Assassin v1.2 Training

Assassin General Information

Assassin Configuration Sets

- Running
  - Current settings, only stored in memory
  - All modifications to the configuration are made to the running configuration
  - If changes are not explicitly persisted, they will be lost on restart
- Persistent
  - Settings the implant will revert to upon startup
  - Updated using the <persist_settings> command
- Factory
  - Settings the implant had when it was built and originally deployed
  - Can be reverted to using the <restore_defaults> command
  - Stored as a patch in the implant binary and is never modified

Assassin Directories

- Input
  - Directory tasking files are downloaded to and stored in until they can be processed
- Startup
  - Directory tasking files designated for startup execution are moved to and processed whenever the Implant starts
  - The directory may also contain a configuration file of the implant’s persisted settings

Assassin Directories (cont'd)

- Output
  - Directory files are packaged and chunked for the upload queue to be transmitted during beacons
  - Third-party tools may use this feature to forward files to the listening post
- Push
  - Directory where files are packaged and uploaded immediately, ignoring the beacon interval and chunk size
  - If unable to upload the file, it is placed, un-chunked, in the upload queue with priority status
  - Third-party tools may use this feature to forward files to the listening post
- Staging
  - Directory used to store the files in the upload queue
  - Directory is reserved for Implant use

Deployment Executables

- Injection Launchers
  - When the launcher runs, it drops an instance of the Implant DLL to the disk and injects it into an existing Windows SYSTEM process
  - Carries an Implant DLL embedded as a resource, which it is responsible for deploying
- Injection Extractor
  - Carries both the 32- and 64- bit Launchers as resources and runs the correct executable based on the operating system before self deleting

Deployment Executables (cont'd)

- Service Installers
  - When the launcher runs , it registers the Service DLL as a service that should be run by the netsvcs svchost on startup
  - Carries an Implant Service DLL embedded as a resource, which it is responsible for deploying
- Service Extractor
  - Carries both the 32- and 64- bit Implant Service DLLs and installs the appropriate Implant based on the operating system before self deleting

Implant Executables

- Provide the core functionality for the implant
- Can be run directly or through one of the Deployment Executables
- Includes three available types; DLL, Service DLL, and EXE

Assassin Tasking

For Fun and Profit

Run Mode

- <run_mode>
  - Combination of the letters described below:
  - R – the task will be run on receipt
  - S – the task will be run on implant startup
  - P – the task results will be immediately pushed to the LP on execution without chunking

Generate Batch

- generate_batch <run_mode>
  - Task that combines one or more assassin commands into a single task.
  - All embedded commands will be executed in sequence
  - Any error during execution will stop the remaining batch commands from being executed
  - A batch can be exported to XML for later use

File System Tasks

- get <run_mode> <r_file> [offset=0] [bytes=0]
  - Get a file from the target machine
- put <run_mode> <l_file> <r_file> [offset=0] [bytes=0]
  - Put a local file on the target machine
- file_walk <run_mode> <r_dir> <wildcard> <depth> [time_check='no_check'] [date]
  - Walk the directories on the target, collecting information on files specified by the provided parameters
- get_walk <run_mode> <r_dir> <wildcard> <depth> [time_check='no_check'] [date] [offset=0] [bytes=0]
  - Walk the directories on the target, collecting files specified by the provided parameters

File System Tasks (cont’d)

- delete_file <run_mode> <r_file>
  - Delete a file from the target
- delete_secure <run_mode> <r_file>
  - Securely delete a file from the target. The file is overwritten with zeroes before being deleted

Program Execution Tasks

- execute_bg <run_mode> <r_file> [args=‘’]
  - Execute a program on the target in the background. The implant will continue to operate. The standard output and return code of the program is ignored
- execute_fg <run_mode> <r_file> [args=‘’]
  - Execute a program on the target in the foreground. The implant will wait for the program to exit. The standard output and return code of the program are captured and returned

Global Configuration Tasks

- persist_settings <run_mode>
  - Save the current settings as the default configuration that will be loaded at implant startup. Configuration changes must be explicitly persisted, or they will revert on next startup
- restore_defaults <run_mode> <options>
  - Change the running implant configuration to factory settings, any changes must be persisted in order to survive a reboot

Beacon Configuration Tasks

- set_beacon_params <run_mode> [initial=0] [default_int=0] [max_int=0] [factor=0.0] [jitter=0]
  - Set one or more of the beacon parameters. Note that 0 indicates ‘do not alter this value’
- set_blacklist <run_mode> [programs=[]] [files=[]]
  - Set the process blacklist
- set_whitelist <run_mode> [programs=[]] [files=[]]
  - Set the process whitelist

Beacon Configuration Tasks (cont'd)

- safety <run_mode> <seconds>
  - Set the implant beacon interval during idle beacons. This task will not generate a result
- set_interval <run_mode> <seconds>
  - Set the implant beacon interval. This task will not generate a result. The command is used by the 'safety' command and is required by Collide.

Comms Configuration Tasks

- set_transport <run_mode> [xml_file=none]
  - Set the communication transport configuration
- set_chunk_size <run_mode> <chunk_size>
  - Set the chunk size to limit network traffic per beacon

Operation Window Configuration Tasks

- set_hibernate <run_mode> <seconds>
  - Set the hibernate time in seconds after first execution. The implant will lie dormant until the hibernation period has passed
- set_uninstall_date <run_mode> <date>
  - Set the uninstall date for the implant
- set_uninstall_timer <run_mode> <seconds>
  - Set the uninstall timer to seconds from the time that the task is processed by the implant
- set_beacon_failure <run_mode> <count>
  - Set the maximum number of sequential beacon failures that can occur before uninstalling

Maintenance Tasks

- get_status <run_mode> <status_mode> <options>
  - Request the current implant configuration and status information
- clear_queue <run_mode>
  - Clear all files from the implant upload queue. The clear_queue task will delete all files from the output, push, and staging directories on target. This may include file chunks that have been partially uploaded
- upload_all <run_mode>
  - Upload all files currently in the upload queue. The upload_all task will upload all files in the output, push, and staging directories to the listening post as quickly as possible, ignoring the chunk size setting.
- unpersist <run_mode>
  - Stop the implant persistence mechanism on the target
- uninstall <run_mode>
  - Uninstall the implant from the target immediately