

Tyrant Release Notes

Lockheed Martin Advanced Technology Labs, May 18, 2015

In this latest release of Tyrant, the following changes have been made:

Features/Enhancements

- Support for programmatically switching the network to which a Windows ESXi VM is connected (and properly reconfiguring the network interface in Windows guests) has been added. Usage of this new feature is documented in the Tyrant user manual. This change involved:
 - Adding the code for the feature to the tyutils repository (as part of a larger effort to synchronize the three Tyrant repositories between classified and unclassified spaces).
 - Adding new fields "netmask", "gateway", and "dns" to the "vlan" table in the Tyrant database. An update script is provided in the tyworkflow repository (`src/tyworkflow/support/updates/987_vlan_added_fields/update.sh`) which adds the fields to the existing database.
- Post-test hang detection was added to undermine. After completion of a test, but before shutting down the assets, undermine will check to see if each of the assets is still responsive to a palantir ping. If an asset fails to respond to the ping, it is regarded as possibly hung. For each possibly hung asset, a screenshot is taken and stored in the test output directory, allowing later reviewers to see what state the hung machine was in (for example, if it was bluescreened, one could see the error code). If one or more assets are possibly hung, then the test result code is set to ERROR, and the result value is set to a string identifying which hosts are possibly hung (and also noting the original result code and value).
- A new optional method was added to the Leaf class (the class which all class-based leafnodes [the most common type] are based on) to allow custom handling of hung hosts. This method, named "hangDetectedHandler", receives a list of the indices (into the `self.hosts` list available in a leafnode class) of the hosts which are possibly hung.
 - An example implementation of this is provided in the tyutils repository at `leafbag/tyutils/leaf_fetch_dump_on_hang.py`. This module provides a Leaf subclass which implements `hangDetectedHandler` to reboot each possibly hung host and then, if it is a Windows asset, fetch the memory dump, if present. The Leaf subclass is a working example which can be used as a mix-in with other class-based leafnodes (via multiple inheritance).
- As previously mentioned, the tyutils repository was synchronized between classified and unclassified spaces. This involved exporting a significant amount of unclassified code from classified spaces. As a result of this, many new features have been added to tyutils and some bugs have been fixed. Most of these new features are not documented, and could be considered beta-level. One useful new feature is the domain module (`leafbag/tyutils/domain.py`). This module contains functions which can automatically set up domains during a test.
- Code in tybase which looks up network configuration information on Windows assets was reworked to use WMI instead of running utilities like "ipconfig" or "netsh". This avoids

having to specially handle various foreign language editions of Windows, as the commands and field names in WMI are never localized, and also extends support back to Windows 2000.

- Code in tybase which looks up network configuration information on Linux was updated to handle the output format of various command line tools (e.g. ifconfig) on modern Linux distributions.
- A "name" field was added to the "sys.nics" asset property. On Windows, this field gives the name of the interface seen by the user (e.g. "Local Area Connection 2"). On Linux, this field is the same as the "dev" field (the name of the device, such as "eth0").
- The palantir installer was updated to install palantir as a service using the canonical method for popular modern Linux distributions (e.g. systemd for Fedora, upstart for Ubuntu).
- The makefiles used in tybase and tyworkflow were modified to have proper prerequisites. This allows make to determine what parts of Tyrant have been changed and only do the work necessary to incorporate those changes. For example, if the files which make up the palantir installers have not changed, then running "make" in tybase will not rebuild the palantir installers. Support has also been added for running some of the tasks performed by make in parallel. This should provide some speed up in running remote_commit, which sometimes involves a call to "make" in tyworkflow.
- A build of python for 64-bit OS X was added. This allows palantir installed on a 64-bit version of OS X to run natively in 64-bit mode.
- The usb_utils module in tyutils (leafbag/tyutils/usb_utils.py) was updated to use WMI to detect the drive letter at which a newly-added USB is mapped in Windows (greater reliability over the old method) and to attempt to handle some USB errors by formatting the USB drive. In addition, USB drives which repeatedly fail to connect are now marked fubar to prevent possible systemic problems with a USB drive from ruining many scheduled tests.
- Subcommands were added to the db_admin tool in tyworkflow to allow deletion of individual testcases, test plans, or whole test namespaces. This tool will remove the records of the test(s) from the database, and also optionally delete output directories. This can be used to clean up a test server should it start to get full.
- A bug in the code used to retrieve a new unused IP address from the database (such as is used during cloning) was fixed. This bug would cause the code to crash when presented with an IP address from the database which was not a valid IP address (which can be the case if the database contains definitions of USB assets).

Bug Fixes

- A bug was fixed which prevented running test combos in which two different parameters had the same value. The bug involved an improper database uniqueness constraint. An update script is provided in the tyworkflow repository (`src/tyworkflow/support/updates/963_testcase_param_mapping_unique`)
- A bug was fixed in the ESXi cloning support. This bug was caused by an invalid assumption made about the data present in a source VM's VMX (configuration) file during the process of creating the clone's VMX file. Very rarely, a source VM could end up in a state which violated this invalid assumption, causing cloning to fail due to creation of an invalid VMX file for the clone.

- The HAL's use of the ESXi SOAP API was made more robust against connections dropping due to disuse. This was done by tracking the age of each ESXi connection and recreating the connection on-demand once the connection was more than 10 minutes old.
- A bug was fixed which would cause errors about "__main_leaf__" not being found when submitting tests with remote_commit when the remote_commit client's clock was in the future relative to the test server's clock. The bug was caused by use of an old, deprecated python module to pre-compile leafnode python modules to python bytecode (pyc files). This module, when presented with a future dated .py file, would generate a matching pyc file which lost certain elements (including the __main_leaf__ decorator). The fix was to use the current preferred python compiler package, compileall.
- It was discovered that modern glibc, in violation of the ELF specification, requires the unused bytes in the ELF header where we had been placing our salt value (an integer value used to cause the checksum of ELF files to be different) to be zero. This meant that, for example, the salted python binary delivered as part of a salted palantir installation would not run on certain modern Linux distributions. This was fixed by appending the salt value to the end of the ELF file.
- A bug was fixed in palantir installations in which certain .pyd files that were part of the pywin32 package were incorrectly placed at install time, rendering some advanced pywin32 functionality broken. The fix resulted in an increase in the palantir version number and requires a reinstallation of palantir to apply. However, the functionality this bug makes unavailable is very rarely used in our experience, so reinstalling palantir across a range may not be worth the time.
- A bug was fixed in palantir installer building code. Rather than the salt value specified in the palantir.rc file being used, a random number was used. The fix causes the configured salt value to be respected.
- Various bugs involving improper initialization/teardown when using COM on Windows were fixed. These issues were due to not properly calling CoInitialize at the start of COM usage, and/or calling CoUninitialize and then later attempting to use a COM object.
- A race condition in tyworkflow was fixed which would sometimes cause the overmind server to fail to clean up properly during shut down, especially if the database server were responding slowly. This bug would cause incorrect combo counts on overview pages (showing that some tests were running or pending when they in fact were not) and/or would leave some assets marked as in use.