1.5. Testing Timeline

The following table outlines key milestones during the penetration test: **Penetration Timeline**

Date	Milestone
February 11, 2020	Start of Project
February 18, 2020	Final Deliverable

1.6. Target Description

The penetration testing for **IBM Notes** Window Application was carried out with one package. The approach conducted was a black box testing followed by black box testing. The tester was communicated with window application internally with web server **Production environment**.

Application	Type	Version	MD5 Checksum
IBM Notes	Window Application	N/A	200613546a98dde29fb135cf7dfe0886

2. Keywords

The following format shows a typical vulnerability representation and provides in detail information of vulnerabilities discovered during Application Vulnerability Test. The title bar for each vulnerability table is color coded for a quick identification of the risk level. Title bar color codes are as follows:

Risk Level	Description
	High risk vulnerability can be exploited by an attacker to gain full administrative access to the application or its underlying operating system.
	Medium risk vulnerability reveals information about the application and its underlying infrastructure that can be used by an attacker in conjunction with another vulnerability to gain administrative control of the application or its underlying operating system.
	Low risk vulnerability can result in enumeration of vital information held by or about the Application or its underlying operating system.

- **OWASP Category –** Refers to OWASP top 10-2017 vulnerability category.
- **Abstract –** Describes the flaw or bugs that cause the vulnerability.
- **Ease of Exploitation –** Provides a metric for the skill level required to exploit the vulnerability. The categories are:

Metric	Skill-level
Easy	Casual user
Medium	Computer-savvy individual

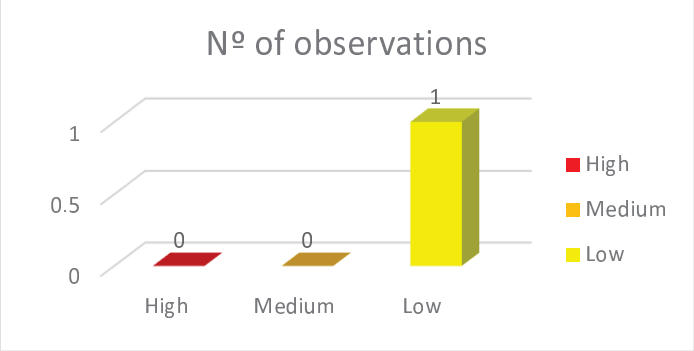
Hard	Determined hacker

- **Impact –** Describes the possible business impact if this vulnerability is successfully exploited.
- **Recommendation –** Provides solutions or workarounds to mitigate the risk arising from this vulnerability.
- **Substantiated Assessment –** The evidence of the vulnerability being present, wherever possible, is provided in the form of screenshots.
- **Affected URL –** Provides URLs and respective parameters which are affected with that specific vulnerability
- **Note –** A short description of how the vulnerability can be exploited by internal/external attacker.
- **Reference –** It provides reference to outside resource such as OWASP, SANS etc.
- **CWE –** Provides Common Weakness Enumeration ID

3. Observations Summary

The Window application is containing vulnerabilities that an attacker can target or exploit. It is important to periodically check, review and modify logic if any kind of change is being applied to the production. The graph below gives the status of severity of the vulnerabilities found during the Window Application Security Assessment.

Risk Severity Level	No of Observations
High	0
Medium	0
Low	1
Total	1



Given below is the summary of the observation

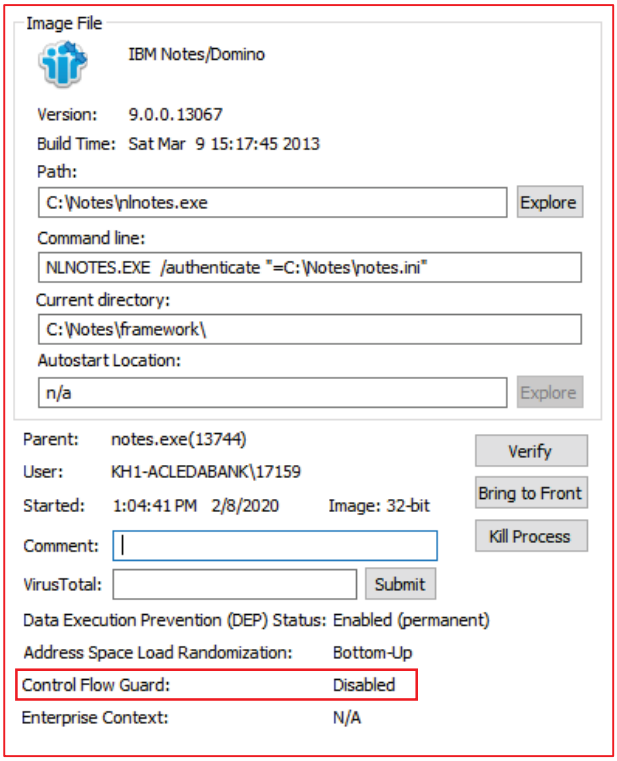
No.	Observations	Risk Level
1	Control Flow Guard is disabled	Low

3.1.Statement of compliant

The tester has determined that **IBM Notes** window application is **NOT-COMPLIANT** with validation requirement as mentioned in section 1.3.

4. Vulnerability Discovery Detail

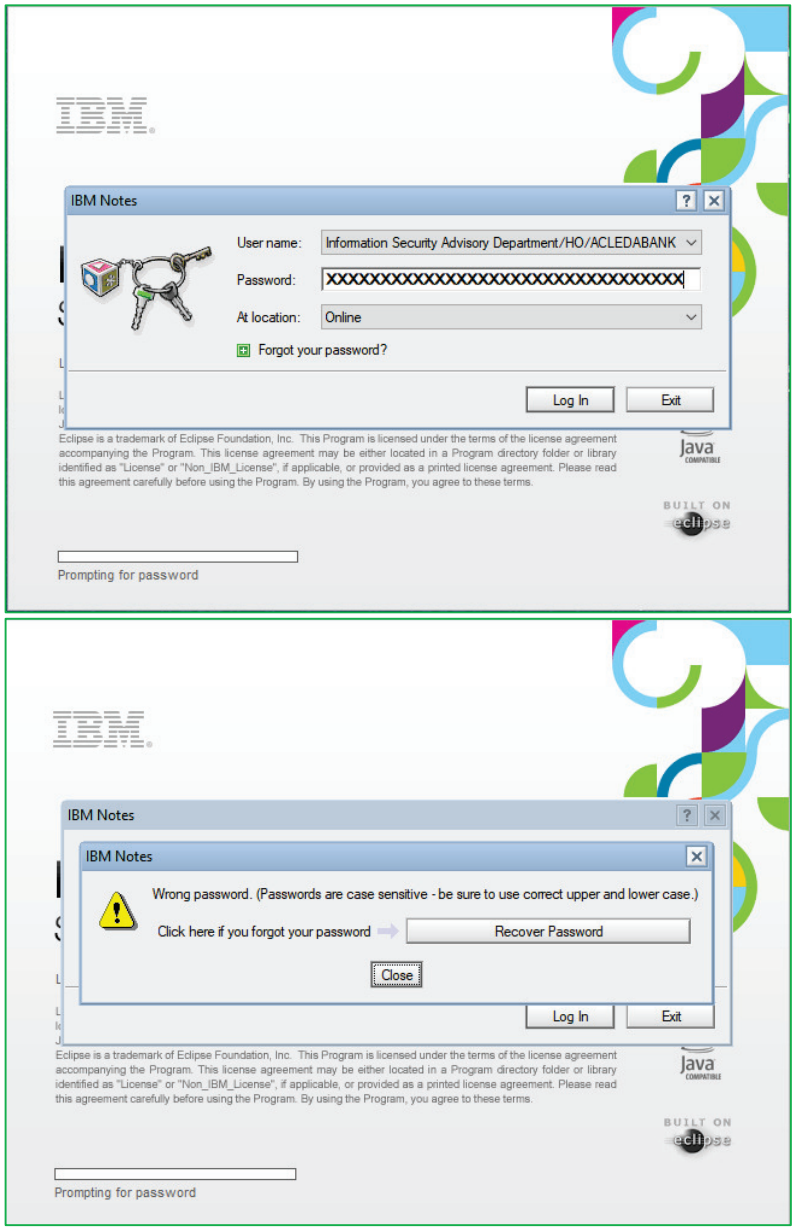
4.1. Control Flow Guard is disabled

Risk Level	**Low**
OWASP Category	**A6-Security Misconfiguration**
Abstract	Control Flow Guard (CFG) is a highly-optimized platform security feature that was created to combat memory corruption vulnerabilities.
Ease of Exploitation	**Hard**
Impact	By placing tight restrictions on where an application can execute code from, it makes it much harder for exploits to execute arbitrary code through vulnerabilities such as buffer overflows.
Recommendations	It's strongly recommended to enable CFG for application.
Substantiated Assessment	- Tester used a tool to identify security features. Control Flow Guard is not enabled. 
Affected URL	N/A
Reference	https://docs.microsoft.com/en-us/windows/win32/secbp/control-flow-guard https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration https://cwe.mitre.org/data/definitions/119.html
CWE	**CWE-119**

5. Appendix

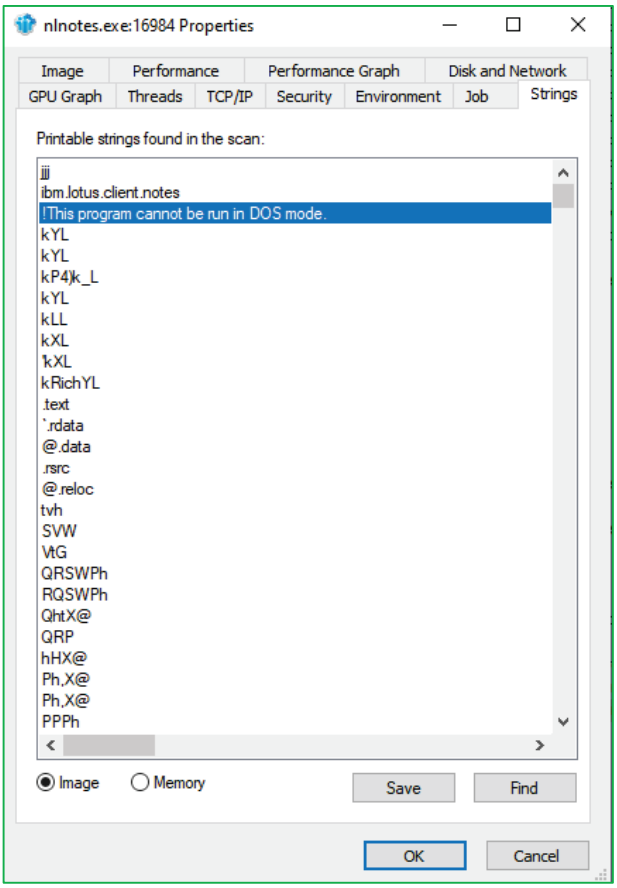
5.1. SQL Injection

- On Login window form, tester submitted request with string (special character) to comment the required password. The application showed wrong password message.



5.2. Hard-coded credentials

- Tester used a tool to scan the string in application. There is no interesting in the application.



6. Acknowledgement

18/February/2020	18/February/2020	18/February/2020
Approved by	Certified by	Prepared by
Chhay Yaroth	**Kry Dalin**	**Por Chetra**
SVP & Head of Information Security Division	VP & Head of Information Security Advisory	Staff (Information Security Advisory)