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

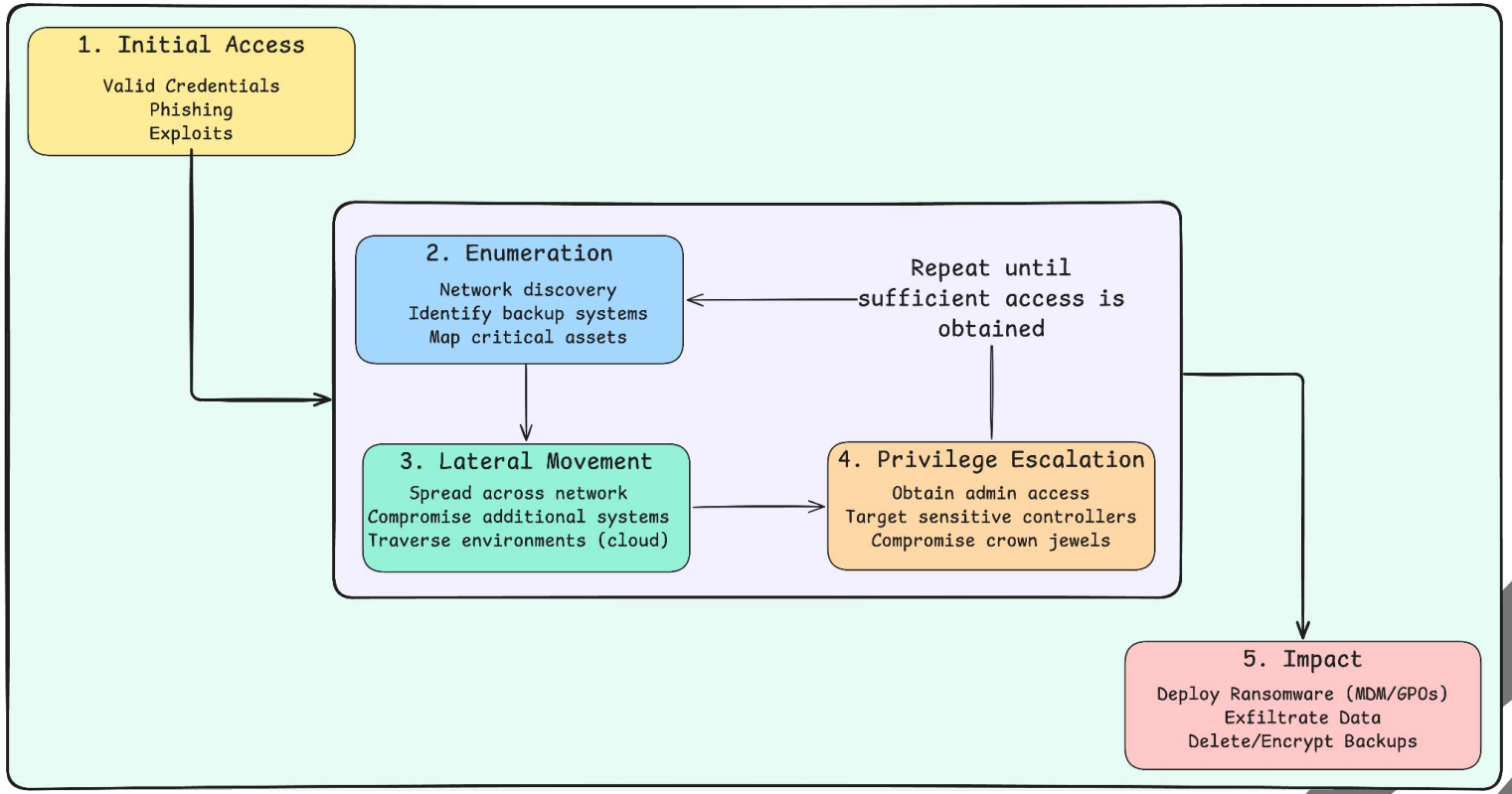
Ransomware Development Environment

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

Ransomware Lifecycle



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

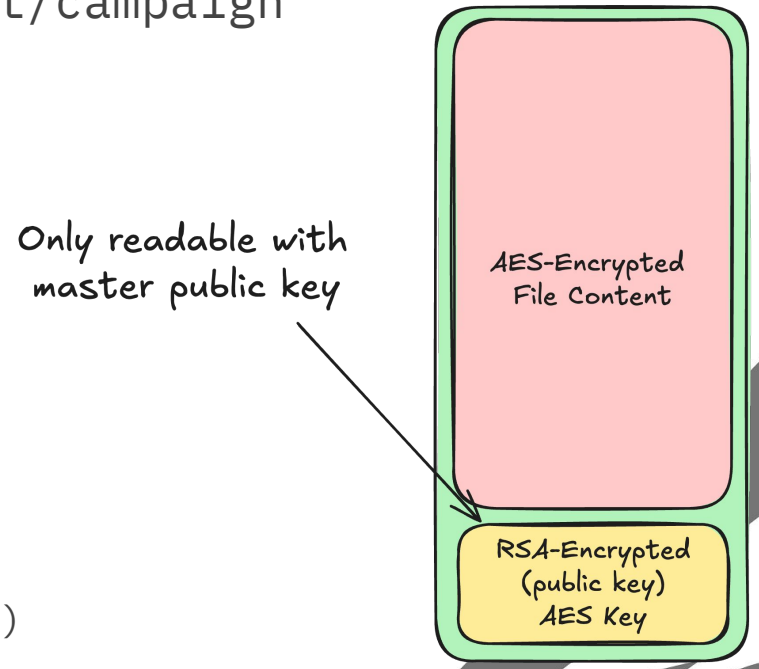
Encryption

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

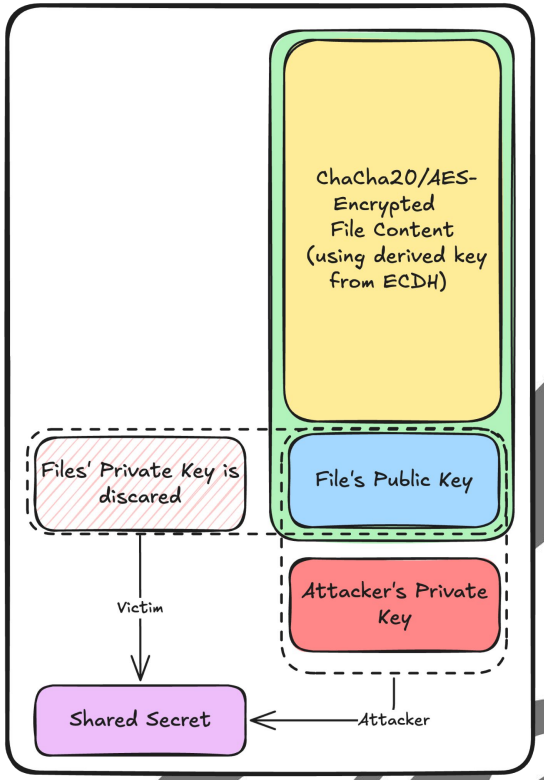
Traditional Hybrid Encryption: RSA + AES

- 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)



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



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

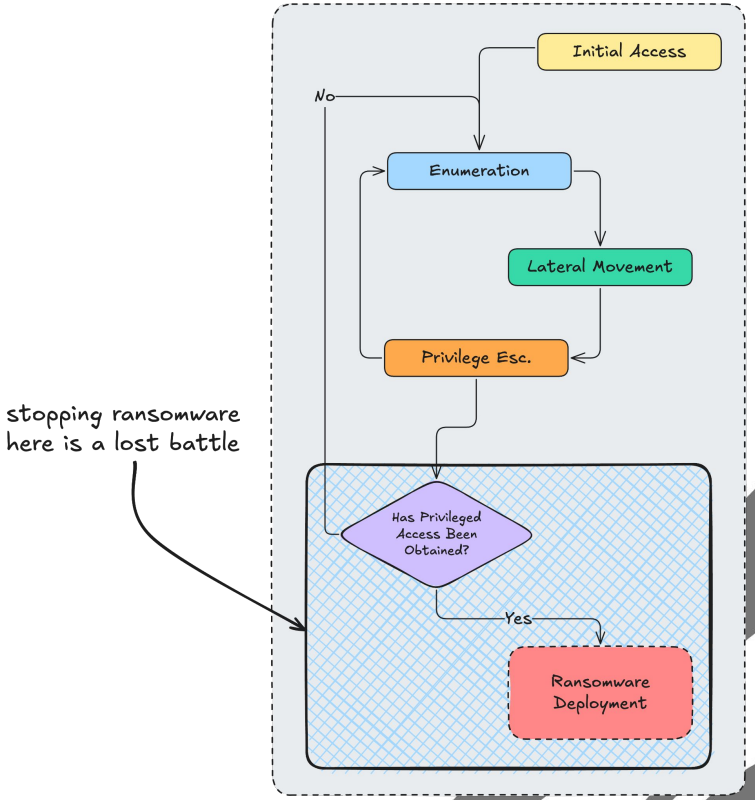
https://github.com/rad9800/byor/

Security Controls

Safeguards or Countermeasures implemented to protect information systems from threats, and minimize the impact of security incidents

A Lost Battle

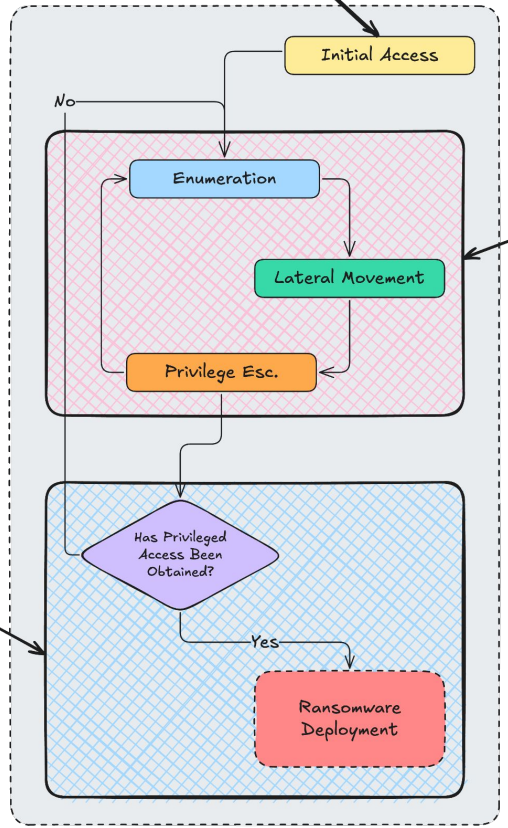
- Ransomware deploys AFTER full compromise
  - Attackers will have privileged access to the environment
    - Domain Admin
    - Root
- <u>Preventive Security Controls can/will be</u>
  - Disabled
  - Modified
  - Bypassed
- *However*, they add friction
  - And making the attackers life harder is always good



DECEPTIQ

ransomware, rarely begins from a privileged foothold

focusing efforts here will almost always provide better results



stopping ransomware here is a lost battle

Winning the War

DECEPTIQ

Preventive

Stop attacks from happening

Modern EDR with ransomware rollback - Auto-reverts encrypted files

DNS filtering - Blocks C2 communication

Vulnerability scanning + patching - Address critical CVEs within 48 hours

Attack Path Modeling - Identifies routes to Domain Admin before attackers do

Restrict admin privileges - Standard users can't install software

Detective

Identify attacks in progress

UEBA/Network Analytics - Detects abnormal user behavior and lateral movement

Deception Technology - Decoy resources (network shares/endpoints) and canary files

Identity Threat Detection - Detect attacks against AD/IDP (Okta, etc.)

Unusual data egress - Catches exfiltration before encryption

Corrective

Respond & recover from attacks

Immutable Backups - WORM storage (cloud) /+ offline rotation

Automated Isolation & Containment - Network/Host segmentation on detection

Cyber Recovery Vault - Isolated environment for clean restoration

Alternative communication channels - When email/phones are down/assume attacker has compromised internals comms.

DECEPTIQ

Preventive - Stop attacks before they happen

Windows

Windows Defender Application Control (WDAC) - Blocks unsigned/unapproved applications

Controlled Folder Access - Prevents ransomware from encrypting protected folders

LAPS - Randomizes local admin passwords

Credential Guard - Protects domain credentials from theft

Group Policy hardening - Disables PowerShell, WMI, RDP where not needed

macOS

Gatekeeper + Notarization - Only runs Apple-verified software

System Integrity Protection (SIP) - Prevents modification of system files

TCC Framework - Controls app access to sensitive data

FileVault 2 - Full disk encryption with central key escrow

MDM Configuration Profiles - Enforces security settings remotely

Linux

SELinux/AppArmor - Enforcing mode prevents unauthorized file access

Immutable files (chattr +i) - Critical configs/binaries can't be modified

AIDE/Tripwire - File integrity monitoring for early detection

Kernel hardening (sysctl) - Disable module loading, restrict ptrace

systemd service hardening

Detective - Identify attacks in progress

- **Deception Technology**
  - Decoy resources (network shares/endpoints)
  - Canary files - triggered on encryption
- **EDR/XDR Telemetry -** (CrowdStrike, SentinelOne, MS Defender)
- System-level logging for forensics
  - Sysmon (Windows)
  - Auditd (Linux)
  - Unified Logs (macOS)
- **Command line monitoring**
  - PowerShell Script Block Logging
  - Bash History
  - Zsh Logging
- **File Integrity Monitoring**
- **Network Traffic Analysis -** (C2/Exfiltration)

Corrective - Respond & recover from attacks

- **Immutable Backups**
  - Survives admin compromise
  - Cloud WORM storage (S3, Azure, GCS)
- **Automated Isolation**
  - EDR instantly quarantines infected endpoints
- **Tested Recovery Procedures** (drills)
- **Alternative Infrastructure**
  - Clean recovery environment separate from production

Evasion

Detecting File System Encryption

Process-level scoring system

1. Each file modification event evaluated against heuristics
2. Score accumulates based on anomaly severity
3. Alert triggered and process terminated **at threshold**

Detection	Feature Description
Header Mismatch	File magic bytes don't match extension (PDF without %PDF header)
Entropy Mismatch	Checking whether the file entropy exceeds expected range for file type
Path History	Tracks delete/create patterns for same file path within process
Ransom Notes	Pattern matching for "YOUR FILES ENCRYPTED" type messages
Honey Files	Decoy files and folders monitored for encryption attempts
Known Extensions	Database of ransomware family signatures (.locky, .cerber)

github.com/elastic/protections-artifacts/blob/main/ransomware/artifact.lua
github.com/rad9800/RansomFS/tree/main/RansomFS

DECEPTIQ

Header Mismatch - *On file write/handle close check if file header (magic bytes) match the file extension.*

How to Bypass

- Can preserve original file headers by:
  - Only encrypting file content after magic bytes
  - Copying original headers back after encryption
- Some file types lack consistent magic bytes
  - For example text/markdown based files (.txt/.md)
- Headers can be valid but content still encrypted
- *Performance impact of reading file headers for every write operation*

DECEPTIQ

Entropy Jump - *An increased entropy of file content is a good indicator of encrypted content.*

How to Bypass

- Partially encrypted files may not:
  - Trigger threshold (e.g. if looking for 50% jump)
  - Raise overall past entropy thresholds
- Pre-compressed/encrypted files already have high entropy
  - Compression tools behave a lot like ransomware 🤔
- Can lower entropy by:
  - Padding encrypted content
  - Partial encryption of content
    - For example every other 4KB
  - Measuring entropy of a file before
    - Encrypt until the new entropy hits a threshold

DECEPTIQ

File Renaming - *Single process renames several files to an unknown extension in a short period of time.*

How to Bypass

- If they maintain known bad file extension dictionary to check against (ASR)
  - Don’t use a known extension - generate on the fly
- Avoid rename operations by:
  - Creating new encrypted files and deleting originals
- Distribute file encryption/renaming across multiple PIDs
- *False positives from legitimate batch rename operations*

DECEPTIQ

Canary Files - *Decoy files and folders monitored for encryption attempts*

How to Bypass

- Target specific directories
- Skip hidden folders and files
- Check if file was modified after created at date 🙃
- Don't encrypt files when found
  - Shuffle (random)
  - Sort by comparable properties

Detection Details

- FindFirstFile behavior on NTFS:
  - Returns files in directory entry table order (roughly alphabetical)
- Force early enumeration using prefixes:
  - $ has lowest ASCII/Unicode value
  - *aa / zz* prefixes (elastic's approach)
- Deploy folders in known starting points for ransomware encryption
  - User/Root Directories
- Insert decoy files into folders
  - *.txt, .doc, .docx*

Windows Specific Evasion

(Windows) Volume Shadow Copy Service

- Enables creation of volume backups
  - Point-in-time copies of data (shadow copies)
- \Device\HarddiskVolume1
  - \Device\HarddiskVolumeShadowCopyN
- Two mechanisms
  - Complete copy (split mirror)
  - Copy-on-Write (differential copy)
- Shadow copy providers
  - Default system provider (leverages CoW)
    - Provided by volsnap.sys and swprv.dll

Tampering VSS (1)

```
vssadmin.exe delete shadows /all /quiet"
WMIC.exe shadowcopy where \"ID='%s'\" delete"
diskshadow.exe delete shadows /all
```

- Living-off-the-Land
- EDRs have telemetry & detection for command lines
- PsSetCreateProcessNotifyRoutineEx
  - Synchronous callback on process creation/exit
- **PCREATE_PROCESS_NOTIFY_ROUTINE_EX**
  - Set **CreateInfo.CreationStatus** to veto process creation
- Bypass by using COM providers
- **IVssSoftwareSnapshotProvider::DeleteSnapshots**
  - Delete snapshot
- **IVssDifferentialSoftwareSnapshotMgmt::ChangeDiffAreaMaximumSize**
  - Resize storage association for shadow copy storage
  - Set to smallest acceptable byte size (1 byte)
  - Causes shadow copies to disappear - deleting them

Tampering VSS (2)

- Reimplement the actual IOCTL's used by these providers/CLI utilities
  - Approach first documented by Fortinet in 2020 (1)
- All implementations at the heart will use the same IOCTLs
- Detection/Prevention requires IRP filtering at kernel level

```
vssadmin delete shadows /for=c: /all
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=1
```

1. Open handle to **\Device\HarddiskVolumeShadowCopyN**

2. Send either:
IOCTL_VOLSNAP_DELETE_SNAPSHOT
IOCTL_VOLSNAP_SET_MAX_DIFF_AREA_SIZE

- Growing number of ransomware families leveraging this.
- Public PoC by @gtworek in 2021 (2)

1 - https://www.fortinet.com/blog/threat-research/stomping-shadow-copies-a-second-look-into-deletion-methods
2 - https://github.com/gtworek/PSBits/tree/master/IOCTL_VOLSNAP_SET_MAX_DIFF_AREA_SIZE

Safe Mode

- EDRs do not run in Safe Mode
- Reboot into Safe Mode to evade detection
- Prevention methods:
  - Block bcdedit command line via
    - **PsSetCreateProcessNotifyRoutineEx**
  - Monitor BCD registry via
    - **CmRegisterCallback**

```
bcdedit /set {current} safeboot minimal shutdown /r /f t 00
```

Early Boot - Security Considerations

if we run before EDRs/other services...

could we not just encrypt files here? 🧐

- *BootExecute* mechanism:
  - Runs unsigned executable before Win32 initialization
    - Executes before EDR loads (services/drivers)
  - Enables disabling of EDR services during boot
  - Enables manipulation of Shadow Copy volumes
- Vendor detection gaps:
  - Only monitor BootExecute registry key
  - Miss other boot-time registry keys (new keys added in Windows 11)
  - *BootExecuteNoPnpSync* , *SetupExecute* , *PlatformExecute* , e.g.

Time to Apply These Techniques!

- Windows Evasion
- Different Encryption Strategies
  - Every other page
  - First X Bytes
- Different File Searching Strategies
  - Shuffling
  - Intelligent Sorting

DECEPTIQ

Thank You. Any Questions?