Karlsruhe Institute of Technology

SSC4 Debriefing — Forensics

Tobias Dussa • EGI Technical Forum, September 17, Amsterdam

COMPUTER EMERGENCY RESPONSE TEAM

Introduction & Overview

This talk addresses the following questions:

- Why forensic analysis?
- Where and how to gather evidence?
- How to analyze evidence data?

It does *not* address:

- How to contain damage?
- What to communicate when to whom?
- How to recover from an incident?

What Is Forensic Analysis Good For?

To assess and answer several important questions about an incident:

- Where did the attacker come from?
- How was access gained?
- What damage was done?
- What other machines were affected?
- . . . and many more related questions.

Data Sources for Forensic Investigations

- Broad classes of data sources:
  1. Highly volatile (e. g., CPU registers),
  2. Volatile (e. g., RAM),
  3. Static (e. g., hard drives), and
  4. Highly static (e. g., archive tapes).
- More volatile evidence must be gathered and preserved first, if possible.
- Obviously, not all classes available or applicable in every instance.

Find the Right Machine

Usually, this is the first thing to do.

1. Collect all relevant network-related data:
   - NAT data,
   - proxy data,
   - netflow data,
   - and so on.

   No problem if there are log files, interesting if not (live NAT tables etc.).
2. Correlate data to find your suspect host, if any.

So We Have a Suspect . . .

. . . or at least a suspect machine. Now what to do?

1. Gather general information and evidence:
   - Running processes,
   - open network connections,
   - who is logged on,
   - who was logged on,
   - mounted devices
   - and their mountpoints,
   - etc.
2. Look if there is anything suspect.

Freeze!

What to do with your suspect (process):

1. Stop the process (do *not* terminate it!).
2. Collect and secure:
   - the binary being executed,
   - its core memory,
   - its shared memory regions, if any,
   - its file handles,
   - other volatile data.

Now That We Have More Time

Finally, collect less volatile stuff:

- If possible and sensible, power off the machine and grab the hard drive.
- If not possible or sensible, at the very least collect the following stuff:
  - All related log files,
  - any files involved in the incident,
  - actually, if possible, the entire file system.

After Compiling, Interpretation!

Take a close look at the collected data. Some pointers:

- Inspect suspect executables (with `strings`, `hexdump`, `gdb`, `rec`, IDAPro, . . .).
- Look at core dumps (using `gdb`).
- Grep through log files and the like.
- Check files' MD5 sums against the known-good list.
- Perform further filesystem analysis, for instance with `autopsy` or `rkhunter`.
- If necessary, iterate.

SSC4: Situation

- Known facts for Security Service Challenge 4:
  - IP addresses 192.108.46.248 and 195.140.243.2 are evil.
  - (End of list!)
- Unknown: Everything else, particularly
  - Are the bad IPs involved with our systems?
  - If so, how?
  - And what happened, if anything?

SSC4: Initial Investigation

- First step: Find out whether the IPs in question have shown up at our site.
- Sifting through the appropriate logs yields a machine connected with the suspect IPs (boring).
- Surprisingly, we have a winner!
- (Watch out for timestamp time zone!)

SSC4: On the Machine

Culprit process was quickly identified (no stealth measures).

- Job was submitted via Panda.
- Panda logs show
  - what DN was used to submit the actual job and
  - what host the actual job was submitted from.
- Next step, obviously: Dump all the information we can get.

SSC4: Closer Look

- Running the binary through `strings` reveals some fishy strings in the binary:
  `JOIN, NICK, PONG, PRIVMSG, USER`
- Disassembling yields information about:
  - Communication and
  - other activity.
- Inspecting the core dump gives actual ID strings used in communication.

SSC4: RE Results

After reverse-engineering, this was known:

- Binary was an IRC bot (communication endpoints and parameters known),
- (tried to) install
  - at job and
  - cron job

  to become persistent, and
- (tried to) transfer `/etc/passwd` out to drop site, but
- no root exploit used and no root kit installed.

Common Pitfalls

Things to watch out for when doing forensics:

- Modifying evidence while collecting (e. g., file access times).
- Dropping volatile evidence (e. g., memory content).
- Failing to document actions properly (timestamps!).
- **IMPORTANT:** If the incident looks like it will involve legal action, **stop everything you are doing** and **call the police**!

Good Approaches

Common sense and good practices:

- Prepare a checklist.
- Strictly separate evidence acquisition and evaluation.
- Gather evidence, then produce a working copy of the evidence locker, then work on the working copy only.
- Go out of your way to ensure you work in read-only mode whenever possible, even on the working copy.
- And, most importantly, if you are unsure what to do, talk to somebody who has a better chance of knowing (e. g., EGI CSIRT).

Th-Th-Th-Th-Th-. . . That’s All, Folks!

Any questions?

Th-Th-Th-Th-Th-. . . That’s All, Folks!

Thank you for your attention!