

Project Part 2 Documentation
Foundations of Artificial Intelligence

Divesh Badod(db9932@rit.edu)

Prof. Leonid Reznik

1. Executive Summary

The problem addressed here is the question of the security of computers and mobile devices. Every device is vulnerable to viruses and threats from hackers or just from browsing the wrong section of the internet. Hence, I have built a system which takes in some of the inputs from the user and evaluates the security of the device the user owns. The system built here will show how safe the device is or how vulnerable it is in the form of rating ranging from 1 to 10 where 10 is the highest and 1 the lowest. To keep the system compatible with Android, Windows, Linux and even macOS the metrics involved in evaluation expand into the region's most commonly extant in all the systems. The metrics range from System architecture to Data security to Network security, where the user can be asked about the latest software installed or security patches installed. All these metrics and the rules have been developed through extensive research in the topics of Android and Computer Security. All these metrics have been fleshed-out by looking into some concepts on security concerns of the said devices, like downloading applications from a 3rd party websites or even using some unknown and unsecured browsers will have a greater impact device's security. Hence every metric has its weights on how much they help in securing the device or how much they keep the device vulnerable, enough for an attack. Based on the inputs from the user there will be calculations done and the output of the security rating will be presented.

This system is built using the JESS expert system and JAVA programming language. These two were chosen because JESS doesn't have too much of a learning curve and it was written entirely using JAVA hence making it much easier to use. JESS can be extended using JAVA or embedded in JAVA and for this project, I have used JESS by embedding it in JAVA. JESS is an expert system in which the knowledge is provided in the form of IF-THEN rules, hence in our project, the knowledge provided to JESS is based on the facts about metrics which the user will input. Whatever input the user gives JESS will use those inputs as facts and use its knowledge base provided by me to determine that metrics' impact on the security rating. All these metrics' input will then be calculated again using JESS engine to present the final rating. In this project, JAVA has been used mostly for embedding JESS and for GUI purposes whose guide has been exhibited in the subsequent sections.

2. Requirements

The basic requirements will be most of the latest operating systems used today like Windows 10, Linux, Android and macOS. A processor with enough GPU power to load the GUI without any delay and latest JAVA installed. Installing JESS won't be necessary since it is embedded in JAVA the engine will be running through JAVA. The concise description is given below.

- Hardware Requirements:-
 - Processor:- Intel(Core, Core 2, i3, i5, i7, i9), AMD(K10 Series, Zen Series, Z+ Series, Zen 2 Series)
 - GPU:- The CPUs mentioned above are enough for the GUI to load but Nvidia and AMD has a varied range of graphics processors
 - RAM:- Above 2gb
 - Storage:- Above 1gb

- **Software Requirements**
 - OS:- Windows(XP, Vista, 7, 8, 8.1, 10), Linux, macOS(10.8, 10.9, 10.10, 10.11, 10.12, 10.13, 10.14, 10.15), Android(4.4, 5.0, 5.1, 6.0, 7.0, 7.1, 8.0, 8.1, 9, 10, 11)
 - JAVA:- JAVA SE(6, 7, 8, 9, 10, 11, 12, 13, 14)
 - JESS Engine(Optional):- JESS(5.2, 6.1, 7.1)

The requirements are stretching from the software used since 2007, but this system has not been tested in the older software so there might be compatibility issues. And for some reason, you have a software older than the ones above there is no guarantee that the system will run at all. Even though the requirements show the older versions I would recommend you using the latest available software and hardware for this system to run at its best.

3. Specifications

- **Using an Expert System and Learning about its Applications**

Expert systems have been in usage since the 1970s and they might have been the first truest form of Artificial intelligence during the topic's inception. These systems have helped greatly in science and medical fields where human decisions might have been inept. Hence to gain a rudimentary understanding of expert systems in AI and other computational fields I am developing this project using such system called JESS. Here I am implementing JESS by providing rules in the form of IF-THEN structure and trying to make it closer to emulating decision-making process as human-like as possible. The expert systems require knowledge, which is provided by the expert, this knowledge is represented in the IF-THEN rules which are done by the knowledge engineer and the programmer. And on the other end of the expert system is the user who may or may not be an expert who will be using this system for purposes like security check of the device.

- **Understanding the development process of AI applications like the real-world convention**

AI has been the pioneer field of research since the 1950s excluding the AI winter of the 1970s and the progress made in these years has been remarkable. For this project even though I have been using the most straightforward concepts of AI the final application built will be a combination of implementation of AI concepts while gaining a deeper understanding of the said concepts as though I am working as a project manager of an AI development team. Gathering knowledge of the computer and mobile devices about their security and vulnerabilities, using such knowledge to analyse what metrics to use, after choosing the metrics again using said knowledge developing rules for the expert system, learning about the expert system, applying the developed rules in the expert system, making it as human-like decision-making machine as possible and finally developing a GUI whilst learning how to embed the said expert system with the GUI, are just some of the vital checklist of processes involved in building this project which in turn simultaneously is imparting me with the industrial applications. Finally delivering an application to the users with a guarantee that the system built will accurately show you how good or bad the security rating is in your device.

- **Knowledge analyses and developing rules**

Expert systems require an expert to fill out the knowledge in the knowledge base. This is done by the experts who have a great deal of practical and theoretical knowledge on the subject, of which the expert system will be built. For my project, I had to go through various research papers regarding security on android devices and computers alike. In this process, I gathered knowledge about what makes these devices most vulnerable to attacks from attackers and/or viruses. There was a multitude of results which I had acquired, this led me to drop various interesting but unfitting outcomes to finally scope out the metrics which I have used for this project.

After scoping out the metrics the rules were needed to be developed for the expert system again the knowledge at hand was useful because while searching for the metrics I researched about how much these metrics affected a device's security. This process of developing rules is done by the knowledge engineer and implementing these rules in the expert system is done by the programmer. Some metrics had too much influence over the system's security like the type of browsers being used or if the antivirus was installed or not, and some had too little control like the system's hardware yet they were vulnerable enough for threats of hardware corruption depending on the architecture. Similar rules were applied to all the proposed metrics and then the rules were developed for the expert system. The metrics used were not too specifically pertaining to a single OS or device, these metrics could be used in multiple devices or OS thereby replenishing the robustness lost while embedding the JESS expert system with GUI.

4. Description of Domain Problem

- **Rule Engine (JESS)**

JESS stands for Java Expert System Shell, written entirely in Java programming language, essentially a rule engine in which expert systems are built using knowledge as rules. Originally inspired by CLIPS expert system shell JESS has grown into a Java-influenced environment of its own. It is used to make Java servlet, Enterprise JavaBeans, and Applets. By using the knowledge which is supplied as declarative rules in JESS the applications have the capacity of 'reason' resembling human-like decisions. Since it was inspired by CLIPS the programming syntax in JESS is Lisp-Like. Knowledge in JESS is represented as Rules, Functions, Object-oriented Programming (Classes, Encapsulation, Abstraction, Polymorphism, Inheritance).

- **Rete Algorithm**

This is an undisclosed algorithm used for embedding JESS with Java. By using this algorithm you can also say that the expert system we are using is a Rete Based Expert System. Rete algorithm provides a basis for more efficient implementation to rather computationally expensive rules in the rule engine. This algorithm works with a network of nodes where each node corresponds to the LHS of the rules in the rule engine. Each node serves as a memory filled with facts which will satisfy the pattern on the rules (LHS). When a combination of facts is triggered the network will traverse through its node and when the leaf node is reached then the corresponding rule will be triggered. This helps in determining and tracking which rule

should be fired and when it should be fired based on the given facts. Since this is more focused on speed it is memory intensive.

- **GUI**

Java's GUI widget toolkit called Swing was used make the GUI of this application. Swing is described as a more sophisticated and easy-going library compared to the Abstract Window Toolkit(AWT) an older GUI toolkit of Java. Since Swing is not implemented as platform-specific code the compatibility issues of the GUI is no more risk for this application.

5. Feasibility

- **Expert System alternatives**

Even though here we are using JESS rule engine and Rete Algorithm embedded with Java to create a fully functioning expert system to cater the easiness and uncomplicated information gathering and processing from the user. The next closest software I could've used was CLIPS or Prolog as well.

	JESS	CLIPS	Prolog
Features	Great set of features like Lisp-like language syntax, Java API, and interactive graphical shell. Provides GUI creation like Java. Pattern matching feature for running collection of rules on a set of facts at once making it more efficient.	High set of features written in C language hence it can be called from C. Most widely used tool for expert systems. Combines programming paradigm of procedural, OOP and Logical languages. Also provides GUI creations	Limited Features mostly catered to declarative programming method. Newer versions have started supporting GUI creations
Portability	Highest portability. Can run on any engine which has Java 1.4 or above	Through Windows its easier but for other systems, you would require C source code to be compiled.	Complicated programming since not all Prolog compilers support modules and compatibility issues within the modules of Prolog compilers
Resource Consumption	High	High	Very small

- **GUI alternatives**

Using Java's swing library has made the project more robust to platform changes. There are many alternatives to swing like using AWT an older GUI library provided by Java or for more advanced look JavaFX could've been used or even using JESS to create the same GUI.

	Swing	JavaFX	AWT
MVC Support	Does support MVC pattern but the components lack consistency	Highly supports MVC and it is pattern friendly	Does not support MVC
GUI Components	Lightweight GUI components but has a god number of components	Lesser number of components but it has the rich feel to every component to create an advanced GUI	Heavyweight GUI components due to dependencies on native code functioning.
Portability	High portability but low performance	Lower portability but better performance	No portability at all since the code resides in the OS of the system to run this in a different device the entire native code should be compiled first but has the best performance.

- **Metrics alternatives**

Metrics are the essence of this project and through thorough research, I concluded with using the 10 proposed metrics. Digging deeper into the OS of devices, hardware compatibility, network usage, you can get to various other options for metrics like USB port debugging in Android, developer options tool, background process settings, application formats like APK, JAR, EXE, but these metrics hinders the robustness of this application because all of these options only work on one specific OS, e.g. the USB debugging tool, hardware like SIM cards or even the access to bootloader of the device works only on Android devices, changes in Bios Settings will only have an effect on devices with Windows. To keep the application more system specific you can use these metrics and get a more accurate and detailed security rating than it produces now embodying that OS.

The alternatives are great solutions to make the same application with different approaches and distinctive design and distinct set of rules too. But my intent for this project was to keep the overview security questions in check for all the devices it will run on. To keep it simple and robust yet cover all the factions of security concerns was my motto and by proceeding with my approach I have tried to make this application further access to all the users and keep it friendly enough for everyone's understanding of the security of their devices.

6. Implementation

- **Knowledge Base**
 - **Forward Chaining**

With the knowledge acquired for this application's metrics, the expert system is implementing the process of Forward-Chaining also called the Data-Driven approach. This approach takes in data and uses that data as facts to trigger the rules in the expert system and to reach a particular goal. Here the data taken is from the user inputs and some inputs the application detects on its own, these inputs will work as facts for the rules set in the rule engine JESS. Then a Rete Algorithm used to embed JESS in Java is prompted, triggering the rules, with the given facts the impact of these metrics will be decided ranging from 0% impact on security to 100% impact on security(Varies for different metrics) after that these ratings will be generalized into 3 categories explained in later subsection then the calculation rule is triggered to give out the final rating of the security. Forward chaining is an efficient technique to perform the security rating of the system because it is data-driven. The security rating of the device is the final goal of the forward chaining process implemented here.

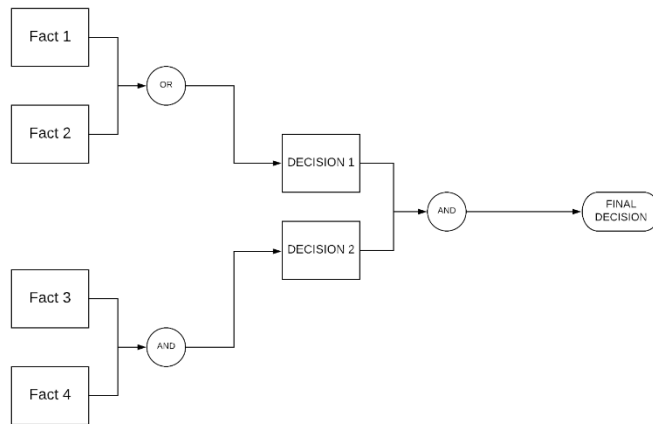


Fig 1. Forward Chaining Process

- **Structure and Contents(Metrics and Hierarchy)**

The following metrics were considered for this application:-

Metrics	Contribution	Weights
The current software in use	Checking if the latest update of the software is installed	0.3
Password Protected	Checking if the device is password or fingerprint protected	0.5
Firewall	Checking if the latest firewall is installed	0.4
Number of users	Checking how many users have access to this machine	0.1
Security Patches	Checking if latest security patch is installed	0.3
3 rd Party Apps	Checking how many 3 rd party applications have been installed	0.4
System Architecture	Checking which system architecture is the device running on	0.2
Antivirus	Checking if the latest antivirus software has been installed	0.3
Browsers	Checking which browsers are safe to use	0.2
Peer to Peer Network	Checking if you use Peer to Peer networks	0.3

The metrics presented above are then generalized in 3 categories culminating into an overall device rating system by considering the 3 important aspects of a particular device:-

SECURITY RATING(0,10)									
SYSTEM(3.33)			DATA(3.34)				NETWORK(3.33)		
SOFTWARE(0.3)	ARCHITECTURE(0.2)	PASSWORD PROTECTION(0.5)	ANTIVIRUS(0.3)	BROWSER(0.2)	3RD PARTY APPLICATIONS(0.4)	NUMBER OF USERS(0.1)	FIREWALL(0.4)	SECURITY PATCHES(0.3)	PEER TO PEER NETWORK(0.3)

Fig 2. Metrics Hierarchy with given weights

The given hierarchy is used as a forward-chaining technique for our application, first, the user enters the base inputs i.e. SOFTWARE, PASSWORD PROTECTION, BROWSER etc. With the given inputs the security impact percentage is set via rules set in the rule engine. Then after setting the impact rating they are multiplied with their given weights which were set after meticulous researching. The rule for calculating the value of individual categories SYSTEM, DATA, NETWORK is then triggered where the output value after calculating the lower hierarchy of rule is taken as the input to the higher hierarchy of the rule, again multiplied by the given weights the final rating is calculated by adding all the three categories' value.

- **Expert System Applications(Unanimity)**

The elemental section of this application is the expert system implementation for security evaluation, to get the expert system working in a rational manner the knowledge base created was the accumulation of multiple security concerns of devices across multiple platforms. Thereby keeping this application substantial enough to work across multiple platforms. Consider a user decides to use this application on his or her macOS device and Linux device at the same time when the user inputs the data into the application the expert system will evaluate the security rating of the respective devices and present the user with the security rating, suppose macOS has a lower rating than Linux then the user can determine immediately as to why is this the case since the questions for both the devices were same and the expert system built around those questions will evaluate the data for macOS in the same way it will

do for Linux, and when the user put the inputs according to the respective devices' orientation, the system might have calculated the macOS device with a lower rating than the Linux because it might have a higher number of 3rd party apps than the Linux ones, or maybe the latest security patch isn't installed, or maybe the browser the user uses isn't a safe one. This whole conduct again leads to having a selection of metrics universal across platforms for the expert system to work consistently, consequently making it easier for the user to keep his or her devices as secure as possible.

- **Limitations**

Since an Expert System is developed here, most of the limitations of this application tend to be inherited from the limitations of the expert system. Even though I established unanimity of this application, Expert Systems do not have the flexibility to adapt to a more creative scenario. It's difficult to maintain because for every change in facts there will be a tonne of coding and rule-setting to be done in the expert system. Knowledge acquisition tends to take its time because the rules can only be set up after scrupulous research and wrong rule setup can break the system's integrity. Hence all these factors combine demonstrations that maintenance and development of a simple application like this can still be rigorous because of the implementation of an expert system.

- **Applied Classes**

- **Swing**

A GUI widget toolkit for Java, Swing is an easy to use package because of its 'look and feel' aspects. An extension of AWT, Swing has major updates which makes it more powerful and flexible than AWT. This package is accessed by using `javax.swing.*` module in Java programming language. For the event handling of the buttons of AWT's event package was accessed from which `ActionListener` and `ActionEvent` classes were used by calling `java.awt.event.ActionListener` and `java.awt.event.ActionEvent` respectively.

- **Rete Algorithm**

Rete algorithm is a pattern binding algorithm and most of its gist and workings I have described in the sections above. This package was imported using the JESS jar file which was download from the official site. As explained above this pattern binding algorithm helps us in controlling which rules to trigger and reach a definite goal. The rule controlling package helped us attaining the required process of forward-chaining. Along with this a memory management class was used to mark where the engine had stopped before and run the engine again from that point to keep the flow of the rules intact.

7. Testing Description

- **Rating with valid inputs**

This application is tested on a Windows and a Linux device below are some of the examples: -

The screenshot shows a web application titled "Security Evaluation". The form contains the following fields and options:

- System you are currently using:** Windows 10
- Is the machine password protected?:** ☒ Yes ☐ No
- Is the latest Firewall installed?:** ☒ Yes ☐ No
- How many users have access to this machine?:** 1 (dropdown menu)
- Is the latest security patch installed?:** ☒ Yes ☐ No
- How many 3rd party(unknown source) applications are installed?:** 0 (text input)
- System Architecture in this machine:** amd64
- Which antivirus is installed?(If Linux or Mac user then select 3rd Party):** ☒ Windows ☐ 3rd Party ☐ None
- Which browser do you normally use?:** ☒ Chrome ☐ Mozilla ☐ Edge ☐ Safari ☐ Other
- Do you use peer to peer network?:** ☐ Yes ☒ No

A "Submit" button is located at the bottom of the form. A pop-up window titled "Security Evaluation Score HIGH" is displayed, showing an information icon, the score "9.9001", and an "OK" button.

Fig 3. Screenshot of Application

The screenshot above shows the highest score achievable on a Windows device with the ideal inputs. A device is with Windows OS incorporated within will always be vulnerable even if the best precautions are taken hence the highest achievable score is 9.9 and not 10. This applies to every system in use in today's world of internet of things because every device is dependent on other systems, Wi-Fi or Bluetooth, System updates require a network connection and yet foul connections can harm your device this kind of entanglement amongst the systems will always prevail and a device will never be 100% secure, be it Windows, Linux, Android or even macOS.

Since a device can never be 100% secure a device is never 100% vulnerable, the OS and hardware installed in the systems were built to tackle the most basic of threats like illegal applications or hardware corruption. Hardware nowadays is made with a trigger that if the system overheats shut down the system for a cooldown period and restart it similarly OS nowadays are built with a recovery system. Hence even though there might be a heap of threats

against a device's security nowadays systems are built to be a little less susceptible to very little and minor hindrances. The following figure shows the system providing the lowest rating possible on a Windows device after entering the inputs considered for the worst-case scenario on a Windows device.

The screenshot shows a web application titled "Security Evaluation". The main form contains the following fields and options:

- System you are currently using:** Windows 10
- Is the machine password protected?:** ☐ Yes ☒ No
- Is the latest Firewall installed?:** ☐ Yes ☒ No
- How many users have access to this machine?:** More (dropdown menu)
- Is the latest security patch installed?:** ☐ Yes ☒ No
- How many 3rd party(unknown source) applications are installed?:** 10000 (text input)
- System Architecture in this machine:** amd64
- Which antivirus is installed?(If Linux or Mac user then select 3rd Party):** ☐ Windows ☐ 3rd Party ☒ None
- Which browser do you normally use?:** ☐ Chrome ☐ Mozilla ☐ Edge ☐ Safari ☒ Other
- Do you use peer to peer network?:** ☒ Yes ☐ No

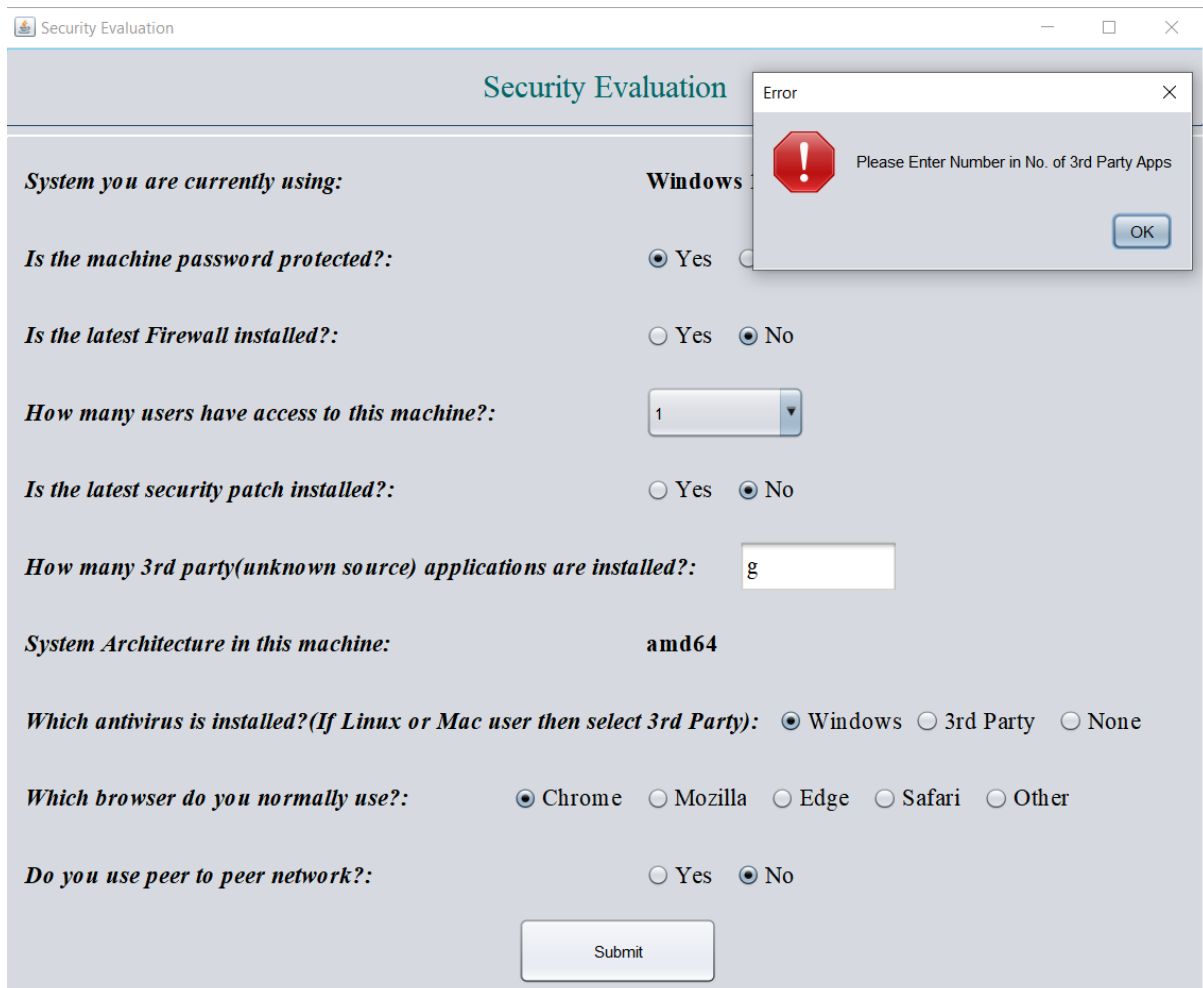
A "Submit" button is located at the bottom of the form. A modal dialog box titled "Security Evaluation Score LOW" is open, displaying an information icon, the score "3.2485", and an "OK" button.

Fig 4. Screenshot with the lowest possible score on Windows

I tested this application of a Linux device using SSH protocol with the same inputs and the outputs were more or less the same with the only changes being the System architecture and the OS because those two metrics are perfunctorily taken in by the application.

- **Rating with Invalid inputs**

In the question of ‘Number of 3rd party applications installed in your device?’ the system will generate an error shown in the following image if you enter anything other than a number. This number cannot be fractions or decimals or equations it should only consist of integers.



The screenshot shows a web application titled "Security Evaluation". It contains several input fields and radio buttons for a security assessment. The fields are:

- System you are currently using:** Windows
- Is the machine password protected?:** Yes (selected)
- Is the latest Firewall installed?:** No (selected)
- How many users have access to this machine?:** 1
- Is the latest security patch installed?:** No (selected)
- How many 3rd party(unknown source) applications are installed?:** g
- System Architecture in this machine:** amd64
- Which antivirus is installed?(If Linux or Mac user then select 3rd Party):** Windows (selected)
- Which browser do you normally use?:** Chrome (selected)
- Do you use peer to peer network?:** No (selected)

An error dialog box is displayed over the "How many 3rd party(unknown source) applications are installed?" field. The dialog box has a red exclamation mark icon and the text: "Error Please Enter Number in No. of 3rd Party Apps". There is an "OK" button in the dialog box.

Fig 5. Error in one of the inputs

After you click the ‘OK’ button the application will continue to calculate the security rating, but it will set the impact of this metric to 0%. The same situation is applied to every other metrics when there are no inputs for them. The following figure shows the rating of a device without any inputs given. The system will still not show 0 because there are 2 metrics system and system architecture which are still not being considered as 0 because they were taken in by the applications itself hence those metrics will always be available in every device. Other than that, the metrics dependent on users will be set to 0 if there are any errors in the input for those metrics.

The screenshot shows a web application titled "Security Evaluation". The form contains the following questions and inputs:

- System you are currently using:** Windows 10
- Is the machine password protected?:** ☐ Yes ☐ No
- Is the latest Firewall installed?:** ☐ Yes ☐ No
- How many users have access to this machine?:** 1 (selected from a dropdown menu)
- Is the latest security patch installed?:** ☐ Yes ☐ No
- How many 3rd party(unknown source) applications are installed?:** (empty text input)
- System Architecture in this machine:** amd64
- Which antivirus is installed?(If Linux or Mac user then select 3rd Party):** ☐ Windows ☐ 3rd Party ☐ None
- Which browser do you normally use?:** ☐ Chrome ☐ Mozilla ☐ Edge ☐ Safari ☐ Other
- Do you use peer to peer network?:** ☐ Yes ☐ No

A "Submit" button is located at the bottom of the form. A pop-up message box titled "Security Evaluation Score LOW" is displayed, showing an information icon, the score "2.9011", and an "OK" button.

Fig 6. Output for no inputs

Even though the 'Number of user' question shows 1 by default the user has to click on the menu and choose the number 1 for the application to register the input as 1 otherwise it will be set as invalid and the impact of this metric will be set to 0.

8. User's guide and GUI

The GUI was created using Java swing with the help of NetBeans IDE in which GUI development is as simple as drag and drop. The source code of the GUI is automatically constructed by the IDE while you set it up on the design section. This makes the development of GUI a lot easier. The application is fairly easy to use because most of it is just selecting and entering the correct inputs from the given options. Most of them are radio buttons too, making it much simpler to understand how to select the options.

Security Evaluation

System you are currently using: Windows 10

Is the machine password protected?: ☒ Yes ☐ No

Is the latest Firewall installed?: ☐ Yes ☐ No

How many users have access to this machine?: 1

Is the latest security patch installed?: ☐ Yes ☐ No

How many 3rd party(unknown source) applications are installed?:

System Architecture in this machine: amd64

Which antivirus is installed?(If Linux or Mac user then select 3rd Party): ☐ Windows ☐ 3rd Party ☐ None

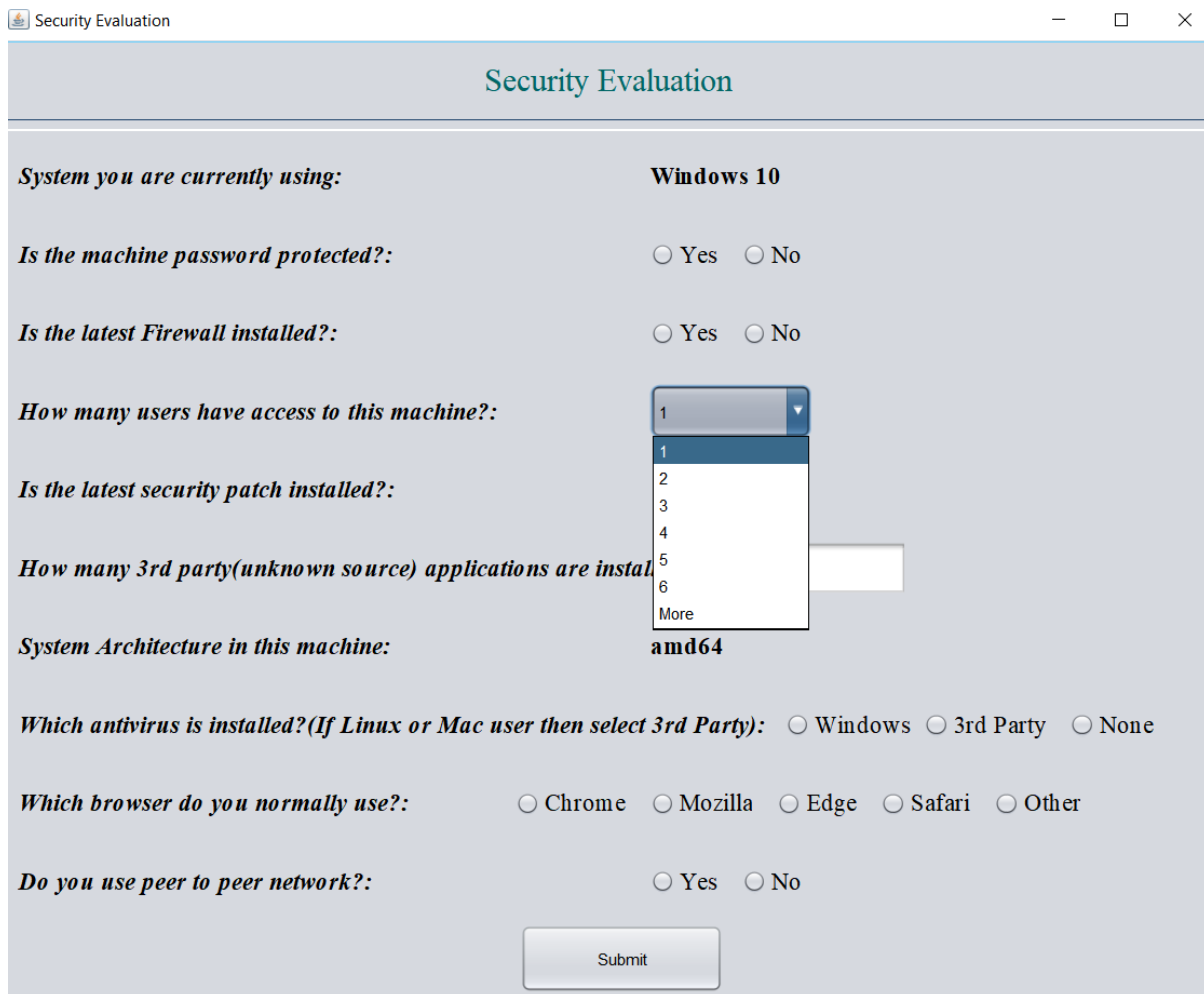
Which browser do you normally use?: ☐ Chrome ☐ Mozilla ☐ Edge ☐ Safari ☐ Other

Do you use peer to peer network?: ☐ Yes ☐ No

Submit

Fig 7. GUI of the Application

Every metric can be provided with just a single click from the given options. 3rd party application question will require you to enter nothing other than an integer. The 'number of users' question has a drop-down menu for answering the question.



The screenshot shows a web application titled "Security Evaluation" with a light blue header. The form contains several questions with corresponding input fields or radio buttons. A drop-down menu is open for the question "How many users have access to this machine?". The menu lists options 1 through 6, and a "More" option at the bottom. The "Submit" button is located at the bottom center of the form.

Security Evaluation

System you are currently using: **Windows 10**

Is the machine password protected?: ☐ Yes ☐ No

Is the latest Firewall installed?: ☐ Yes ☐ No

How many users have access to this machine?:

Is the latest security patch installed?:

How many 3rd party(unknown source) applications are installed?:

System Architecture in this machine: **amd64**

Which antivirus is installed?(If Linux or Mac user then select 3rd Party): ☐ Windows ☐ 3rd Party ☐ None

Which browser do you normally use?: ☐ Chrome ☐ Mozilla ☐ Edge ☐ Safari ☐ Other

Do you use peer to peer network?: ☐ Yes ☐ No

Fig 8. GUI of Application with the drop-down menu open