Project Report: Educational Security Research Tool

Introduction

This project, an "Educational Security Research Tool," is a Proof-of-Concept (POC) designed for security research and demonstration purposes. It illustrates functionalities commonly found in malware, such as keystroke capture, remote data exfiltration, and command and control (C2) capabilities, primarily using Telegram as a communication channel. The tool is strictly for educational and ethical use, emphasizing that unauthorized use is illegal and unethical.

Abstract

The "Educational Security Research Tool" demonstrates various techniques for data collection and remote control within a system. Key features include cross-platform keystroke capture, remote data exfiltration to a Telegram bot, basic remote command execution, system information gathering, screenshot capture, clipboard monitoring, microphone recording, webcam capture, and network information collection (public and local IP addresses, and IP-based geolocation). The project highlights the mechanisms behind such tools to foster a deeper understanding of cybersecurity threats and defensive measures.

Tools Used

The project is primarily developed in Python and utilizes several libraries and tools for its functionalities:

- **Python**: The core programming language.
- **pynput**: For monitoring and controlling input devices, specifically for keystroke capture.
- **requests**: For making HTTP requests, used for Telegram API interaction and fetching IP information.
- **cryptography**: For encryption, specifically Fernet for secure communication.
- **python-dotenv**: For managing environment variables (BOT_TOKEN, CHAT_ID, ENCRYPTION_KEY).
- **psutil**: For retrieving system and process information.
- **PyAudio**: For audio recording from the microphone.
- **opencv-python (cv2)**: For webcam capture.
- **Pillow (PIL)**: For image manipulation, particularly for screenshot capture.
- **pywin32**: Windows-specific library for clipboard monitoring and potentially other system interactions.
- **socket**: For network communication, specifically for obtaining local IP addresses.

- **subprocess**: For executing system commands.
- **threading**: For managing concurrent operations.
- **colorama** and **pyfiglet**: Optional dependencies for enhanced terminal output.

Steps Involved in Building the Project

1. **Repository Cloning**: The initial step involves cloning the project repository from GitHub.
2. **Virtual Environment Setup**: A Python virtual environment is created and activated to manage project dependencies in isolation.
3. **Dependency Installation**: All required Python libraries are installed using `pip install -r requirements.txt`.
4. **Telegram Bot Setup**: A new Telegram bot is created via BotFather, and the `BOT_TOKEN` and `CHAT_ID` are obtained for communication.
5. **Environment Variable Configuration**: A `.env` file is created to store sensitive information like the Telegram bot token, chat ID, and a generated Fernet encryption key.
6. **Core Logic Development**: The `telegram_keylogger.py` script contains the main logic for:
   - **Keystroke Logging**: Capturing and storing keyboard inputs.
   - **Data Exfiltration**: Sending captured data and system information to the configured Telegram chat.
   - **Command and Control (C2)**: Implementing basic remote command execution capabilities.
   - **System Information Collection**: Gathering details about the operating system, CPU, memory, etc.
   - **Screenshot, Clipboard, Microphone, Webcam Capture**: Integrating functionalities to capture various forms of data from the target system.
   - **Network Information**: Collecting public and local IP addresses and geolocation data.
7. **Execution**: The main script `telegram_keylogger.py` is executed to start the monitoring and data exfiltration process.

Conclusion

This educational security research tool effectively demonstrates the principles and functionalities of a Telegram-based keylogger and remote control system. By showcasing features such as keystroke capture, data exfiltration, and C2 capabilities, the project provides valuable insights into how such malicious tools operate. It serves as a critical resource for understanding potential vulnerabilities and developing robust cybersecurity defenses. The project strictly adheres to ethical guidelines, emphasizing its use for educational purposes only to prevent misuse and promote responsible security research.