GALORATH CYBER SECURITY ADVISOR

Team Galorath: Joshue Shear, Nathan Evans, Joe Lynn, Ian Dansereau
Faculty Coach: Kal Rabb
Project Sponsor: Lee Fischman of Galorath Inc.
Special Thanks to Professor Meneely, Professor Malachowsky, Professor Łukowiak
Senior Project - 2018

Software Engineering
Rochester Institute of Technology

Data Management

Database Reasoning

The Galorath Cyber Advisor heavily relies on a relatively static database, only storing information necessary for the survey and results. Early on, we made the decision to not to store user data due to the immense risk storing such data would pose. The data we would have stored would not only outline specific ways to attack persons and companies, but common trends in ways to attack groups of persons or companies. By not storing the data, Galorath is not at risk of exposing its customers to targeted attacks from the data being compromised.

Answer Mapping

The most interesting part of the database is how answers are mapped to results. This is done through many layers, but the first layer in particular, where answers are mapped to the next layer, is done through minterms. Minterms are a computer engineering concept used to express logic circuits at their most basic level, combinations of sets of inputs that trigger an output. Answers in our database do much the same, the combinations of answers trigger the next layer. Given the truth table in table 1 with inputs X_0, X_1 and X_2 and output f we can derive equation 1, the minterm of the expression.

Table 1: Truthtable for the minterm in equation 1

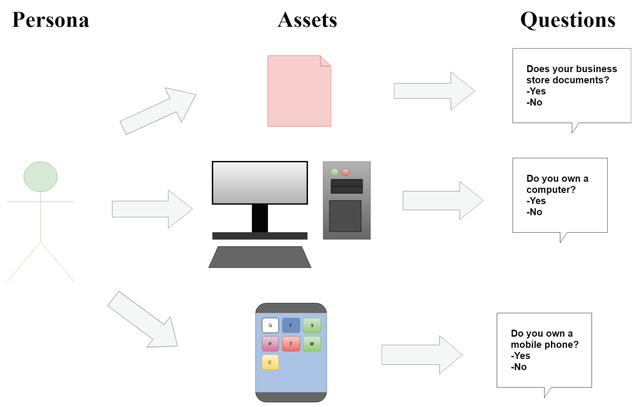
term	X_0	X_1	X_2	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

$$Equation1 = \overline{X_0}X_1X_2 + X_0\overline{X_1}\,\overline{X_2} + X_0\overline{X_1}X_2 + X_0X_1X_2 \quad (1)$$

In equation 1, the minterm items next to each other are logically anded. Blocks of items separated by + signs are logically orred. The overline represents not, so if we want X_0 to be false we say $\overline{X_0}$ Each orred block represents the logic nessicary when $f = 1$. We could say that X_0 is an answer to asking if your operating system is up to date, X_1 is an answer to a question asking if you are connected to the internet and X_2 is an answer for if the device is a mobile device. Using this logic with the answers we can allow certain answer combinations, or lack of certain answers to trigger elements in subsequent layers.

Data Creation

After some trial and error our team found creating personas and figuring out what questions they could answer and what questions they should be asked, was the best way to create questions for the survey. A persona is a fictional person that should represent a specific potential end user. Personas are generally used for user interface design (which we used them for as well), however we adapted the personas we created to work better for survey design. We used these personas to figure out what kinds of cyber security issues they could have, what devices or processes could leave them vulnerabile and brainstorm questions and answers for information we need to know to diagnose any cyber security issues they may have.

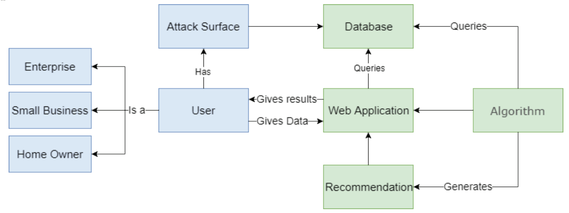


Adding Data to the Database

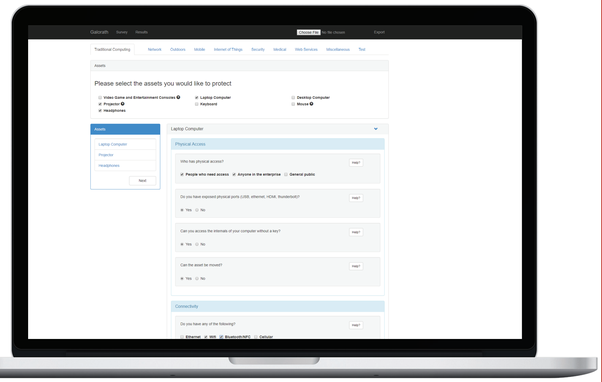
We initally did not anticipate the need for a program to enter in data to the database. As it turned out we underestimated the tediousness of entering hundreds of lines of SQL statements by hand. As a result we entered thousands of C# and JavaScript lines by hand to create a Database User Interface, a seperate application that allowed easy creation, reading, updating and destruction of the data in the database.

About

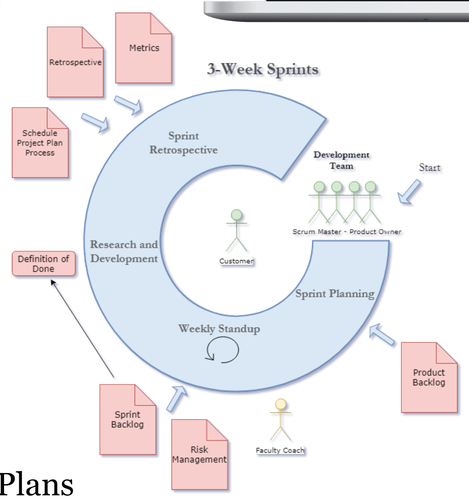
The Online Cyber-Security Advisor Service will allow anyone, be they a private individual, to security contractor, to CTO of a large enterprise to answer questions on their attack surface in a clean and understandable user interface, and be presented with several options to increase their cyber security. These options may be to password protect their smart thermostat, buy their IT department self encrypting hard drives or install swipe locks on their doors. To accomplish this, the team will need to craft an extensive survey that can be completed to varying levels of detail, allowing for novices and experts in cyber security to spend as much or as little time as desired filling it out. These questions will need to feed a highly adaptive algorithm that will analyze the results of the survey to grab the most pertinent cyber security improvements for the individual or business from a database.



User Experience



Process



Activities

- Planning poker
- Weekly standup
- Pair programming
- Retrospectives
- Burndown chart
- End of semester retrospective/role realignment
- Formal code review
- Living documents review

Metrics

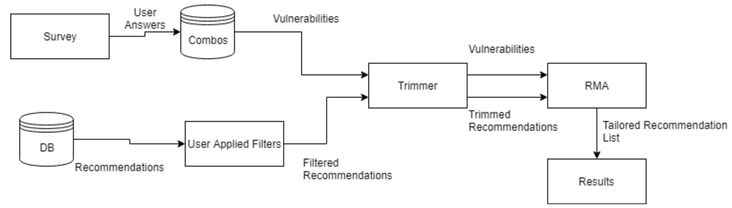
- Velocity
- Rework
- Software Usability Score

Future Plans

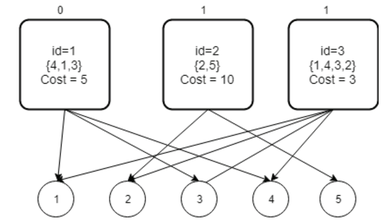
- Once an asset is checked in the survey by the user, an additional section pops up for the user to enumerate the quantity, give a label, a loss factor and enumerate any other groups of the same asset
- Unit Tests that hit the database
- Results page search
- Further UI customizations
- Script to convert from SQL server script to usable database data update script in the database UI
- Differential Evolution Algorithm research
- Save all items at once in the database UI
- Add support for optimized boolean expressions in addition to minterms
- Building an actual threat model for the user and for debugging
- Performance Refactoring for the Trimmer
- Results priority sorting

Algorithm

Architecture



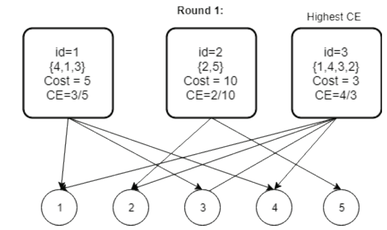
The Problem



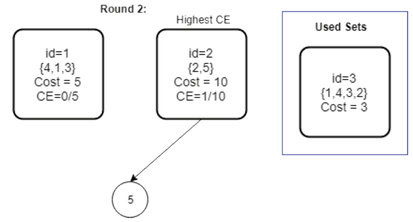
Choosing sets 1 and 2 would cover all of the elements, and yield a total cost of 15. However sets 2 and 3 would cover all elements, and yield a total cost of 13. How do we reliably pick the collection of sets to cover all elements, for the lowest total cost?

Repeated Set Cover

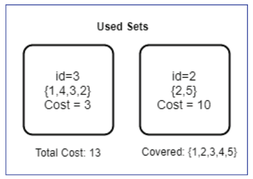
The classic Set Cover algorithm tries to find the most cost effective collection of sets to cover a collection of elements. Set Cover finds this solution by picking the set with the highest cost effectiveness(number of elements covered divided by the cost of the set), removing it and its elements from the scenario, and repeating this until either there are no more sets or no more elements.



In the first iteration, set 3 covers 4 elements with a cost of 3, yielding a cost effectiveness of 4/3. This is higher than the cost effectiveness of either of the other sets, so it gets picked.



After set 3 was picked, cost effectiveness was recalculated while ignoring all elements that set 3 covered. Set 1 no longer covers any elements, giving a cost effectiveness of 0, while set 2 covers the final element, giving a cost effectiveness of 1/10, so set 2 is chosen.



In the end, sets 3 and 2 cover all 5 elements, for a combined cost of 13.

The Repeated Set Cover runs the Set Cover algorithm until it reaches a run that is either more expensive or unsolvable than the one before it. Before each run, the first picked Recommendation of the previous run is removed from the input set of Recommendations, allowing Recommendations that may not be as cost effective individually to have a chance to work with others to produce a final set of a lower total cost.