IREC Documentation

Release 1.8.0

Binalyze

Jun 12, 2019

Contents

CHAPTER 1

What is IREC?

IREC is an all-in-one evidence collector which makes it possible to acquire critical evidences from a live system in the blink of an eye. No need to lose your precious time for looking for the needle in the proverbial haystack anymore. IREC minimizes incident response time to minutes and increase the effectiveness by presenting you all the needed clues. Additionally, by collecting and preserving evidences, it meets the needs of cyber security and digital forensic at the same time. Imagine an automated IR software that collects and presents all critical data for you. That's IREC and that's all you need for the fastest IR ever.

CHAPTER 2

Contents

2.1 Running IREC from command line

This section describes how to invoke IREC from commandline or remotely using tools such as PsExec.

Note: Command line support is **only available** in TACTICAL Edition.

By default, IREC starts in GUI mode unless a command line option starting with a dash (-) is provided. Command line options come in two flavors: A long form such as –profile and a short form: -p. See the example below for providing evidence acquisition profile

```
IREC.exe --profile full
IREC.exe -p full
```

2.1.1 Command Line Options

--help / -h
: Displays the URL for the latest documentation.

--no-wait / -nw
: By default, IREC will wait for a key press once the requested operation completes. Providing this option will make it terminate immediately without waiting for a key press.

  Note: You should always provide this option when running IREC remotely using tools such as PsExec.

  Examples:

  ```
  IREC.exe --profile full --no-wait
  ```

--license <Key> / -l <Key>
Provides the license key to use for activating IREC. If not provided, IREC will try to read the Key from License section of IREC.Settings.ini file.

Examples:

```
IREC.exe --license AAA-BBB-CCC-DDD

IREC.exe -l AAA-BBB-CCC-DDD
```

--app-dir <FolderPath> / -ad <FolderPath>
By default, IREC uses the directory is executed from as its Application Directory. This option tells IREC to use the provided directory for creating/reading/writing the files and folders listed below:

- `IREC.Settings.ini`: All application settings are saved into this file.
- `IREC.Log.txt`: All application logs.
- `IREC.Error.txt`: Only created when an exception occurs.
- `IREC.Rulesets`: Folder for Custom Content Profiles (.ccp files).
- `IREC.Profiles`: Folder for YARA scripts (.yar files).
- `IREC.Bin`: Created by IREC Dongle Edition (a SFX archive) for extracting its contents.

Note: By default, provided folder path will be used for saving case output as well. You can override this behaviour by providing either **case-dir** or **output-dir** options.

Note: This option is required for performing Triage using a YARA Ruleset or collecting custom content when IREC is executed remotely via PsExec!

Examples:

```
IREC.exe --app-dir "\\MACHINE\IREC-DIR"

IREC.exe -ad "\\MACHINE\IREC-DIR" --triage-ruleset MyYaraRules --triage-memory #␣
↪Uses \\MACHINE\IREC-DIR\IREC.Rulesets\MyYaraRules-memory.yar file

IREC.exe -ad "\\MACHINE\IREC-DIR" --custom-content "Hacked Server"  # Uses␣
↪\\MACHINE\IREC-DIR\IREC.Profiles\Hacked Server.cpp file
```

--profile <Profile> / -p <Profile>
Selects the Evidence Collection Profile. Can be one of the following:

- `full`: Collects all evidence types.
- `custom`: Each evidence type should be provided separately from command line. See *Evidence Types* for more information.
- `memory`: Collects RAM and PageFile only.
- `default`: Collects only default enabled evidence types.

Note: Default selected profile is "Custom" which requires each evidence item to be separately provided from command line. See *Evidence Types* for more information.

Examples:

```
IREC.exe --profile full

IREC.exe -p custom -ram -hbr -pf -evt -evtx
```

--output-dir <DirPath> / -od <DirPath>
Sets the directory in which case directory will be created in. Case directory is in format TIMESTAMP-MACHINENAME. If you want to provide an absolute path, use —case-dir option instead. Trailing backslash is ignored.

Examples:

```
IREC.exe --output-dir C:\Cases\Root

IREC.exe -od "C:\Case Folder\Root"
```

--case-dir <CasePath> / -cd <CasePath>
Sets the absolute path of case directory. Provided path will be used as is without creating any folders inside. If you want IREC to automatically create a directory for each case, use —output-dir option instead. Trailing backslash is ignored.

Examples:

```
IREC.exe --case-dir "C:\Cases\Final"

IREC.exe -cd "C:\Cases\Final"
```

--custom-content <ProfileName> / -cc <ProfileName>
Provides custom content collection profile name. Custom Content profiles can be found in IREC.Profiles folder in IREC.ProfileName.ccp format. IREC expects only the ProfileName portion in this command line option.

Examples:

```
IREC.exe --custom-content "Hacked Server"

IREC.exe -cc SomeProfile
```

--triage-ruleset <RuleSetName> / -tr <RuleSetName>
Selects the provided rule set for performing Triage with YARA. If not provided, Default ruleset will be used in case memory or filesystem triage is enabled with either —triage-memory or —triage-filesystem options.

Examples:

```
IREC.exe --triage-ruleset "New Set" --triage-memory

IREC.exe -tr Default -tm
```

--triage-memory / -tm
Enables memory triage. In case —triage-ruleset is not provided, Default ruleset will be used.

Examples:

```
IREC.exe --triage-memory

IREC.exe -tm
```

--triage-filesystem / -tf
Enables filesystem triage. In case —triage-ruleset is not provided, Default ruleset will be used.

Examples:

```
IREC.exe --triage-filesystem

IREC.exe -tf
```

2.1.2 Evidence Types

You can use the command line options for enabling each evidence type separately when Custom evidence collection profile is selected by providing –profile custom option.

Name	Long Form	Short Form	Default
Clipboard	–Clipboard	-clp	TRUE
Crash Dump Info	–CrashDumpInfo	-cdi	TRUE
Recycle Bin Info	–RecycleBinInfo	-rbi	TRUE
Restore Point Info	–RestorePointInfo	-rpi	TRUE
Driver Info	–DriverInfo	-dri	TRUE
Process Info	–ProcessInfo	-pri	TRUE
Screenshots	–Screenshots	-scr	TRUE
AntiVirus Info	–AVInfo	-avi	TRUE
DNS Server	–DNSServer	-dnss	TRUE
Proxy Info	–ProxyInfo	-prxy	TRUE
Volume Info	–VolumeInfo	-voli	TRUE
MBR	–MBR	-mbr	FALSE
RAM	–RAM	-ram	TRUE
PageFile	–PageFile	-pgf	TRUE
SwapFile	–SwapFile	-swp	FALSE
Hibernation File	–HibernationFile	-hbr	FALSE
Chrome History	–ChromeHistory	-chst	TRUE
Firefox History	–FirefoxHistory	-fhst	TRUE
IE History	–InternetExplorerHistory	-ihst	TRUE
Edge History	–EdgeHistory	-ehst	TRUE
MFT as CSV	–MFTCsv	-mftcsv	TRUE
MFT as Binary	–MFTBin	-mft	FALSE
MFT Mirror	–MFTMirr	-mftmir	FALSE
Ntfs LogFile	–NtfsLogFile	-ntfslog	TRUE
Ntfs UsnJournal	–NtfsUsnJournal	-usnjrn	TRUE
Registry Hives	–Hives	-hiv	TRUE
Registry Hives (Windows.Old)	–HivesOld	-hivold	TRUE
DNS Cache	–DNSCache	-dnsc	TRUE
TCP Table	–TCPTable	-tcpt	TRUE
UDP Table	–UDPTable	-udpt	TRUE
ARP Table	–ARPTable	-arpt	TRUE
IPv4 Routes	–IPv4Routes	-ipv4	TRUE
Network Adapters	–NetworkAdapters	-netadp	TRUE
Network Shares	–NetworkShares	-netshr	TRUE
Hosts File	–HostsFile	-hosts	TRUE
EVT	–EVT	-evt	TRUE
EVTX	–EVTX	-evtx	TRUE
WMI Active Script	–WMIActiveScript	-wmiasc	TRUE

Continued on next page

Table 1 – continued from previous page

WMI Command Line	–WMICommandLine	-wmicec	TRUE
Prefetch	–Prefetch	-pf	TRUE
ActivitiesDb	–ActivitiesDb	-adb	TRUE
AmCache	–AmCache	-amc	TRUE
RecentFileCache	–RecentFileCache	-rfc	TRUE

2.1.3 Command Line Examples

Collecting all evidence types

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD --profile full
```

Collecting RAM and Page File

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD --profile memory
```

Collecting Custom Evidence (Chrome History, Event Logs, Clipboard)

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD --profile custom -chst -evt -evtx -clp
```

Collecting Default Selected Evidence Types

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD --profile default
```

Performing Memory Triage

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD --triage-ruleset RuleSetName -tm
```

Performing FileSystem and Memory Triage

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD --triage-ruleset RuleSetName -tm -tf
```

Collecting Full Evidence into a predefined case directory

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD -p full --case-dir
↪"C:\Some\Folder\Case"
```

Collecting Full Evidence into a predefined directory (a new folder will be created for each collection)

```
IREC.exe --license AAAA-BBBB-CCDD-DDDD -p full --output-dir "C:\Some\Folder"
```

Running IREC via PsExec

```
PsExec.exe \\192.168.25.137 -u "WIN1064\John" -p "password" -h -n 60 -
↪accepteula -c -f IREC.exe -l AAAA-BBBB-CCCC-DDDD -nw -p full -ad
↪"\\NET\SHARE\IREC" -tr "MyYaraRules" -tm -cc "Hacked Server"
```

2.2 Exit Codes

Note: Default exit code for IREC is 0 which indicates requested operation successfully completed.

Exit Code	Value	Description
SUCCESS	0x0	Operation successfully completed.
ERROR_GENERIC	0x40000000	A generic error occured.
ERROR_ALREADY_RUNNING	0x40000001	Another instance is already running.
ERROR_ELEVATION_REQUIRED	0x40000002	Application is not run as administrator.
ERROR_MAINTENANCE_EXPIRED	0x40000004	Maintenance period expired.
ERROR_NO_DRIVER	0x40000005	Driver not loaded.
ERROR_READ_ONLY_FOLDER	0x40000006	Application is running in a read-only folder.
ERROR_IREC_MEMORY_LOW	0x40000007	System is low on resources.
ERROR_LICENSE	0x40010000- 0x40019999	A license related error occured.
ERROR_EXCEPTION	0x40020000	An exception occured.

Index

Symbols

I