PSYCH DEFENDER

Group Members:
Fatima Ali 2024173
Muhammad Saad 2024445
Suleman Malik 2024472

PROBLEM

Humans are far easier to hack than any computer

Fake domains like *microsoft* vs *rnicrosoft*, Using Cyrillic characters (e.g., 'a' vs 'a') to trick eyes.

Manipulative language forcing users to click without thinking.

OUR SOLUTION

WHAT IS PSYCH DEFENDER?

A multi-layered detection engine that analyzes both Message Content and URL Structure.

KEY MODULES:

- CONTENT ANALYZER: SCANS FOR KEYWORDS, HIDDEN FILE RISKS, AND URGENCY PATTERNS.
- DOMAIN ANALYZER: CHECKS FOR VISUAL SPOOFING, SIMILARITY TO TRUSTED SITES, AND HEURISTIC FLAWS.
- EXPLAINABILITY LAYER: PROVIDES A BREAKDOWN OF WHY AN ALERT WAS TRIGGERED.

DATA STRUCTURES

Trie (Prefix Trees)

Hash Maps

Linked List

Stacks

Arrays

2D Matrix DP Table

Wide String

N-gram Model

KEYWORD & FILE ANALYSIS

FEATURE

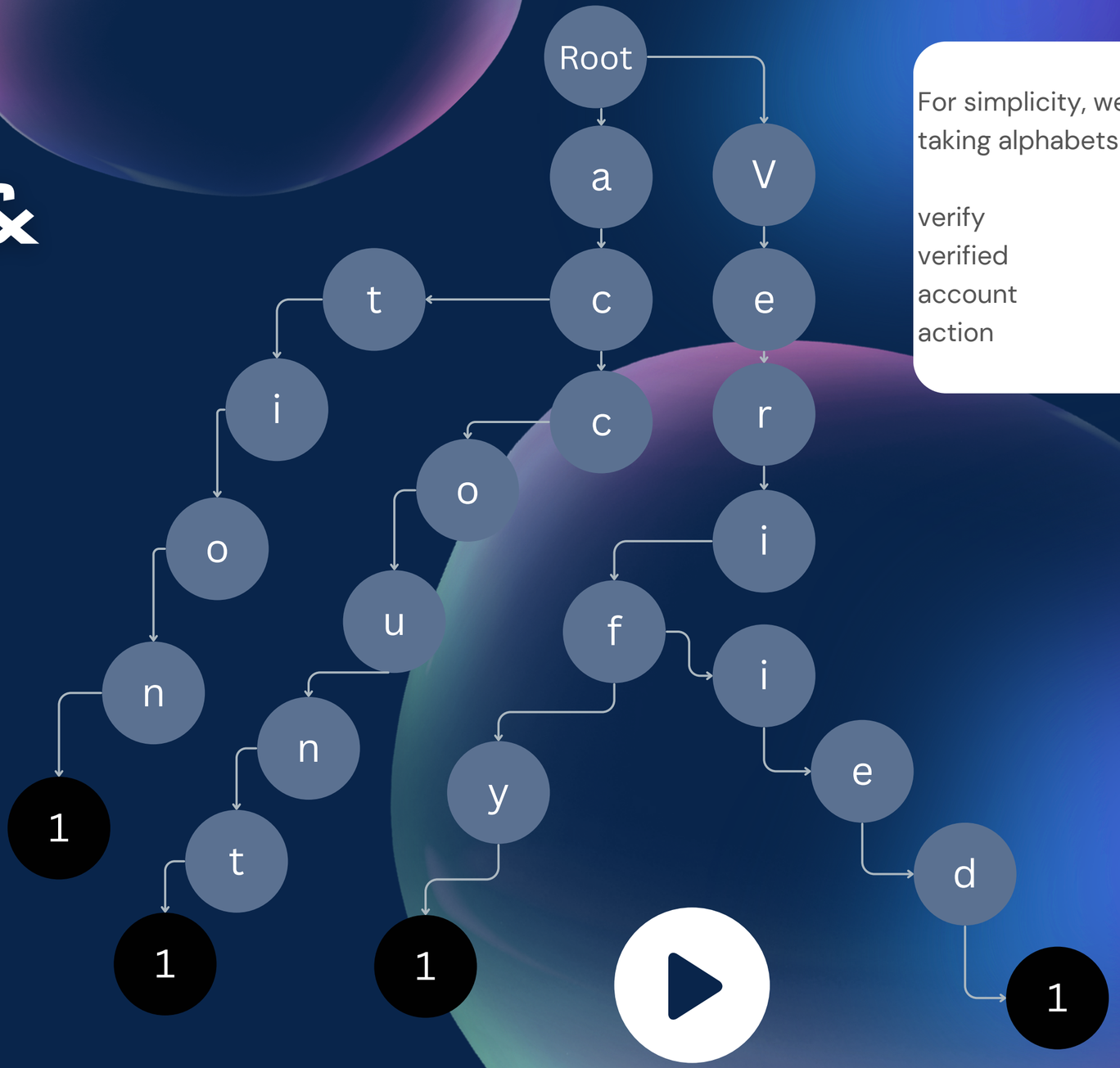
Keyword Scanner & Attachment Risk Analyzer.

DATA STRCUTURE

Trie (Prefix Tree).

ALGORITHM

Aho-Corasick Approach.



For simplicity, we are taking alphabets here.

verify
verified
account
action

N-GRAMS

Input: It is a limited time offer.
Keywords {limited time, offer}

word= It
prev= It
prev2= It

word= is
prev= It is
prev2= It is

word= a
prev= is a
prev2= It is a

word= limited
prev= a limited
prev2= is a limited

word= time
prev= limited time
prev2= a limited time

word= offer
prev= time offer
prev2= limited time offer

TRUSTED DATABASE

Created trusted_domains.txt, a database of verified websites used to cross-reference and validate user input.

- Feature: Trusted Domain Lookup.
- Data Structure: Hash Table with Chaining.
- Implementation Hashtable
- Collision Resolution: Uses a Singly Linked List at each bucket (struct Node).

DOMAIN SIMILARITY

		s	e	t	t	i	n	g
	0	1	2	3	4	5	6	7
s	1	0	1	2	3	4	5	6
i	2	1	1	2	3	3	4	5
t	3	2	2	1	2	3	4	5
t	4	3	3	2	1	2	3	4
m	5	4	4	3	2	2	3	4
g	6	5	5	4	3	3	3	3

- Data Structure: 2D Array (Dynamic Programming Table).
- Algorithm: Levenshtein Distance (Edit Distance).

FUTURE IMPLEMENTATION

Real-Time Browser Integration ("Grammarly for Security"):
Scaling the project into a browser extension that runs in the background. Similar to how Grammarly checks grammar in real-time, this tool would actively monitor the browser's address bar and warn the user immediately if they visit a suspicious site.

Machine Learning:
Replacing the static keyword heuristics with a trained ML model for better accuracy on novel phishing phrases.

Live API Integration:
Connecting to real-time blacklists (like Google Safe Browsing API) instead of a static text file.

THANK YOU