ScamWatch Functional & Non-Functional Requirements

Anmol Mazoo, Arshdeep Singh, Kassandra Montgomery, Molik Verma

February 27, 2024

Info 4190 S50

Table of Contents

1.0 General Requirements 3
1.1 User Groups Use Cases 3
1.2 System Architecture – Data Flow Diagram 4
2.0 Functional Requirements 5
3.0 Non-Functional Requirements 5
3.1 Constraints 6
4.0 Tech Stack and Testing 6

1.0 General Requirements

ScamWatch is a downloadable software designed to provide an extra layer of protection against common scams, specifically scams that involve the use of remote-control access (RCA) applications. The primary functions of this software are to block the installation of RCA applications, block remote connections, and notify the user when a remote connection is attempted. The secondary function of ScamWatch is to help educate the user in an effort to help promote awareness.

The main target user group for ScamWatch are people that are most vulnerable to scams, such as those who have low tech literacy, are elderly, or have medical conditions. A secondary user group contains people who have medium-high tech literacy, for example, people who are concerned about their loved ones becoming involved in scams or businesses looking for an extra layer of protection for their employees.

1.1 User Groups Use Cases

User Group 1:

This user group is comprised of people who are most vulnerable to scams. These people may be elderly, have medical or health issues, are financially struggling, or are simply inexperienced or uneducated when it comes to technology. For Group 1, ScamWatch will hopefully not have to be used, but if it does, there are a few ways it may be used:

1. If the user downloads a remote access application, such as AnyDesk, and they try to install it, they will receive a pop-up telling them that the application is blocked and may be a scam. It will also suggest that they contact someone they trust, such as a family member or friend, and view more information.
2. The user may use ScamWatch to educate themselves on what to look for when talking to a potential scammer, how to deal with it, and who to reach out to for help.
3. If the user requires an RCA application for a legitimate use, they may either contact someone from User Group 2 to help unblock the application temporarily or input a password into ScamWatch.

User Group 2:

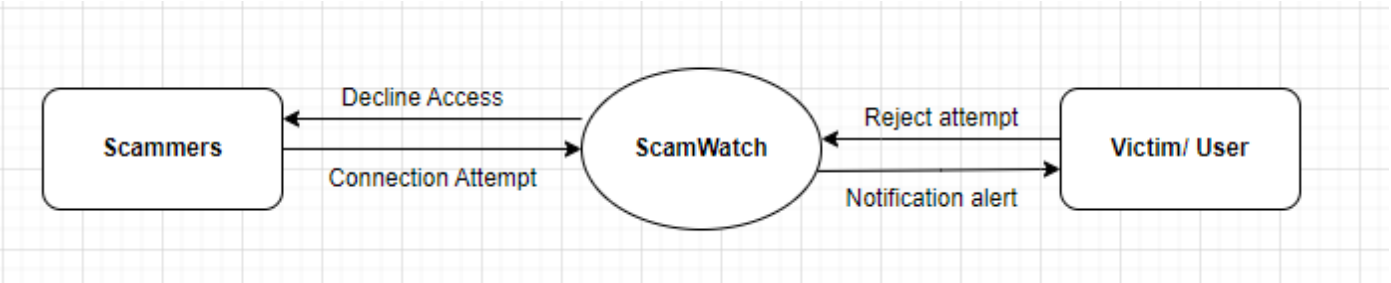
This user group contains people who are less vulnerable to scams, who generally have at least a working knowledge of technology and how to recognize scams. People in this group may have loved ones they are concerned for or may be part of a business and want to add an extra layer of protection onto the employee computers. This user group may use ScamWatch in the following ways:

1. The user may set up the program on the other person's computer, use their contact information to receive an alert, and set a password that is not given to the person who primary uses the computer. By doing this, if the computer user is caught in a scam, the other user must be contacted before allowing the connection.

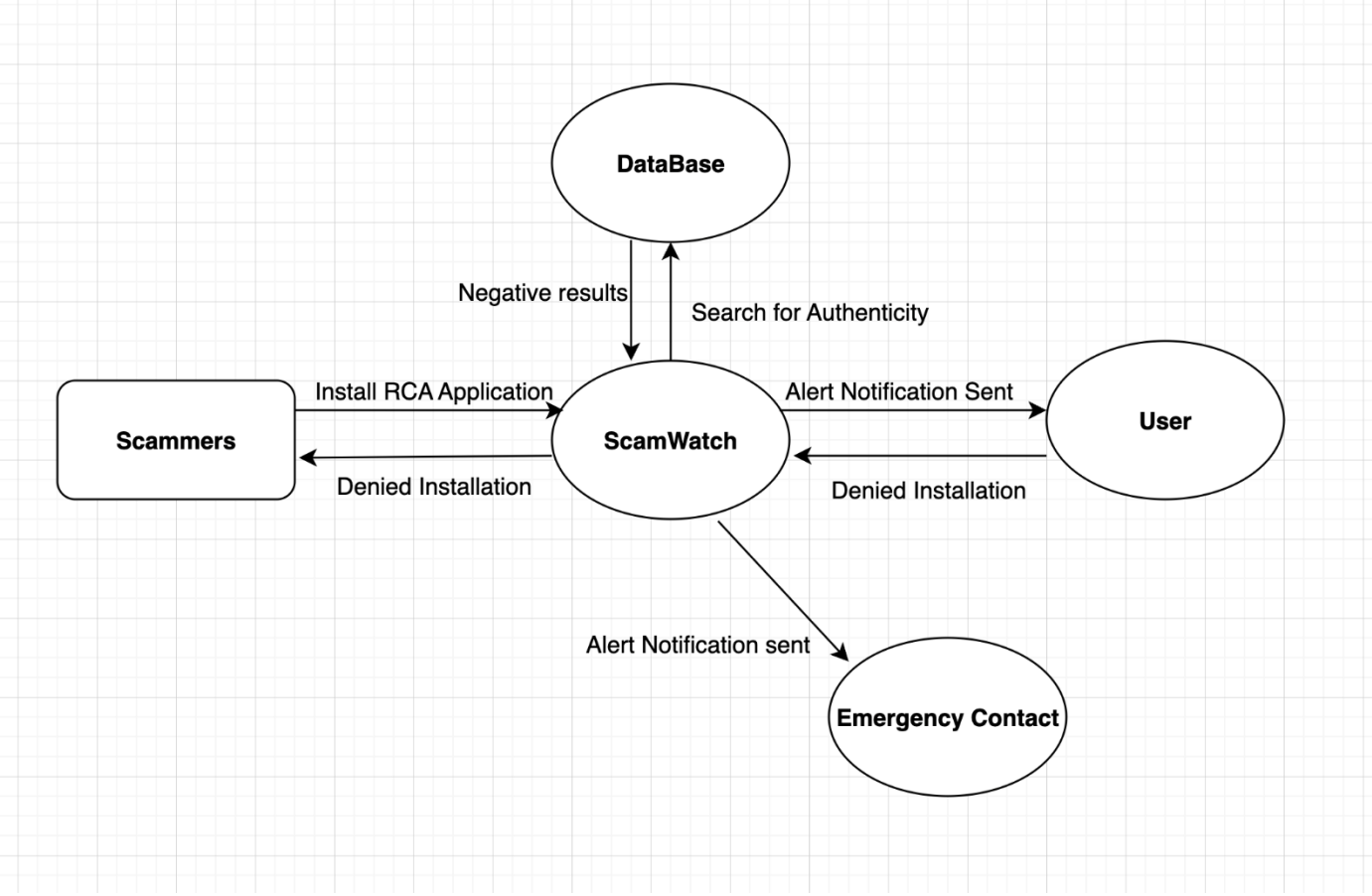
1.2 System Architecture – Data Flow Diagram

The following level 0 and level 1 diagrams provide an overview of data flow for ScamWatch. There are two primary points of data flow in and out of ScamWatch: a connection attempt between the scammer and the victim's computer and the victim's attempt at installing an RCA application, however there is interaction between a database, where information such as a login and common RCA applications is stored, as well as to an emergency contact, if set up.

Level 0:



Level 1:



2.0 Functional Requirements

The following shows a list of functional requirements, separated by milestones as outlined in ScamWatch's proposal:

- **Milestone #1: RCA Block Function**
  - ScamWatch should be able to block the installation of common RCA applications.
  - ScamWatch should be able to temporarily allow a one-time connection after the user enters a password.
- **Milestone #2: Connection Alert**
  - ScamWatch should be able to alert the user of any RCA attempts through, at minimum, a desktop pop-up.
  - ScamWatch should have an "emergency contact" option for the user to allow an alert to be sent to a trusted relative or friend.
- **Milestone #3: Knowledgebase**
  - ScamWatch should provide a knowledgebase and trusted links to educational content.
- **Milestone #4: UI**
  - ScamWatch should have a basic, easily navigable UI.
- **Milestone #5: Optimization**
  - ScamWatch should be optimized for best performance on user's system alongside monitoring for potential threats.
  - ScamWatch should run at OS start up.
- **Milestone #6: Final Product**
  - ScamWatch should be fully functional.

3.0 Non-Functional Requirements

The following is a list of non-functional requirements for ScamWatch:

- Users should be able to easily navigate ScamWatch, however the option to temporarily disable the software should not be obvious.
- The connection alert from ScamWatch should activate immediately when a connection attempt occurs.
- ScamWatch should not interfere with the user's day-to-day computer usage except to alert a connection.
- ScamWatch should be reliable and be available 99.99% of the time. If ScamWatch has an unexpected crash, it should provide useful error messages.
- ScamWatch should ensure the ease of future feature implementations and updates
- ScamWatch software should offer configurable notification settings, allowing users to customize the level of detail in alerts based on their preferences

3.1 Constraints

The following list outlines important constraints and limitations on this project.

- **Time Constraint**
  - This severely limits the possibilities of this software. For example, compatibility with multiple operating systems, the integration of AI, and a browser extension to block access to scam websites would be ideal functions, however they are not possible to implement within this timeframe.
- **Device Compatibility**
  - ScamWatch will currently only be compatible with modern Windows operating systems due to a lack of time and knowledge.
- **New Scams**
  - As new scams and new scam technology are constantly evolving, for long-term life, ScamWatch would need to be constantly updated, have access to common up to date anti-scam resources, or be integrated with AI, which is outside the scope of this project.
- **User Tech Literacy**
  - ScamWatch is created to protect users who may have low tech literacy which may impact the usability of the software.
- **Developers Abilities**
  - Some functions, such as the possibility of sending an alert to a different device or the use of multi-factor authentication, may be impacted by the developers' knowledge and within this timeframe, may be limited.
- **Budget Constraint**
  - As this is a student project, there is a budget constraint. This constraint will influence decisions regarding the services, tools, and programmatic options used.

4.0 Tech Stack and Testing

As ScamWatch is being developed by multiple people, GitHub will be utilized for programming collaboration. Python will be the main programming language used, utilizing its abilities to easily work with system interactions, such as the windows registry through "winreg". It will additionally be used to create a simple user interface, using the Tkinter framework, and to interact with a simple MySQL database, which will contain information such as contact information and the application password.

Testing will be done within a virtual machine using VMware or VirtualBox to ensure the safety of the developers' physical machines as ScamWatch will require system interactions. Each module or function will be thoroughly tested individually to ensure those are working as expected and if we catch any bugs, those will be noted down and fixed as we go along. Additionally, we will do integration testing to ensure that the application is working seamlessly. Further, we will run real-world scenarios to check the potential of our software and make improvements where necessary.