**PROJECT 2 DOCUMENTATION**

**WINDOWS SECURITY CHECK - EXPERT SYSTEM**

**Group – 7 : Sai Tarun Sathyan (SS4005)**

**1. Executive Summary**

Computers are a big part of our day-to-day life. We use computers for every task and thus store a lot of personal and sensitive data on our computers, all the way from photos to bank details. The security of our computer dictates the security of our personal information. Thus, to ensure the safety of our computers I present a tool to evaluate the safety of the computers we use. This tool is used for the security evaluation of the windows operating system in a personal computer or a laptop. The tool analyzes the systems features and parameters that may influence the security of the system, while also going through the installed applications and software that may affect the security of the system. This tool can be used by anyone and does not require expertise in the windows operating system. This tool can be used on stock devices by simply downloading it as an application.

The tool that has been developed in python. This tool provides performs a set of tests to check each security metric. After which the expert system provides analysis and advice on what can be done to make the device safer. Problems such as windows device not having a password lock, windows firewall has not been activated or the device not having an antivirus software must be addressed. This tool helps identify all such problems. After doing so it gives your device a security rating and analysis on what can be improved in your windows PC/Laptop.

**2. Requirements**

The tool that has been developed in python so the user must have python (latest version) installed on their device. Python IDEs like Pycharm, VScode or Sublime can also be used to make things easier. The code uses a couple of libraries that need to be installed.

The required libraries are:

* + win32com
  + windowsapps
  + winerror
  + win32security
  + windows\_tools
  + pywin32(optional depending on version)

Last but not least the user requires a windows device for this expert system to work. A windows 10 system would be preferable, since this system was designed with windows 10 in mind.

**3. Specification**

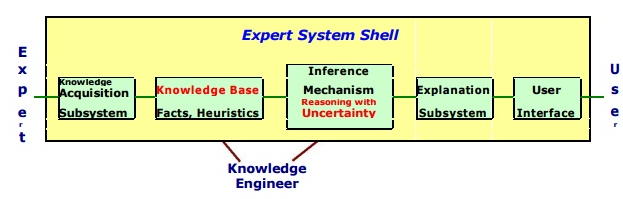
The security metrics that I’ve finalized for this expert systems analysis are:

* Device Password Lock
* List of Trusted Applications
* Risky Permissions
* Update Version
* Firewall Check
* Anti-Virus Check

These metrics were decided based on the different types of vulnerabilities a windows device could face. Even though most of it is software based, it is what dictate information security. Each metric has its own purpose to fulfil. To gain such information about the windows device the windows api for python was very useful. Pywin32 and windows\_tools being the most useful libraries by far.

**4. Description of The Domain Problem**

The tool is an expert system that works on a simple knowledge based that is used.



The knowledge acquisition was done by me through online research. The reasons for choosing the listed security metrics(rules) are as follows:

**The Device Password** – Is a metric that is used to help assure that strangers cannot simply pick up your device and access all of your personal data. The password for a device acts as the first layer of protection if stranger has physical access to a person’s personal computer or laptop.

**Trusted Applications** – Is a metric that ensures that the programs installed onto the PC are trustworthy, have reliable source and don’t have any malware running in the background. The apps or programs on a person’s device can act as an access point for someone trying to steal a person’s data.

**Risky Permission** – Giving basic apps and programs administrative privileges or access to your location/email might be a risky task and has to be handled properly. Identifying which programs should have what level of access is an important aspect of data security.

**Update Version** – Each version of windows has its own set of bug fixes and new software patches. Thus, making sure that your windows operating system is up to date is also a step towards added security. Since your previous windows version might have security flaws. Hence keeping your system firmware up-to date will also act as a security metric.

**Firewall Check** – This security metric is to ensure that the windows firewall is active. Making sure that a firewall is active ensures that we will not receive any spoof/virus through the network and also ensure unusual packet transfers will be detected.

**Anti-Virus Check** – This security metric makes sure that the device has windows security. Windows security is the default anti-virus software for windows 10 and windows 11. Make sure this is enabled is important because the windows security runs all the time and makes sure no malware gets downloaded onto the system.

The inference engine works based on simple if-then logic. Each metric has an if-then or if-and-then type statement that evaluates the metric. The user interface is a text-based output and there is no need for much input from the user. The user simply needs to run the system and let it analyse and produce an output. After the inference engine has finished evaluation, the user will obtain his security score and expert analysis in text format.

5. Feasibility Study

The project so far has been able to produce an expert system that can calculate simple metric which are binary. I did have further upgrades in mind which would take far more time and resources. The additional upgrades that I wish to complete are:

**Full system virus scan**: That goes through every file and scans it for viruses or malicious scripts. But the problem with this is that there are so many different types of viruses and different forms of malicious scripts that coding something that covers everything would take too much time. But the current expert system does perform an antivirus check that makes sure the windows device has an antivirus software installed thus making sure there is some form of protection against these malicious scripts.

**List of Trusted Programs**: The Idea behind this was to display a list of applications or programs that can be trusted, essentially letting the user know that there was no malicious software. But once again I face the same problem as before, since this takes too much time and resources. Coding this in Python was also a bit of a hassle due to how resource intensive it is to scan every .exe file in a system and make sure it is safe.

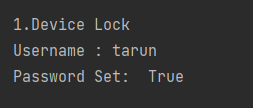
**Future Optimizations**: The current expert system can be optimized using machine learning. So that the user gets faster results and more precise outcome. But this requires further study and research into this subject.

**6. Implementaion**

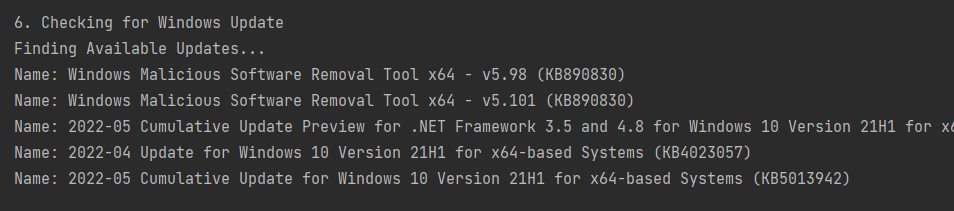
The knowledge base is a big but organized data set of knowledge about a particular subject. The knowledge base consists of facts and expert knowledge that a system can use to make educated guesses about the judgement that it needs to make. Since this project is related to expert systems that provide a security score. The knowledge base for the current expert system is a set of security metrics that represent the safety of the user’s device. Each security metric is a rule that helps the user safeguard his/her data.

The expert system applies various tests to provide expert feed-back. Examples are as follows:

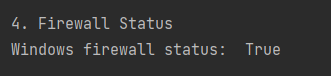
Device Password lock test output : Checking if the user has a password lock setup for the device, thus ensuring that strangers that have physical access to the device can not easily access it’s data.



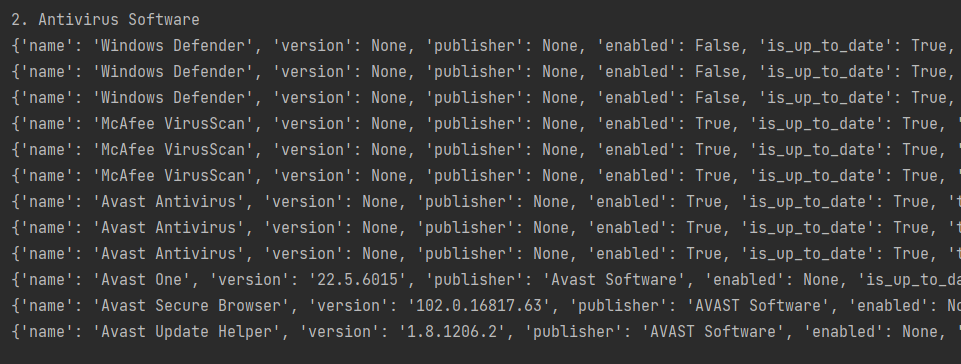
Windows device update output : Checking if the system software is upto date and hence making sure that no security patch was missed.



Fire wall check output: Making sure the network firewall is enabled.



Antivirus Software output : Let’s us know if the windows device has an active antivirus software enabled. This makes sure malicious scripts will be detected by such antivirus software.



References for libraries:

[1] Library: windows\_tools ; Author: NetInvent - Osiris De Jong  
[2] Library: windows\_apps ; Author: Tushar Goyal  
[3] Library: pywin32/win32com ; default

Requirements:

* Software – Python (latest version), IDE like Pycharm
* Hardware – Windows 10 device

**7. Testing Description**

The code was tested with various conditions, such as switching off the firewall, disabling password protection, uninstalling antivirus software and updating the windows software to the latest version. Since most of these security metrics produces binary results such as Fire wall enabled or Fire wall disabled being the only two options. The output was either true or false. But the updates portion and antivirus software list were endless you can either have one antivirus software or two or three. Similarly, you may have a single pending update or two pending updates or ten pending updates for the windows software. Thus, based on these results the output and security score would vary for these metrics. Hence there are various possibilities for variations in the security score. Sometimes the device lock maybe on but the fire wall would be switched off. In-some Cases the firewall would be on but there would be no antivirus software monitoring the system. But sometime there maybe an installed antivirus software that has not been enabled yet. So depending on these there are different possibilities for security scores.

**8. User’s Guide**

The user does not have to put any input for the program. The user simply needs a windows device and python to be installed. Running the script would produce this output. That gives a security score.

**9. Development Proccess**

The project was done by me alone, since I am working by myself in group 7. Pywin32 and Windows\_tools libraries were used to allow interaction with windows api through python.