Foremost-NG: An Open-Source Toolkit for Advanced File Carving and Analysis

Cristian Souza

DFIR Specialist, Ph.D Student
(Kaspersky & IME-USP)

Agenda

Introduction

In the rapidly evolving landscape of cybersecurity and digital investigations, the ability to recover and analyze deleted or hidden artifacts can make the difference between a successful incident response and an unresolvable case.

Introduction

- Traditional file-carving tools, while indispensable, often struggle with newer file formats, complex binary structures, and the diversity of data types encountered on modern endpoints.
- The original `Foremost` project laid important groundwork, pioneering a signature-based carving engine that could recognize common file headers and reconstruct deleted files from raw disk images.
- However, as new operating systems, executable formats, and log containers emerged, extending `Foremost` proved increasingly needed.

Introduction

- To the best of our knowledge, the last official release of `Foremost` (version 1.5.7) dates back to 2009 [1].
- Since then, changes in the GNU C Library (`glibc`) have introduced compatibility issues that prevent the original source code from compiling on modern operating systems[2].
- `Foremost-NG` was conceived to address these limitations by refactoring and modernizing the tool's architecture from the ground up.

[1] `http://foremost.sourceforge.net/pkg/foremost-1.5.7.tar.gz`
[2] `https://github.com/korczis/foremost/issues/8`

Introduction

- Beyond refactoring, `Foremost-NG` introduces first-class support for file types that until now were difficult to carve.
- Windows EVTX event logs (rich sources of forensic evidence) are now parsed natively, allowing investigators to extract and interpret event records directly from a raw partition or memory dump.
- Similarly, script files (shell scripts, Python, PowerShell) are identified by their characteristic shebangs and Unicode signatures.

Introduction

- On macOS and Linux systems, `Foremost-NG` detects Mach-O and ELF headers, respectively, reconstructing binary executables even when fragmented across disk sectors.
- Arguably the most notable new feature is the integration with VirusTotal's public API[3].
- This permits `Foremost-NG` to perform an on-demand lookup based on the cryptographic hashes (SHA-256, MD5) it computes for each recovered artifact.

[3] `https://docs.virustotal.com/reference/overview`

Introduction

- Within a few seconds of carving a file, the user gains insight into whether that artifact is known to threat actors, what tags or AV detections exist, and whether it merits deeper reverse engineering.
- This real-time enrichment transforms `Foremost-NG` from a simple file recovery utility into an intelligence-driven forensics assistant, guiding analysts toward high-priority items and reducing time wasted on benign fragments.

Foremost-NG workflow

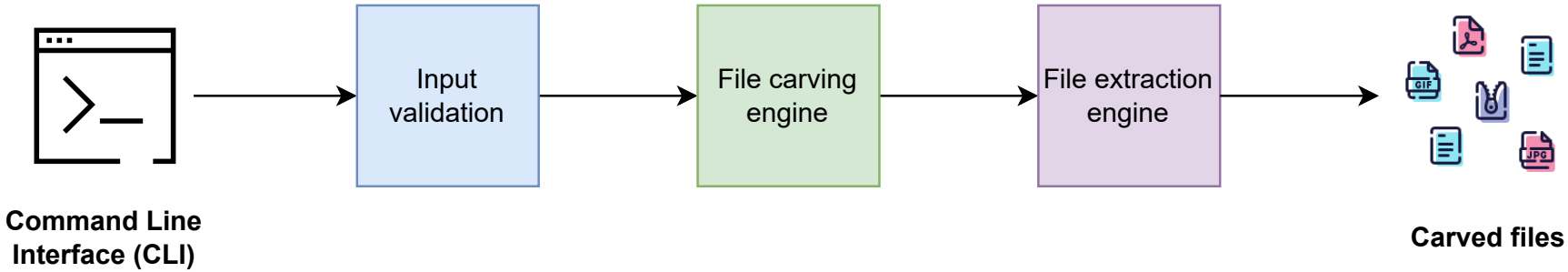

Figure: Foremost-NG workflow

Architecture and Implementation

- `Foremost-NG` retains the lightweight C implementation of the original tool but rearranges core components and provides integration with VirusTotal for threat intelligence lookups.
- At launch, a central dispatcher initializes a registry of available parsers embedded in the source code.

Architecture and Implementation

Each parser contains a set of characteristics for identifying valid files:

- A unique format identifier (e.g., `"EVTX"`, `"MACHO"`, `"ELF"`).
- Byte signatures or heuristic checks for header detection.
- A callback to validate extracted headers/trailers.
- A recovery function that writes out the reconstructed file and its metadata.

Architecture and Implementation

- The VirusTotal lookup in `Foremost-NG` is implemented as a lightweight, self-contained module.
- When a file has been successfully carved, `Foremost-NG` computes its cryptographic checksum (either SHA-1 or SHA-256) using OpenSSL's EVP interface.
- Next, it retrieves the user's API key from the `VT_API_KEY` environment variable.

Architecture and Implementation

- With both checksum and key in hand, the tool formats an HTTP GET request to `https://www.virustotal.com/api/v3/files/hash`, setting the `x-apikey` header and requesting JSON in return.
- Once the HTTP transaction completes, the code searches for the `last_analysis_stats` object in the returned JSON.

File format support

- JPEG (JFIF/EXIF).
- PNG (IHDR/IEND).
- BMP (Bitmap).
- GIF (GIF87a/GIF89a).
- RIFF Containers (AVI, WAVE).
- WMV (Windows Media Video).

File format support

- OLE Containers (DOC, XLS, PPT).
- ZIP (OOXML/OpenOffice).
- RAR Archives.
- PDF (Linearized or Non-Linear).
- C/C++ Source Files (CPP).

File format support

- HTML (HTM).
- MPEG-1 Video (MPG).
- MP4 Video.
- EXE/DLL (PE Executables).
- Windows Registry Hives (REGF).

File format support

- Windows Event Logs (EVTX).
- ELF (Linux/Unix Binaries).
- Mach-O (macOS/iOS Binaries).
- Script Files (Bash, Python, PowerShell).

Installation

To install `Foremost-NG`, clone the GitHub repository and compile with a standard GCC toolchain. Example on a Unix-like system:

```
git clone https://github.com/cristianzsh/
foremost-ng.git

cd foremost-ng
make
sudo make install
```

Basic usage

Users must export an environment variable called `VT_API_KEY` and include a valid VirusTotal API key if they wish to enable the VT lookup:

```
export VT_API_KEY=<YOUR_VT_API_KEY>
```

Documentation:

```
man foremost-ng
```

Basic usage

Carving a disk image:

```
foremost-ng -i disk.img -o /temp/output/carved
```

Restricting a subset of formats:

```
foremost-ng -t evtx,elf,exe -i disk.img
-o /temp/output/carved
```

Querying VirusTotal:

```
foremost-ng -x -t elf -i memory.dump -o
./carved-files
```

Carving results

```
$ ./foremost-ng -v -x evidence2.dd
foremost-ng version 1.5.7
Audit File

foremost-ng started at Mon Jun  9 23:33:41 2025
Invocation: ./foremost-ng -v -x evidence2.dd
Output directory: /tmp/foremost-ng/src/output
Configuration file: /tmp/foremost-ng/src/foremost.conf
[INFO] Processing: evidence2.dd
---------------------------------------------------------------
File: evidence2.dd
Start: Mon Jun  9 23:33:41 2025
Length: 2 GB (2845109643 bytes)

  ID  Name (bs=512)              Size       Offset  Comment                VT
   0: 00004352.elf              57 KB      2228224                        Clean (0)
   1: 00004820.elf              57 KB      2467840                        Clean (0)
   2: 00005108.elf               1 KB      2615296                        Clean (0)
   3: 00005492.script            3 KB      2811904                        Clean (0)
   4: 00399884.exe             600 KB    204740608  01/01/1970 00:00:00   Malicious (2)
   5: 00405060.exe              52 KB    207390720  01/01/1970 00:00:00    Clean (0)
   6: 00414932.exe               2 MB    212445184  01/01/2015 00:00:00    Clean (0)
   7: 00428728.exe               2 MB    219508736  01/01/2015 00:00:00    Clean (0)
   8: 00442520.exe             536 KB    226570240  01/01/1970 00:00:00    Clean (0)
   9: 00447356.exe             600 KB    229046272  01/01/1970 00:00:00    Malicious (2)
  10: 01786101.zip             470 KB    914484119                         Clean (0)
  11: 01787042.zip             350 KB    914965615                         Clean (0)
```

Figure: Foremost-NG output

Conclusion

- `Foremost-NG` modernizes the original carving engine.
- It extends support beyond legacy formats to include Windows EVTX logs, script files, Mach-O binaries, and ELF executables.
- Built-in VirusTotal integration enriches every carved file with real-time threat intelligence.
- As future work, we plan to extend the supported file types and improve the overall tool's performance.

kaspersky

Foremost-NG: An Open-Source Toolkit for Advanced File Carving and Analysis

Cristian Souza

DFIR Specialist, Ph.D Student
(Kaspersky & IME-USP)