Basic Details of the Team and Problem Statement

PSID: KVH-014

Problem Statement Title: Chat messenger decryption tool

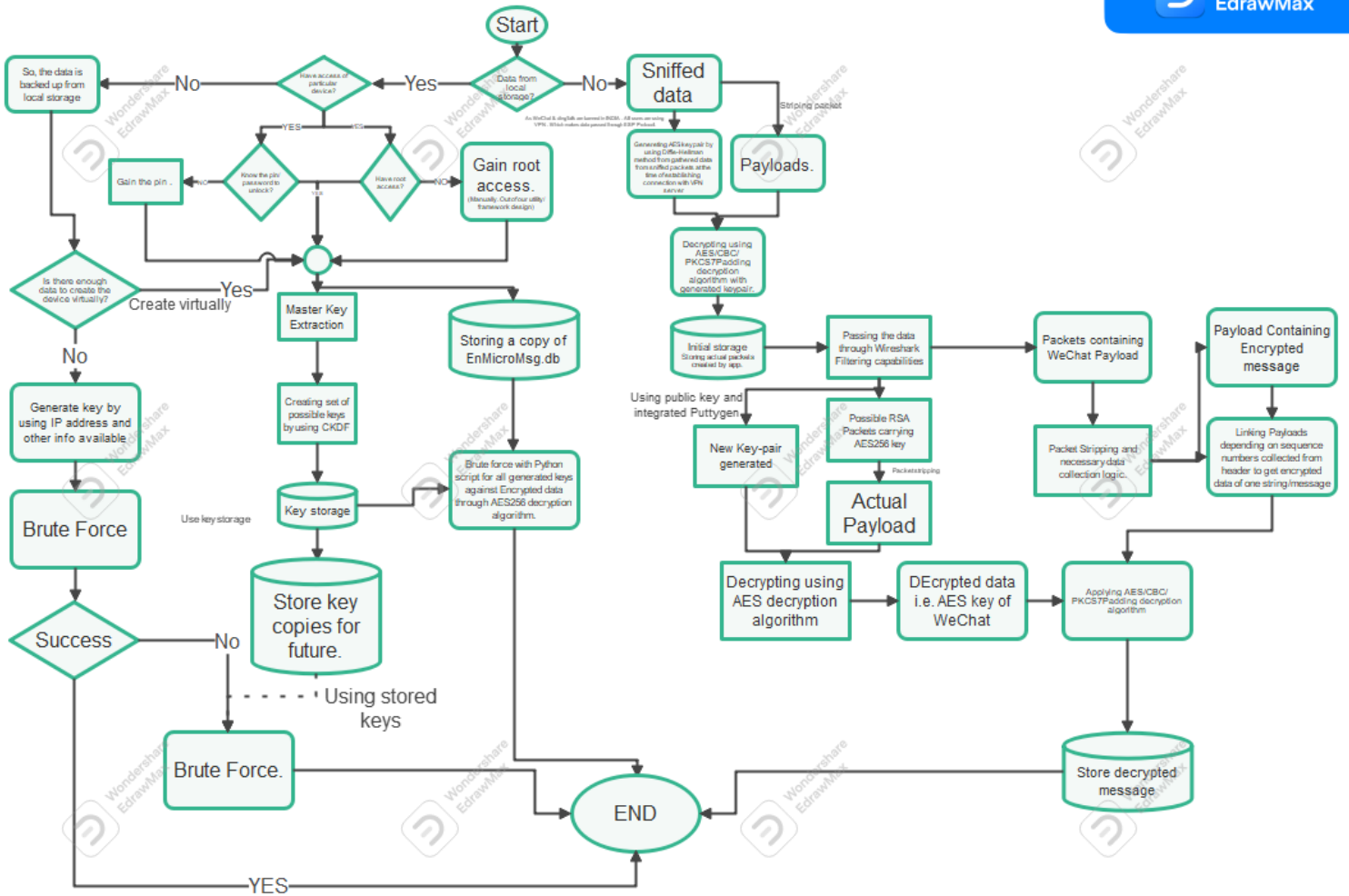
Team Name: Enigma Hunters

Team Leader Name: Durjoy Majumdar

Institute Code (AISHE):

Institute Name: Koneru Lakshmaiah Education Foundation

Idea/Approach Details



For clear view of the diagram please visit - [diagram link]

Idea/Solution/Prototype:

- Due to the complexity of WeChat and DingTalk's encryption methods and for protecting the work instead of a portable utility, we propose one framework with a web interface. (Based on Kali Linux)
- Our framework can decrypt WeChat messages in three scenarios. 1) Have access to the device 2) Have backup from the device but no access to the device. 3) Have sniffed data.
- Depending on the scenario it takes multiple steps that differ from one another but the ultimate target is getting the AES256 key and decrypting data.
- It is not possible to describe the process in this limited space. We are requesting from the core of our hearts to view the pdf.
- PDF LINK

Technology stack:

Base operating system – Kali Linux. Programming Language – Python, java. Development Environment – Visual Studio.
Build Tools – GNUTools (primary) Package Manager – ATP Package Manager. Integration Tools – Bash and Python Scripts

Idea/Approach Details

Use Cases:

- Decrypting WeChat or DingTalk messages from the cloud or local storage.
- Actors: User, Web interface.
- Preconditions: The user must prove his/her identity for the authentication and authorization process.
- Postconditions: The user gets an output of plain text.
- Flow-1: (Having access to the device)
  - Connect the device with a cable to the device.
  - Provide root access and enter unlock code.
  - When the key generation is done upload the chat database.
  - Run the brute force.
  - Get output.
- Flow -2 : (Having device data not device)
  - Make a virtual device with existing data.
  - If possible then from here onwards follow flow-1.
  - If not, upload the database.
  - Fill in as much required data as possible.
  - Run brute force.
  - Get output.
- Flow – 3: (Having sniffed data)
  - Make sure of the completeness of data.
  - Upload the data and start the process.
  - Wait some time and get the output.

Dependencies:

The proposed solution need,

- To be set up on the Debian distribution of Linux for giving the best performance.
- Python version 3.10+
- Wireshark API
- Super user access on OS.
- CPU must support SSE2 instruction set.
- C compiler.
- CMake.
- The proposed solution also use some tool of Kali Linux as ‘Mentalist’ or ‘John The Ripper’ so all of their dependencies also need to be satisfied.

Team Member Details

Sr. No.+	Name of Team Member	Branch	Stream :	Year	Position in team
1	Durjoy Majumdar	Btech	CSE	2nd	Team Leader
2	Asadur Zaman Nabin	Btech	CSE	2nd	Encryption Expert, Full Stack Developer
3	Mundra Lakshmiprasanna	Btech	Cs&IT	2nd	Tools Expert, Front End Developer
4	Jakka shirini	Btech	Cs&IT	2nd	Linux Expert, Back End Developer.
5	Anuj Kandel Sharma	Btech	CSE	2nd	Data Management & Analysis, Code Analysis
6	Sambandha Bhattarai	Btech	CSE	2nd	Code Analysis, Tools expert.

Team Mentor/s Details

Sr. No.	Name of Mentor	Category (Academic/Industry):	Expertise (AI/ML/Blockchain etc):	Domain Experience (in Years)
1	Dr. Radhika Rani Chintala	Academic		