JAFF RANSOMWARE INFECTION

ARUN M MYAGERI
ADITHYA BM
AMIT KUMAR

Objective

- Analyze the Jaff ransomware
- Focus on:
  - Infection vectors
  - Malicious document structure
  - Behavioral indicators
  - No execution of ransomware binary

Collected Samples

- Retrieved from Malware-Traffic-Analysis.net
- Data includes:
  - 📂 PCAP: 2017-06-01-Jaff-ransomware-infection-traffic.pcap
  - 📊 CSV: Malspam tracker file
  - 📧 EML emails with:
    - Spoofed senders
    - PDF attachments
    - Embedded DOC files
- 🔒 Hashes used to verify sample integrity (SHA256)

PCAP & Network Flow Analysis

- 🧰 **Tool used:** Wireshark
- 📊 **Tracked:**
  - Infection path from PDF to DOC to binary download
  - Connections to suspicious IPs
  - TCP RST packets detected — signs of:
  - Network interruption
  - Possibly anti-sandboxing techniques or aborted malicious connections

Malspam Vector

- Emails impersonated trusted contacts
- Example sender: Marcos.7077@[victim-domain]
- Attached PDF files like 77586054.pdf
- PDFs embedded Word documents with macros
  - 🛠️ **Purpose:** Trick users into opening malicious documents and enabling macros — a common ransomware delivery method.

Tool – oletools + oleid

🧰 **What it does:**

- **oletools:** Extracts and analyzes VBA macros in Office documents.
- **oleid:** Gives high-level OLE document metadata (suspicious features, macros, embedded objects).

🔍 **Our Results:**

Found AutoOpen and Document_Open macros (trigger on open)

- Detected functions like:
  - CreateObject("WScript.Shell")
  - Shell("cmd /c powershell")

- Showed obfuscated strings and downloader code

📌 **Conclusion: Word docs acted as droppers, initiating the next stage of infection.**

URLs in Macros

📡 Extracted from obfuscated VBA:

- dsopro[.]com/7rvmnb
- katoconsulting[.]ro/7rvmnb
- fabriquekorea[.]com/7rvmnb
- tasfirin-ustasi[.]net/7rvmnb

🛠️ Indicators of Compromise (IOCs)

- These were compromised websites used to host the ransomware binary.
- These URLs were hidden using string obfuscation and only assembled during macro execution.

PDF Analysis with pdf-parser.py

🧰 What it does:

- Analyzes PDF structure for objects, JavaScript, and embedded files.

🔍 Our Results:

- Found embedded file: XKDQK1N.doc
- Detected JavaScript:

```
obj 24 0
 Type: /Catalog
 Referencing: 20 0 R, 23 0 R, 18 0 R

  <<
    /Type /Catalog
    /Pages 20 0 R
    /Names 23 0 R
    /OpenAction
      <<
        /S /JavaScript
        /JS 18 0 R
      >>
  >>
```

🧨 Danger: This script auto-extracts and launches the embedded malicious Word doc upon opening the PDF in a vulnerable reader.

Word Doc Macro Behavior

🧰 Tools used:

- **olevba from oletools**
- **mraptor: Quick macro risk scanner**

🔍 Findings:

- **Macros included:**
  - **PowerShell command execution**
  - **Payload download via XML,HTTP**
  - **Launch of executable using rundll32.exe (Living-off-the-land Binary)**

📌 **Implication: These macros are heavily obfuscated, but goal is to download & run the Jaff payload.**

Executable Analysis – bruhadson8.exe

- 🧰 **Tools used:**
- **binwalk for entropy**
- **Strings & PE header analysis**
- 📦 **File Type: Windows PE 32-bit GUI binary**
- 🔒 **Behavior:**
  - **Contains:**
    - **Ransom note with .onion link: rktazuzi7hbln7sy.onion**
    - **Hardcoded Decrypt ID (placeholder: 0123456789)**
    - **High entropy regions → indicates packed/encrypted data**
    - **Calls to encryption-related WinAPI functions:**
    - **CreateFileW, WriteFile, FlushFileBuffers**

API Call Behavior (Static Indicators)

- 🧬 **Analyzed DLL imports:**
- **KERNEL32.dll – File I/O, memory handling**
- **ADVAPI32.dll – Permission & ACL modifications**
- **USER32.dll – GUI (possibly decoys)**
- **NTDSAPI.dll – Rare for ransomware; indicates Active Directory queries**
- ⚙️ **Suspicion:**
- **Possible lateral movement or domain awareness**
- **Fake GUI elements as distractions**

Entropy Analysis (binwalk -E)

- 📊 Purpose: Measures randomness to detect packing/encryption
  - 📍 Observations:
    - Entropy > 0.96 in many regions — typical for compressed/encrypted payloads
    - Clear drop-offs around:
      - 0x11800
      - 0x2D000
    - Suggests boundary between loader code and hidden payload
  - 📌 Inference: bruhadson8.exe is likely a packed dropper that decrypts and runs the real ransomware at runtime.

VirusTotal Scans

🛡️ **Samples flagged as malicious:**

File	Detection Count
`bruhadson8.exe`	✅ High (Multiple AVs)
`FXCHG1Y.doc`	✅ Macro trojan
`35418461.pdf`	✅ Embedded malware
`fabriquekorea.com`	✅ Malicious domain

📌 **Confirms our manual analysis findings using real-world threat intel.**

Final Observations

- **Key Takeaways:**
  - **Jaff uses multi-stage malspam campaign**
  - **Triggers:**
    - **Email → PDF → DOC → Macro → Download → Execution**
  - **Employs:**
  - **Obfuscation**
  - **LOLBins**
- **High-entropy packing**
- **Exhibits ransomware hallmarks, with network & AD awareness**
- **🧩 No execution was done – entire chain analyzed statically and in sandbox-safe methods.**

References

- 🦠 Jaff Ransomware Campaign Info
  - Malware-Traffic-Analysis.net (sample source)
  - 🧬 Analyzed DLL imports:
  - KERNEL32.dll – File I/O, memory handling
  - ADVAPI32.dll – Permission & ACL modificatio

Thank You!