DECEPTIQ

x33fcon 2025

Build Your Own Ransomware

Hands-On Offensive and Defensive Insights

DECEPTIQ

whoarewe🥁

discord.gg/onlymalware

- ❖ Rad Kawar (@rad9800)
  - ➢ Inexperienced Malware Developer
  - ➢ Red Teamer ❤️‍🔥
  - ➢ Founder
- ❖ Paul Ungur (@C5pider)
  - ➢ Contractor for the Private/Public Sector
  - ➢ Experienced Malware Developer

Workshop Goals

- Understand how ransomware works
  - Enumeration
  - File Encryption
  - Exfiltration
  - Evasion
- How ransomware is detected
  - What does and doesn’t work
- Write your own ransomware
  - So you can evaluate your/client security controls

if you have any questions feel free to interupt

Agenda

Setup

Fundamentals

Implementation

Security Controls

Evasion

Setup

Environment Setup

- Golang (all operating systems)
  - https://go.dev/doc/install
  - https://code.visualstudio.com/download
  - https://code.visualstudio.com/docs/languages/go
- C/C++ (Windows) - Optional
  - Visual Studio

Ransomware Development Environment

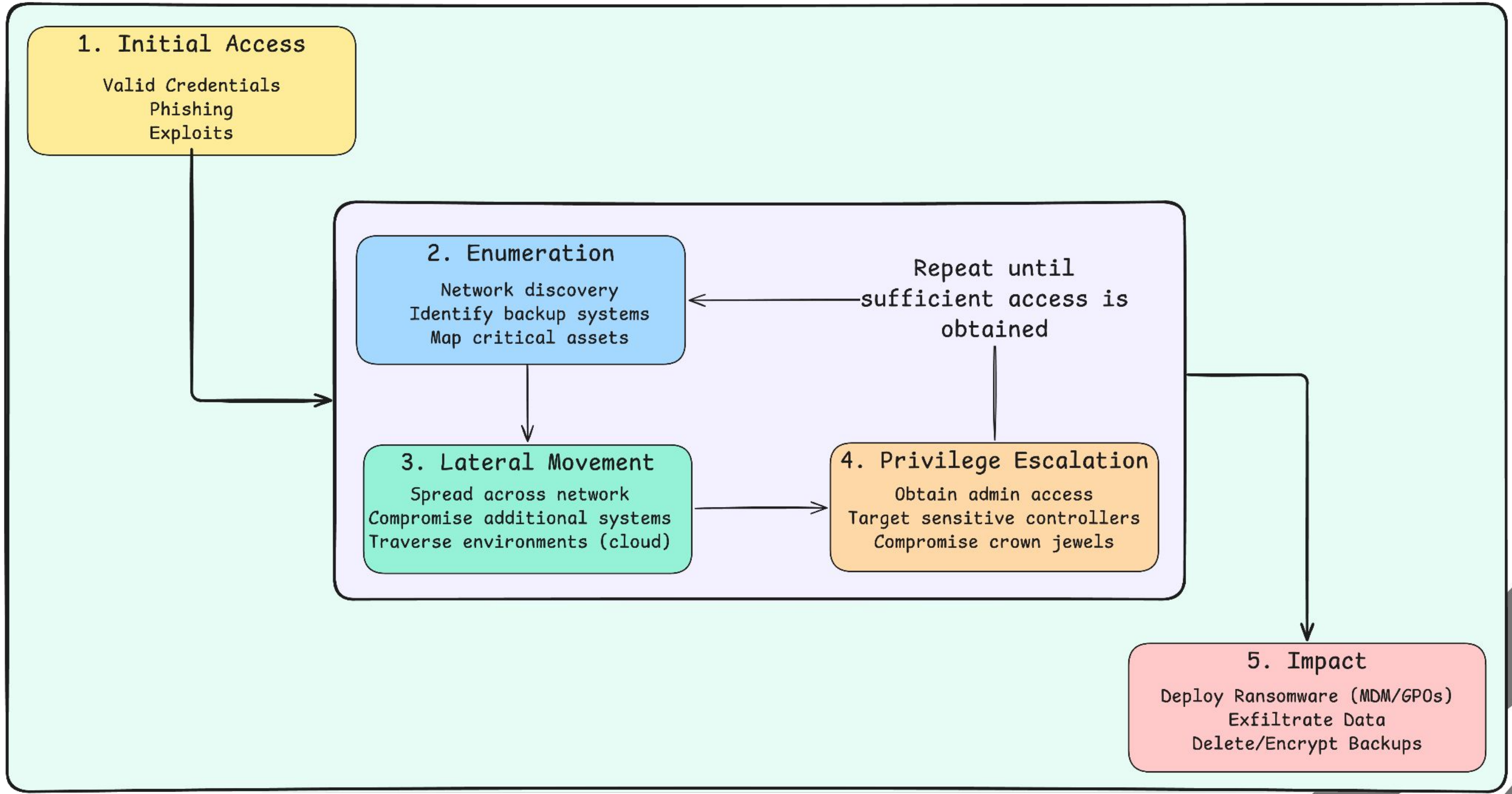
Feel free to use whatever you are comfortable with

- Base Operating System - Anything
  - No Virtual Machine required
    - If you have, you’re welcome to use it
  - We will dive into evasion focused on Windows
    - However, the ideas will be just as valuable for Linux/MacOS
- Set up a language and IDE of your choosing
  - Examples will be in Golang/C++ primarily
  - ANYTHING WORKS FOR 85% OF IT

Fundamentals

DECEPTIQ

Ransomware Lifecycle



1. Initial Access

Valid Credentials
Phishing
Exploits

2. Enumeration

Network discovery
Identify backup systems
Map critical assets

3. Lateral Movement

Spread across network
Compromise additional systems
Traverse environments (cloud)

4. Privilege Escalation

Obtain admin access
Target sensitive controllers
Compromise crown jewels

Repeat until sufficient access is obtained

5. Impact

Deploy Ransomware (MDM/GPOs)
Exfiltrate Data
Delete/Encrypt Backups

Pre-Encryption Operations

- Prevent Recovery
  - Tamper with backups
    - Local
      - Volume Shadow Copy Service (VSS)
    - Remote
- Evade/Disable EDR
  - Stop/Terminate Services
  - For example, on Windows
    - Reboot into Safe Mode
    - Leverage BootExecute to run before Win32
    - Leverage vulnerable drivers to kill services

Backup servers are prime targets for direct attack, providing access to concentrated sensitive data for encryption and exfiltration

Exfiltration

- Leverage Even With Backups
  - Data breach notifications required
  - Regulatory fines (GDPR, HIPAA)
  - Reputational damage
  - Secondary extortion threats

- Exfiltration Methods
  - Legitimate Cloud Services
  - File sync tools (Rclone, MegaSync)
  - Cloud storage (S3, Azure Blob, R2)
  - File sharing sites (anonymously)
  - Direct Transfer
    - FTP/SFTP to attacker infrastructure
  - Custom exfiltration tools

- Timing & Approach
  - Can occur before OR after encryption
  - Manual Exfiltration
    - Operators identify high-value data
  - Automated Exfiltration
    - Search by file extensions
    - Or exfiltrate everything

github.com/BushidoUK/Ransomware-Tool-Matrix/blob/main/Tools/Exfiltration.md

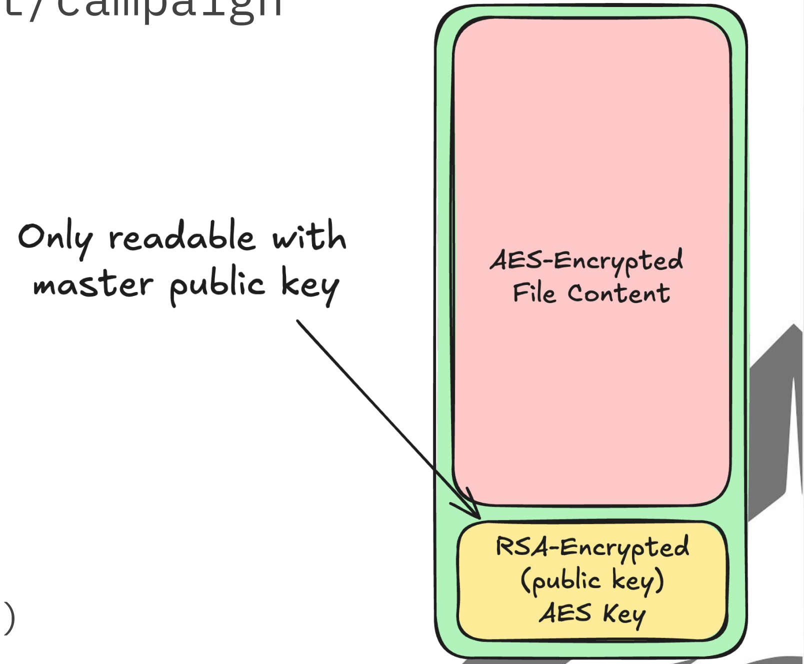
Encryption

Ransomware = File System Enumeration + Encryption

- Multi-Threaded
- File System Enumeration
  - Depth First Search
  - Breadth First Search
- Based on file size/extension
  - partial/full encryption to prevent recovery
  - .pdf = full encryption
  - .vmdk = partial encryption
- Encrypt the file to either the same/new file
  - Rename the file with an extension
  - Or if was to a new file, delete the original file

Traditional Hybrid Encryption: RSA + AES

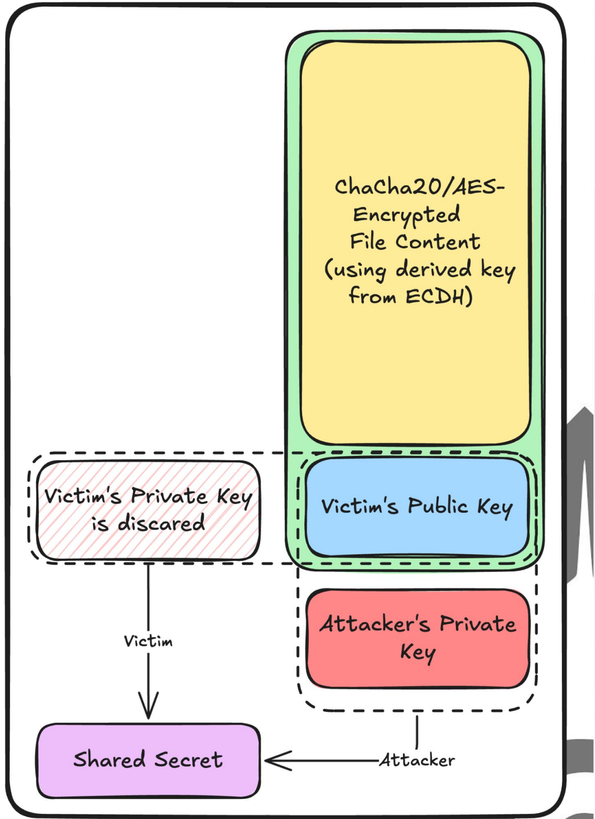
- New build created for each target/campaign
    - Unique public key per campaign
    - Matching decryptor with private key
- Builder generates master key pair
    - Public/Private Key (Curve25519 master)
        - Encryptor <- Public Key
        - Decryptor <- Private Key
- For each file:
    - Generates new random AES key (victim)
    - Encrypts file contents
        - AES-256 in CBC/CTR mode
- Encrypts the AES key
        - RSA(victim AES, master public)
    - **Appends encrypted AES key to file**
    - Discards plaintext AES key (in memory)


AES-Encrypted File Content
RSA-Encrypted (public key) AES Key

Only readable with master public key

Modern Hybrid Encryption: ECDH + Stream Cipher

- Builder generates master key pair
  - Public/Private Key (Curve25519 master)
    - Encryptor <- Public Key
    - Decryptor <- Private Key
- For each file, the ransomware
  - Generates new ephemeral key pair (victim)
  - Computes shared secret
    - ECDH(victim private, master public)
  - Derives encryption key
    - SHA256(shared_secret)
  - Using derived key encrypt with ChaCha20/AES
  - **Appends victim's PUBLIC key to file (unencrypted)**
  - Discards victim's private key


ChaCha20/AES-Encrypted File Content (using derived key from ECDH)
Victim's Private Key is discared
Victim's Public Key
Attacker's Private Key
Victim
Shared Secret
Attacker

Implementation

Time to Write Your Own Ransomware

- Implement hybrid encryption
  - Master + Ephemeral Keys
  - Encryption + Decryption
    - String - one program
    - File - one program
    - File - two program
- File System Enumeration
  - Depth or Breadth First Search
  - Print out the files you discover
  - Realize then you want to skip certain folders

If you
- get stuck
- have questions
- want a challenge
just ask 🙋‍♀️🙋‍♂️

Ransomware = File System Enumeration + Encryption