in Memory post App Logout, Logs with Sensitive Info., Hardcoded Secrets in files, etc.
<b>DA4 - Improper Cryptography Usage</b>	Weak File/Folder Permission per User Role, Missing Principle of Least Privilege approach, Improper User Roles
<b>DA5 - Improper Authorization</b>	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasn't done correctly.
<b>DA6 - Security Misconfiguration</b>	Weak OS Hardening, Misconfigured Group Policies / Registry / Firewall rules etc., Missing File Type check for File Processing Apps, Misconfigured Named-Pipes, misconfigured 3rd party services, etc.
<b>DA7 - Insecure Communication</b>	Usage of weak TLS or DTLS Cipher-suites or Protocols, Unencrypted DB Queries in Transit, Absent Encrypted standard/custom protocol communication like HTTP, MQTT, COAP, etc.
<b>DA8 - Poor Code Quality</b>	Missing Code-Signing and Verification for File Integrity, Missing Code Obfuscation, DLL-Preloading or Injection, Race Conditions, lack of binary protection (Overflows, Null pointers, memory corruption) etc.
<b>DA9 - Using Components with Known Vulnerabilities</b>	Usage of Outdated Softwares, or Usage of Obsolete Components/Services of Windows/3rd Party vendors
<b>DA10 - Insufficient Logging &amp; Monitoring</b>	Missing or Improper Logging of Activities, Missing Regular Monitoring to Detect Abuse

## 7 Glossary

Abbreviation	Description
<b>2FA</b>	Two Factor Authentication
<b>API</b>	Application Programming Interface
<b>BC/DR</b>	Business Continuity and Disaster Recovery
<b>C2</b>	Command and Control
<b>CSRF</b>	Cross-Site Request Forgery
<b>DLP</b>	Data Loss Prevention
<b>ePHI</b>	Electronic Protected Health Information
<b>HIPAA</b>	Health Insurance Portability and Accountability Act
<b>IDS</b>	Intrusion Detection System
<b>IPS</b>	Intrusion Prevention System
<b>JS</b>	JavaScript
<b>OSINT</b>	Open Source Threat Intelligence
<b>OWASP</b>	Open Web Application Security Project
<b>SDLC</b>	Software Development Life Cycle
<b>SIEM</b>	Security Information and Event Management
<b>SOC</b>	Security Operation Center
<b>SOP</b>	Standard Operating Procedure
<b>SQLi</b>	SQL injection
<b>VA</b>	Vulnerability Assessment
<b>WAF</b>	Web Application Firewall
<b>WAPT</b>	Web Application Penetration Test
<b>XSS</b>	Cross-Site Scripting
<b>XST</b>	Cross-Site Tracing
<b>XXE</b>	XML External Entities