**CS 4346- Project #1 Report**

**Connor Steed**

* **The Problem Description:** This expert system has been designed to identify the type of security attack on the user’s computer system, and to determine a solution for dealing with this kind of attack.
* **The Domain:** The domain of this problem is a computer system, which includes its users, applications, files, network, and permissions, among other things.
* **Methodologies:** This expert system uses two lists, a list of rules and clauses, and a list of conclusions that correspond to the rules. These lists will be used by the system’s logic to determine both the attack type, as well as the recommended solution. This system uses Backward Chaining to analyze the system’s current issue and other aspects of the system’s status to identify the kind of attack the system is experiencing. Then the system uses Forward chaining to determine a solution to the issue based on the type of attack and further information about the current system status.
* A picture containing table

  Description automatically generatedDiagram

  Description automatically generated**Decision Trees:**
* **Rules:**

Backward Chaining:

issue computer

visible yes

actions yes

openprograms yes

attack identity

issue computer

visible yes

actions yes

openprograms no

attack malicious\_code

issue computer

visible yes

actions no

slow no

changedfiles yes

damage yes

attack malicious\_code

issue computer

visible yes

actions no

slow no

changedfiles yes

damage no

attack identity

issue computer

visible yes

actions no

slow no

changedfiles no

attack basic

issue computer

visible no

disappear no

attack basic

issue computer

visible yes

actions yes

openprograms yes

attack identity

issue network

access yes

loadtime yes

compass yes

attack identity

issue network

access yes

loadtime no

popup yes

people yes

attack malicious\_code

issue network

access yes

loadtime no

popup yes

people no

attack basic

issue network

access yes

loadtime no

popup no

attack denial\_of\_service

issue network

access no

networkshow yes

serviceload yes

networktype work

attack basic

issue network

access no

networkshow yes

serviceload no

attack malicious\_code

issue network

access no

networkshow no

showbefore yes

showname yes

attack identity

issue network

access no

networkshow no

showbefore no

attack basic

**Forward Chaining:**

attack denial\_of\_service

issue network

access no

networkshow yes

serviceload yes

networktype work

suggestion install\_company\_firewall

attack basic

issue computer

visible yes

actions no

slow no

changedfiles no

suggestion grant\_admin\_permissions

attack basic

issue computer

visible no

disappear no

suggestion check\_monitor\_cables

attack basic

issue network

access yes

loadtime no

popup no

suggestion check\_blocked\_websites

attack basic

issue network

access no

networkshow yes

serviceload yes

network personal

suggestion check\_router\_and\_modem\_cables

attack basic

issue network

access no

networkshow no

showbefore no

suggestion check\_router\_and\_modem\_power

attack malicious\_code

issue computer

visible yes

actions yes

openprograms no

suggestion create\_admin\_file\_lock

attack malicious\_code

issue computer

visible yes

actions no

slow no

changedfiles yes

damage yes

suggestion use\_static\_code\_analysis

attack malicious\_code

issue network

access yes

loadtime no

popup yes

people no

suggestion stop\_visiting\_sketchy\_websites

attack malicious\_code

issue network

access no

networkshow yes

serviceload no

suggestion stay\_on\_trusted\_network

attack identity

issue computer

visible yes

actions yes

openprograms yes

suggestion keep\_login\_private

attack identity

issue computer

visible yes

actions no

slow no

changedfiles yes

damage no

suggestion use\_seperate\_user\_accounts

attack identity

issue network

access yes

loadtime yes

commonpass yes

suggestion use\_different\_passwords

attack identity

issue network

access yes

loadtime no

popup yes

people yes

suggestion monitor\_network\_website\_traffic

attack identity

issue network

access no

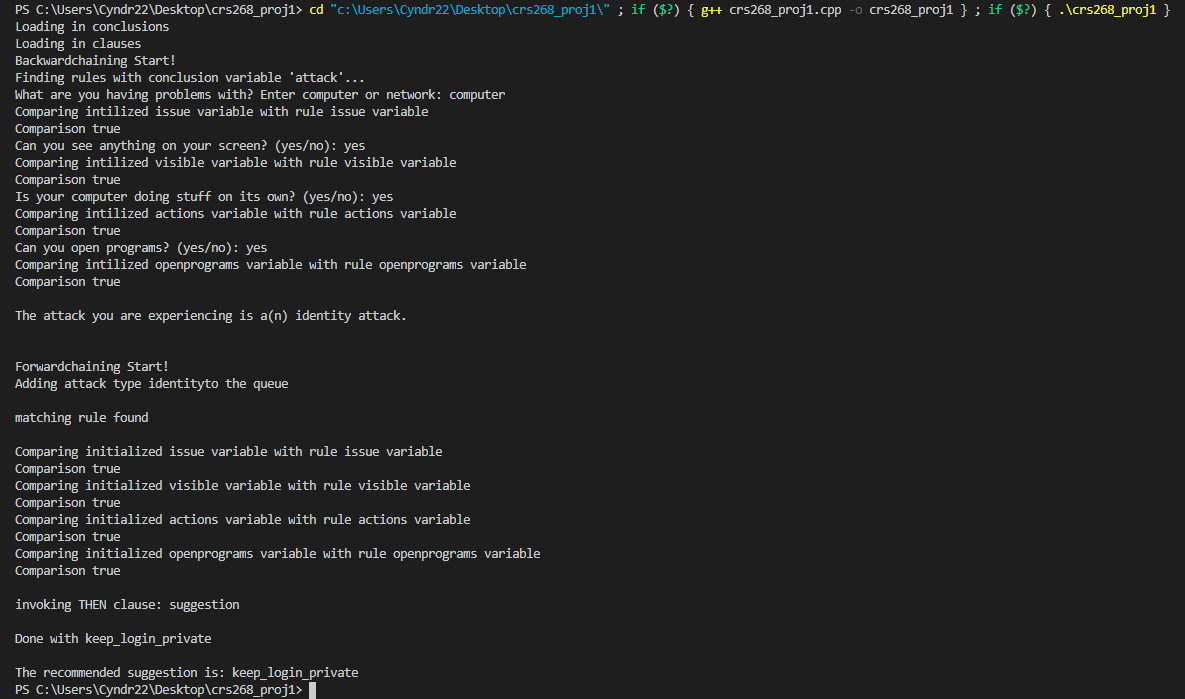
networkshow yes

showbefore no

listshow yes

suggestion limit\_internet\_password\_control

* **Program Implementation:** The program uses a function called getRules which reads the contents of a clauses.txt file into the clausesToRules vector and reads the contents of a conclusions.txt into the conclusionList vector. The program also utilizes an unordered variable map to store instantiated variables. Variables are instantiated based on the rules and some user-answered questions. The program first employs a backward chaining methodology in a function called Attacks\_BW to identify the attack on the system. This function employs a queue to hold the rules that need to be analyzed, and it iterates through the queue and initializes and compares variables until an attack is reached, which is pushed to the next function. The Prevention\_FW function analyzes the data via forward chaining by taking the attack found in the prior function and finding rules that match that attack type in the list. These rules are added to a new queue, and each rule in the queue, and all of its variables, is inspected, initialized, and compared until a valid suggestion is found and the queue is empty. The program then prints out the recommended suggestion.
* **Source Code:** Included in assignment submission
* **A Copy of the program Run:**



* **Analysis of the program:** We utilized vectors, queues, and variable pairs to keep track of the rules, conclusions, and variables in a more efficient way than the sample program. Within the Backward and Forward chaining algorithms, we utilized while loops and for loops instead of GoTos to search and manipulate the data more effectively, as well as just to make the code more readable and organized. Due to some edge cases, we decided a queue would be better for the Backward chaining, so we used that instead of a stack. Also because of abnormal variable answers (answers besides yes or no), we have some hard-coded questions in our initializer function.
* **Analysis of the results:** Our team had some issues with communication at first, but we broke up the tasks efficiently. Kobe made the diagrams and wrote all of the rules. Chance designed the program structure, wrote the file reader, the initializer, the Forward Chaining algorithm, and debugged. I wrote the Backward Chaining algorithm, debugged, wrote documentation, and wrote the txt documents for the ruleset. Overall, the program effectively deals with the given ruleset and user answers and can come to the correct conclusion quickly based on the data it receives.
* **Conclusion:** This project was difficult at first due to the readability and issues with the original source code, as well as my personal understandings of the chaining algorithms and our team’s communication issues, but at the end of the project, I now fully understand both algorithms. We managed to change the organization and structure of the given program, implement new, more complex and efficient data structures, and respond more dynamically to edge cases in the problem.
* **References:**

<https://phoenixnap.com/blog/cyber-security-attack-types>  
<https://us-cert.cisa.gov/ncas/tips/ST04-015>  
<https://www.veracode.com/security/malicious-code#:~:text=Malicious%20code%20is%20the%20term%20used%20to%20describe,be%20efficiently%20controlled%20by%20conventional%20antivirus%20software%20alone>.  
<https://www.thewindowsclub.com/dos-denial-of-service-attack#:~:text=%20DoS%20attack%20prevention%20%201%20Deploy%20an,attacks.%20These%20can%20be%20expensive%20but...%20More%20>