Malware Unpacking

Stefano Maccaglia, RSA Incident Response Team

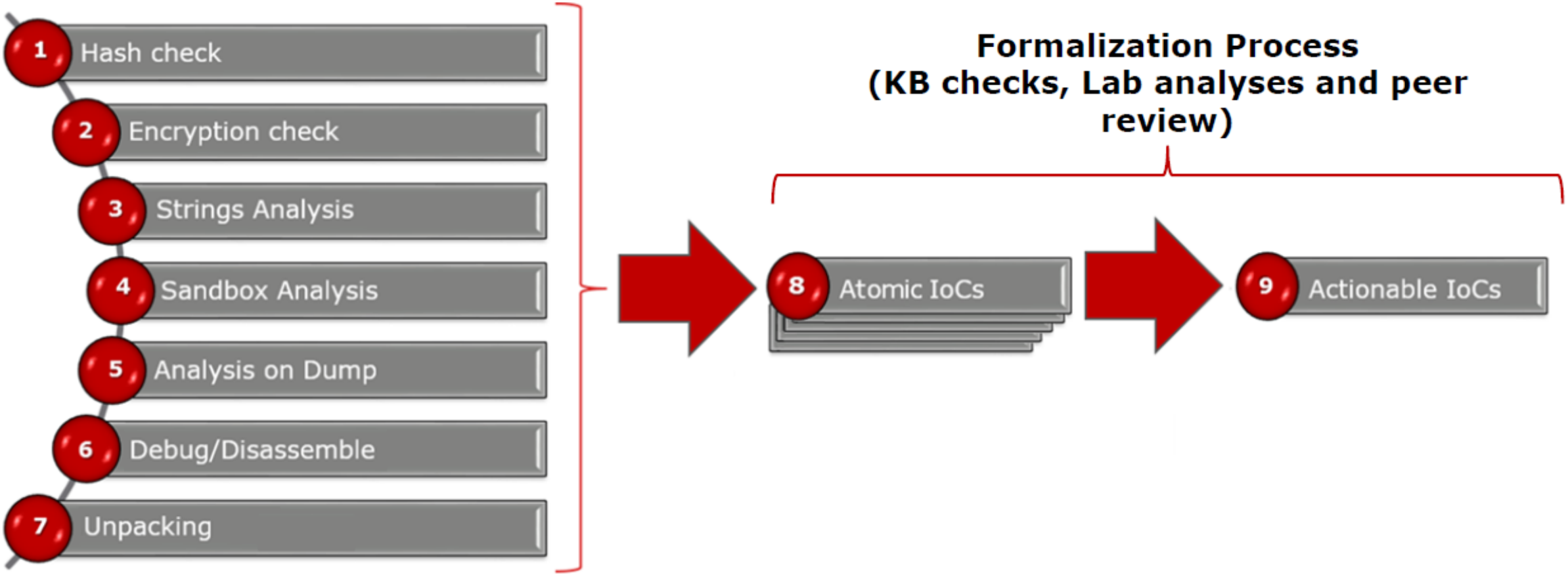
RSA

INCIDENT RESPONSE AND ENCRYPTED MALWARE

- In the Incident Response field the malware analysis process can be extremely straightforward:
  - Check VT for simple descriptors (Hashes, filenames, strings, peculiar hex values),
  - Test the malware in a sandbox,
  - Create a signature (such as Yara, ClamAV, ECAT IIOC, etc…),
  - Run the massive triage process.
- Often, running strings.exe on a malware executable is enough to collect IOCs such as IP addresses, file names, registry keys, and other valuable details out of it.
- But what if we stumble upon an encrypted file? What if the file is also protected with Antis?
- The frequency of these situations are not extremely common, but still there are APT groups that give the analyst some headaches… not to mention cybercriminals…
- Let's see what we do…

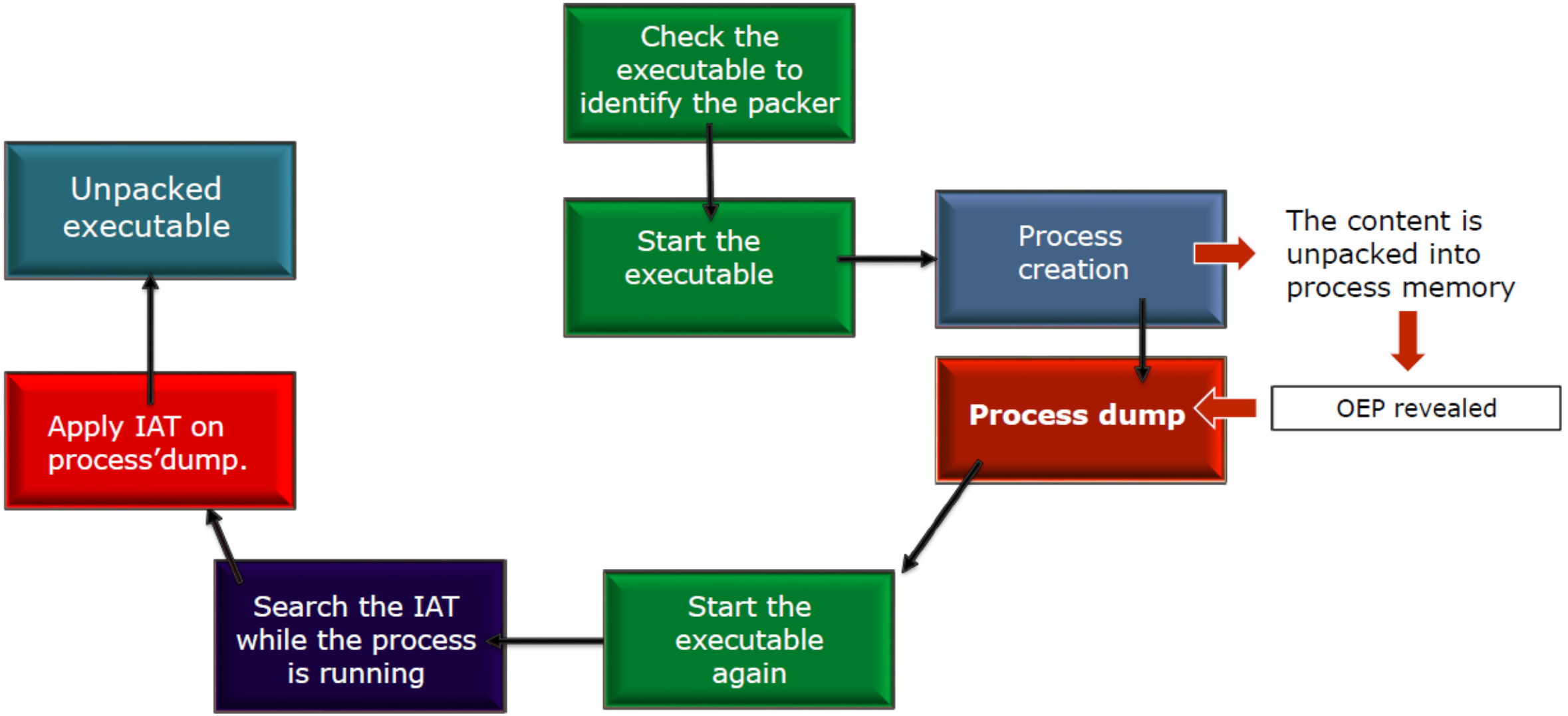
RSA

MALWARE ANALYSIS PROCESS TO BUILD AIOCS



RSA

TRADITIONAL UNPACKING PROCESS

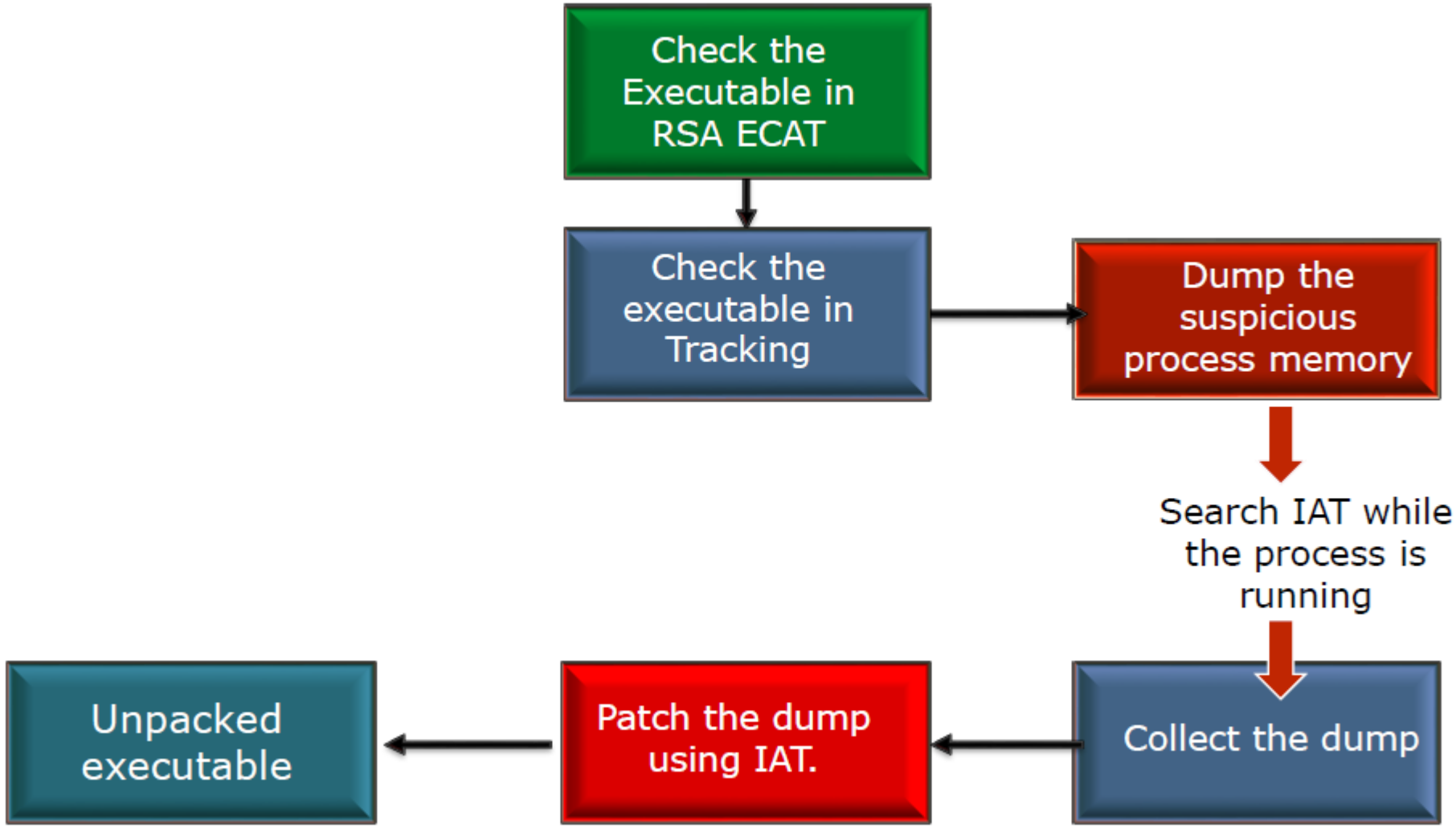


RSA

PACKERS AND MALWARE

- The packer move the executable to data sections of the packed file.
- PE header & section header are no more descriptive since data is compressed by the packer stub function.
- Unpacking a malware is an art. But we don’t need to look at the most fancy ways to unpack. The most precious resource we have is the time... So we need to unpack quickly and reliably.
- This makes the task very challenging because the analyst should use all his capabilities to overcome the different types of mechanisms the packers can use to collect back, at the end of the process a very similar executable to original PE.
- In IR the malware analysis require these skills because often we face such packed malware and we must overcome their tricks in order to understand them and be able to catch them in an environment.

UNPACKING WITH RSA NETWITNESS ENDPOINT



PACKERS AND MALWARE

- The procedures presented are just for generic unpacking of malware encrypted or obfuscated with basic packers such as Aspack, Armadillo or UPX.
- For more complex situations the procedure can be far more complicated, but in the end the approach is still the same, force the malcode to drop a portion or its entire code in the memory (hopefully in clear text) and go for a dump to be patched.
- An interesting procedure I want to discuss is the one involving Unpacking Dynamically Allocated code because that allow the analyst to dump the interesting code directly into a new segment of the file.

UNPACKING DYNAMICALLY ALLOCATED CODE