Zip Bomb Incident Response Documentation

Aligned with NIST SP 800-61

Note

This report documents the investigation of a simulated Zip Bomb attack executed within the same isolated forensic environment previously used for the ransomware analysis. The "before-attack" baseline is taken from the previously acquired CLEAN memory snapshots (BEFORE_RANSOM.*).
All post-incident evidence comes from the memory acquisition and analysis files you provided for the Zip Bomb scenario.

1. Incident Summary

A simulated Zip Bomb attack was executed using the file **42.zip**. When extracted using WinRAR inside a Windows 7 SP1 virtual machine, the archive caused rapid disk consumption, system instability, and eventual system hang due to resource exhaustion.
A full memory dump was later acquired through VirtualBox's host-level functionality because DumpIt.exe could not run after the disk became full. The memory dump was then analyzed using Volatility 3 on an Ubuntu workstation.

2. Preparation (NIST Phase 1)

2.1 Laboratory Environment

The attack was executed inside the same isolated virtual environment (Internal Network mode) to ensure complete containment and prevent accidental spread or system-wide DoS.

2.2 Baseline Establishment

Because the environment was shared with a prior ransomware investigation, the previously collected baseline files were reused:

- BEFORE_PSLIST
- BEFORE_CMDLINE
- BEFORE_HIVELIST
- BEFORE_TIMELINE

These were used as clean-state references for comparison.

2.3 Tooling Readiness

The forensic workstation was equipped with:

- Volatility 3
- Strings utility
- Host-level VirtualBox memory dumping capability (used due to disk exhaustion)
- DumpIt.exe (not usable after disk became full)

This ensured readiness for evidence collection even during heavy system instability.

3. Detection and Analysis (NIST Phase 2)

3.1 Detection

The attack was detected through the following indicators:

- The operating system became unresponsive (system hang).
- WinRAR stopped responding and eventually terminated.
- A system error message indicated “Insufficient Disk Space.”
- The storage capacity of the virtual machine increased extremely rapidly.

These symptoms are consistent with a Zip Bomb extraction.

3.2 Memory Forensic Findings

3.2.1 Absence of WinRAR Process in RAM

The process list extracted from the memory does *not* show WinRAR.exe, indicating that:

1. WinRAR either crashed, or
2. Terminated unexpectedly due to resource exhaustion.

From **AFTER_ZIP_PSLIST**:

```
(No entry for WinRAR.exe)
```

(Source: AFTER_ZIP_PSLIST — confirms abnormal termination)

REPORT_AFTER_ZIP_PSLIST

3.2.2 Strings Analysis Proves WinRAR Activity Before the Crash

While WinRAR.exe does not appear in the live process list, the **strings** extracted from the raw memory dump clearly prove its prior execution.

From **REPORT_FINAL_ZIP_PROOF.txt**:

```
WinRAR.exe
C:\Program Files\WinRAR\WinRAR.exe
WinRAR archiver
Roshal.WinRAR.WinRAR
WinRAR SFX module
```

(Source: REPORT_FINAL_ZIP_PROOF.txt — confirms WinRAR was fully loaded in memory prior to the crash)

REPORT_FINAL_ZIP_PROOF

This indicates that WinRAR was active and allocated in RAM at the time of system failure, even though it was not present as a running process in the final process list.

3.2.3 Execution Environment Confirmed

The process list shows typical system processes, explorer.exe, VBoxTray.exe, and finally DumpIt.exe:

From **AFTER_ZIP_PSLIST**:

```
2808 DumpIt.exe  "C:\Tools\Comae-Toolkit-v20230117\x64\DumpIt.exe"
```

REPORT_AFTER_ZIP_PSLIST

This confirms that memory acquisition happened successfully after system instability.

3.2.4 Registry Hive Consistency

A comparison of the registry mappings before and after the attack shows no persistence mechanisms and no registry-level modifications:

From **AFTER_ZIP_HIVELIST**:

```
\REGISTRY\MACHINE\SYSTEM
\REGISTRY\MACHINE\SOFTWARE
\??\C:\Users\Abo-Ali\ntuser.dat
```

REPORT_AFTER_ZIP_HIVELIST

This matches the baseline registry structure, proving that Zip Bomb attacks impact *availability only* and do not introduce persistence.

3.2.5 Impact Assessment

The attack targeted **Availability**, one of the pillars of the CIA Triad.
Resource exhaustion resulted in:

- Disk saturation
- System-wide unresponsiveness
- Forced termination of applications
- Failure of certain forensic tools (DumpIt.exe)

4. Containment, Eradication, and Recovery (NIST Phase 3)

4.1 Containment

Because the disk was fully consumed, DumpIt.exe could not run. Therefore:

- A full memory snapshot was captured using VirtualBox's host-level memory dump functionality.
- This method ensured the system's frozen state was captured accurately.
- The VM was powered off after dumping to prevent further disk damage.

4.2 Eradication

Root Cause

The root cause was identified as extraction of **42.zip**, a known Zip Bomb.

Removal Actions

Due to lack of a clean snapshot, the system required:

- Manual deletion of the extracted Zip Bomb segments from drive C:
- Removal performed via command-line interface from the host-level tools

- Immediate recovery of disk space following deletion

4.3 Recovery

- The system returned to a semi-operational state after disk cleanup.
- All volatile evidence was preserved before any cleanup took place.
- The extracted memory dump and forensic analysis files were transferred to the forensic workstation for long-term retention.

5. Post-Incident Activity (NIST Phase 4)

5.1 Lessons Learned

- Zip Bombs can completely paralyze a system without executing code.
- Traditional process monitoring may fail during resource exhaustion; however, raw-memory analysis via strings can still reveal important artifacts.
- Snapshots should always be created before conducting experiments involving large or compressed files.

5.2 Recommendations

- Implement endpoint controls that restrict decompression of archives with extremely high compression ratios.
- Apply disk quotas to prevent a single process from exhausting storage.
- Use antivirus or sandboxing tools capable of detecting Zip Bombs before extraction.
- Automate regular system snapshots to enable quick rollback.