JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

11100 Johns Hopkins Road • Laurel, Maryland 20723-6099

AOS-16-0475

August 23, 2016
v8

(U) Cyber & Threat Intelligence Sharing

(U) MARTI Deployment Guide

T. Christian Anthony
Project Manager
Asymmetric Operations Sector
MARTI@jhuapl.edu

1 Introduction

Mission Analysis and Research of Threat Intelligence (MARTI) is a modular, analytic platform that allows cyber threat intelligence analysts to organize and manage data, share selected information with trusted analysts, analyze malware and threat relationships, and document findings. MARTI is a customized version of the open-source Collaborative Research into Threats (CRITs) software developed by The MITRE Corporation[1]. MARTI uses STIX™ and TAXII™, developed by The MITRE Corporation for the Department of Homeland Security, to enable users to share information as desired between instances of MARTI.

This Deployment Guide provides hardware and software requirements for deploying MARTI. Each organization uses its own MARTI instance on an Ubuntu[2] server that includes a local database, users, sources, services, and data. Users access their local MARTI server via a secure web portal where they can add data, release data to selected feeds, and receive data from feeds to which they are subscribed. Each community of interest that wishes to share data with others has one TAXII™ server that distributes the data feeds.

2 Research and Development Components

The MARTI platform consists of commercial, open-source, and customized technologies that are expanding over time to bring cyber and threat intelligence (CTI) to the cyber defensive mission. This CTI analytical role is still developing, which provides opportunity to innovate and apply people, processes, and technologies in new ways to maximize efficiency and contributions toward the mission.

MARTI is a lightweight and extensible analytical platform that comprises a customized version of CRITs and uses STIX™ and TAXII™ to share indicators of compromise when approved by an analyst. MARTI enables a CTI analyst to learn the cyber domain, conduct analysis, and share indicators as desired. Organizations are not yet ready to enable complete automation of their cyber defenses, so MARTI puts a human interpreter on top of the machine-readable indicators for analysis to enable decision-making. MARTI enables analysis of adversary campaigns, IP addresses, domains, emails, and suspicious files. The fields captured for these objects are outlined in Appendix A – MARTI Object Fields. MARTI interfaces with the following services that can be enabled or disabled as desired:

- Malware Analysis: ChopShop, Strings, ClamAV, Cuckoo sandbox, File carver
- External Metadata Enrichment: VirusTotal, MalShare, MetaChecker, Shodan, WhoIs
- Local Metadata Enrichment: Carbon Black
- Relationship Analysis: Maltego, Timeline service, Relationship service
- Orchestration and Automation: Yara Rule Tester
- Sharing and Reporting: Microsoft Word, Flat text files, Chat, Adobe Reader, STIX™/TAXII™

[1] CRITs license agreement provided at https://github.com/crits/crits/blob/master/LICENSE.

[2] Ubuntu license agreement provided at http://www.ubuntu.com/about/about-ubuntu/licensing.

Additional services can be added to MARTI easily via an API and Python integration.

Future research and development of MARTI can expand upon the role of the CTI analyst and improve the contributions to the cyber defense mission via the following areas:

- Relationship analysis, using Analyst Notebook and other visualization tools;
- Enrichment, orchestration, and automation to enable proactive decision making and response actions as defined by the network defenders as depicted in Figure 1;
- Optimizing and streamlining the human analytical process and decision making to improve efficiency and accuracy; and
- Measuring and assessing the training environment and the performance of technologies, individuals, and teams.

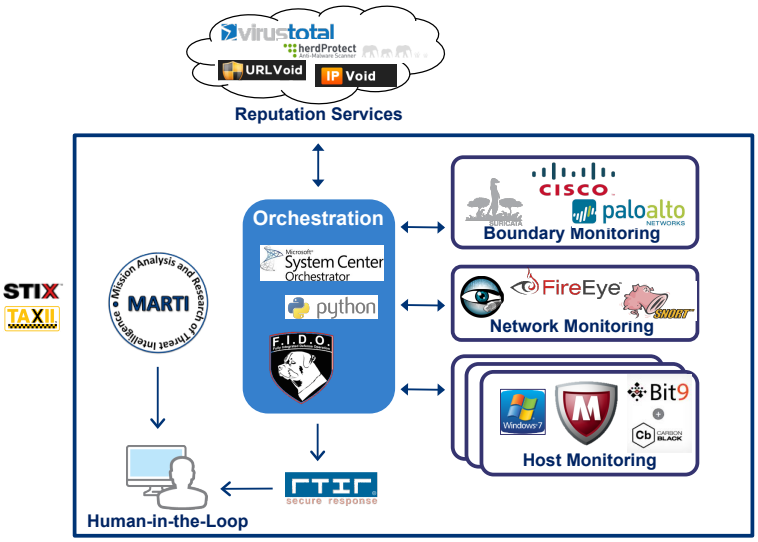

Figure 1. Example MARTI Integration & Automation

3 Concept of Operations

MARTI is a technology to support the Integrated Threat Analysis Capability (ITAC). The ITAC represents an analysis center, such as a State Fusion Center, Joint Operations Center (JOC), or Information Sharing Analysis Organization (ISAO). The ITAC can also be an analytical capability that can be added to existing environments to coordinate responses to distributed cyber threats. Communications and coordination in the ITAC can function in multiple ways depending upon the scope and mission of the deployment. Figure 2 shows one communication structure utilized by the ITAC in a cyber exercise. Each MARTI logo shows a local database of information that can be shared with the other participants as desired.

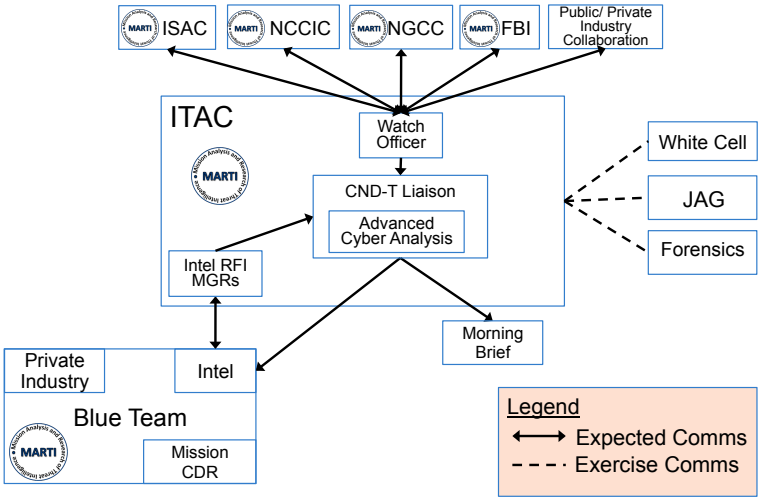

Figure 2. ITAC Command and Control Structure

Participants can share indicators and analysis via MARTI in a peer-to-peer or hub-and-spoke model. In operations, JHU/APL uses a peer-to-peer sharing model for maximum timeliness. For cyber exercises, the hub-and-spoke model is currently in use to prevent teams from achieving too much of an advantage over the opposing forces (OPFOR) or Red Team. The ITAC performs centralized analysis as well as controlling the flow of information in coordination with the White Cell to ensure maximum training value is obtained from the exercise. Figure 3 depicts the flow of information using MARTI in a hub-and-spoke model.

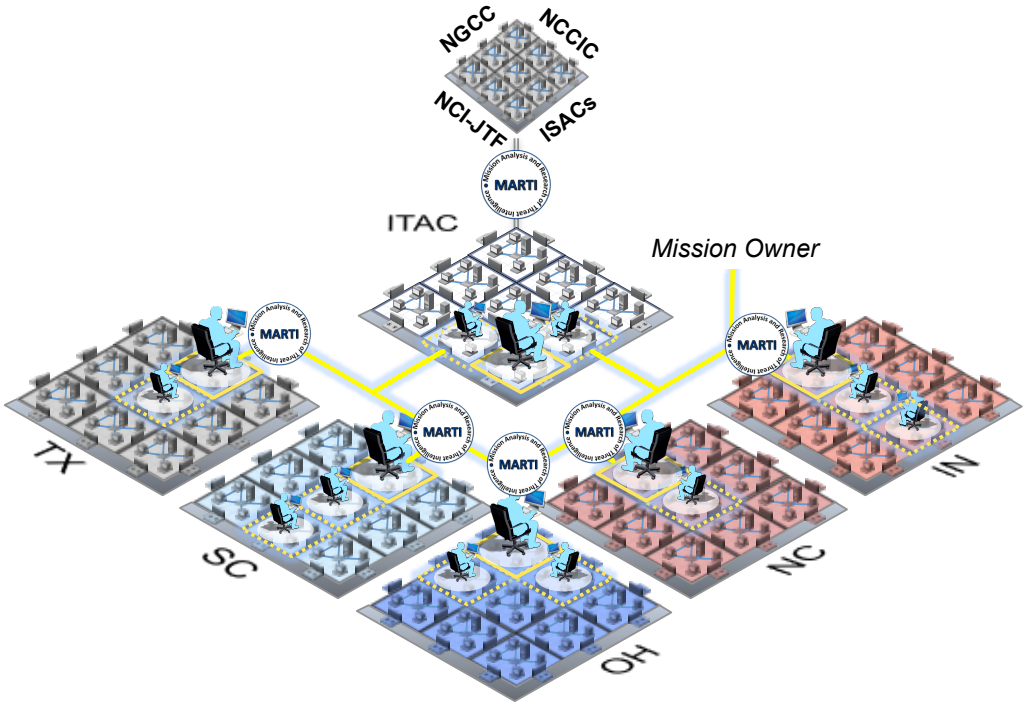

Figure 3. ITAC Concept of Operations

Inside a simulated cyber environment, the ITAC has the following objectives:

- The development of tailored and actionable information to mission owners and cyber defense teams;
- Further definition of the ITAC concept to implement in real-world operations;
- Propose and capture standard operating procedures for the cyber intelligence analyst and how they can support the computer network defense teams; and
- Capture the complexity of the use of titles and authorities used in the cyber intelligence and analysis-sharing construct.

MARTI supports ITAC Measures of Performance (MOPs), such as:

- Legitimacy of information from Blue Team Intelligence analyst to the ITAC based on timeliness and accuracy;
- Applicability of information from the ITAC to the Blue Team based on timeliness, accuracy, tailored, and actionable information; and
- Time to report:
    - From discovery of information to the intelligence analyst,
    - From discovery of information to the mission or network owner, and
    - From the intelligence analyst to the ITAC.

MARTI supports ITAC Measures of Effectiveness (MOEs), such as:

- Effectiveness of recommendation to network owner;
- Accuracy of information reported up and down;
- Percent of information disseminated outside the private company that was approved by the mission and/or the network owner; and
- Improvement in timeliness to respond to threats, such as moving a Computer Network Defense Team left of the cyber boom.

4 Technology Deployment

The MARTI analytical platform consists of two different types of servers:

- Communication Server – Supplies the TAXII™ transport for STIX™ messages using Yeti (https://github.com/TAXIIProject/yeti).
- MARTI Servers – Supplies the MARTI database and web portal for each exercise team or operations center.

Each server runs the Ubuntu operating system on a 64-bit architecture and uses the following software packages:

- Python 2.7, with Python packages as noted in Appendix B – Python Packages
- Apache 2 server
- Mongo DB
- Python Django SSL server
- Yeti 2.0 (on the Communication Server only)
- MARTI – customized CRITs analysis platform with TAXII™ client developed by JHU/APL

The servers can be installed on a single virtual environment, distributed virtual environment, or on physical hardware as required. Recommended hardware settings depend upon the quantity

of traffic, number of users, and number of indicators stored. Minimal settings are 1 CPU, 4 GB of RAM, and 50 GB of disk space. For high intensity usage, 8 CPUs, 16 GB of RAM, and 200 GB of disk space are recommended.

Users access these servers via an Internet Browser (Firefox is recommended) via https over port 443. All communications are encrypted using SSL. The user workstations can run any operating system (Windows recommended) and hardware can be fairly minimal (4 GB of RAM, 15 GB of disk) depending upon the intensity of the local analysis conducted by the analyst. The following software is recommended, though only an Internet Browser is required:

- Firefox Internet Browser
- Maltego (free version from https://www.paterva.com/web6/products/download2.php)
- Chat Client
- Microsoft Office
- Adobe Reader
- Python 2.7

5 Automated Installation

An installation script is available to assist with the MARTI deployment. This script was developed and verified on a fresh install of Ubuntu 16.04 server amd64.

Clone the marti and marti-services folders into /opt/marti and /opt/marti/marti-services respectively, and run the marti-install.sh script.

```
/opt/marti/marti/script/marti-install.sh
```

If the MARTI folders are cloned into a different location, then modify the DIR variable before running the script.

```
#############################################################################
# #                            Global variables
#
#############################################################################
# ... DIR=/opt/marti/marti
```

This script prepares the environment by installing the Python libraries needed to properly run bootstrap and then sets up the Apache SSL options.

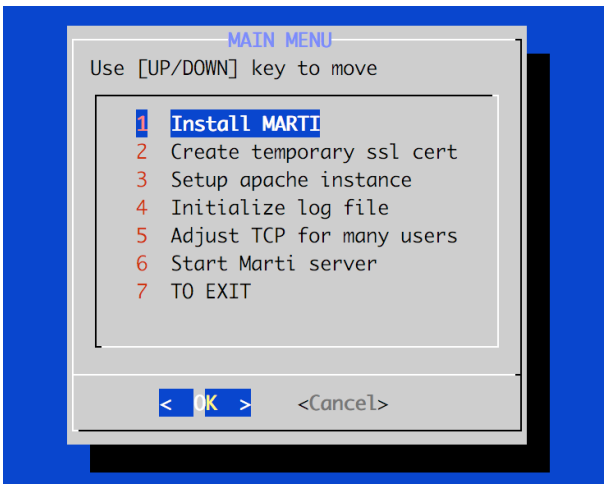

Figure 4. MARTI Installation Script

To setup the machine for production use (with Apache and SSL), you MUST run options 1 -> 2 -> 3 (in order).

Options 4 and 5 are optional, but recommended. The following describes each item in the menu:

1. Install MARTI

   This option installs any dependencies needed to run the CRITs bootstrap script. It then runs the bootstrap script to finish the install. When asked to create/start the database, select 'Y'. In the next menu, select 'a' for "add admin user". [Figure 5]

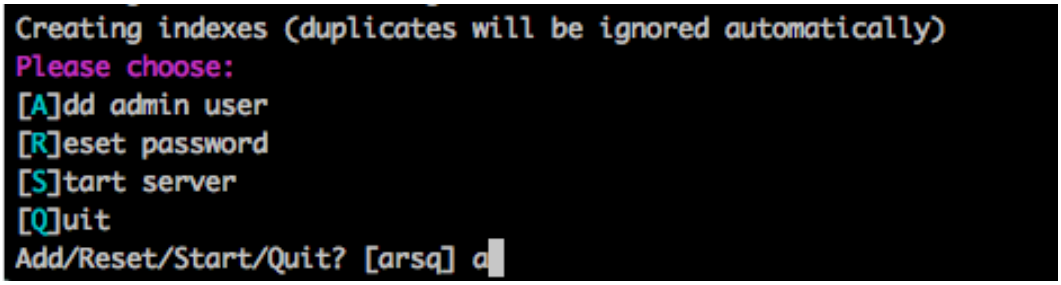

Figure 5. CRITs Bootstrap Script Menu

   Leave the organization name BLANK, as an organization doesn't currently exist in the database and entering a name will cause the admin account to not be created. Write down the temporary password because you will need this to log into the MARTI console after the installation is finished. If you need to reset the password, refer to the CRITs documentation[3] and use manage.py.

[3] CRITs documentation available at https://github.com/crits/crits/wiki

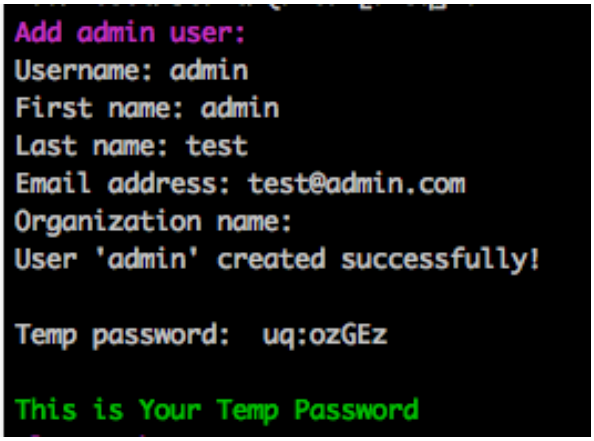

Figure 6. Add an Admin User

Once the user is added, select 'q' to quit and continue with the installation. NOTE: Do NOT start the server until Option 5.

2. Create Temporary SSL Cert

   This option sets up a temporary SSL cert to be used with the Apache setup.

3. Setup Apache Instance

   This option is dependent on options 1 and 2. This will install Apache and set the appropriate files to allow MARTI to run via Apache with SSL.

4. Initialize Log File

   This step was taken from the CRITs production installation wiki. It sets up a "crits" user for logging and cron jobs. Though the administrator will likely not use this account or password, it is advised to select a strong password, because a weak password could pose a security risk.

5. Adjust TCP for Many Users

   This was taken from the CRITs production install wiki and helps with the heavy traffic flow typical in production environments.

6. Start MARTI Server

   This option starts the Apache server. It will only work if the Apache/SSL version was installed.

Review the install script for additional information on how the settings and configurations are implemented.

6 Logging Into MARTI

Each organization has its own MARTI instance that includes a local database, users, sources, services, and data. Users access these servers via an Internet Browser via HTTPS over port 443. All communications are encrypted using SSL.

1. Users log into MARTI by browsing to https://<MARTI server IP address or hostname> in an Internet Browser (Firefox is recommended). Default credentials are provided in Appendix C – Default Credentials. [Figure 7]

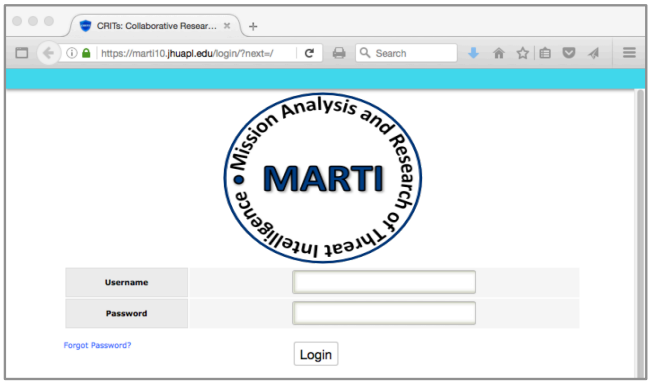

Figure 7. MARTI Login Page

2. When first logging into a MARTI instance, the user needs to accept the security certificate. When the web server is booting, an Internal Server Error screen may be seen. Users should refresh the page and the server will load in less than a minute. [Figure 8]

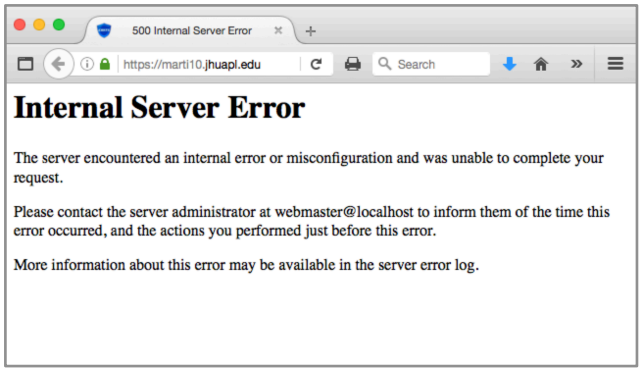

Figure 8. Refresh the Page if MARTI Server is Booting

The MARTI Configuration Guide describes how to customize the interface, create new users, and create new data feeds.

7 Entering Data Into MARTI

This section provides an introduction to how users enter data into MARTI. Additional documentation can be found

1. Select the gear icon in the top left of the screen to open the menu. [Figure 9]

Figure 9. Select the Gear Icon to Open the Menu

2. To add a new object, select the plus sign ("+") next to the Top Level Object (TLO) that is desired. [Figure 10]

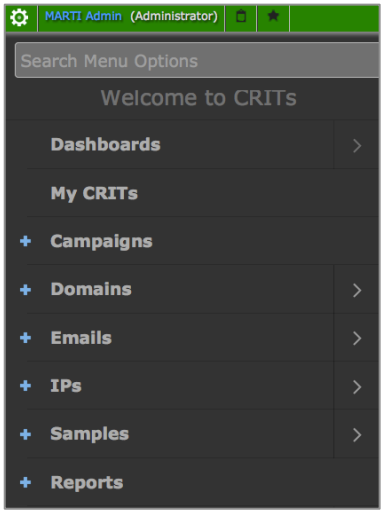

Figure 10. Select the Desired Top Level Object in the Menu

3. A pop-up window will open for the user to enter the desired information. For example, Figure 11 shows the pop-up window to add a new IP address. Note that blue fields are required. Select the button for "IP Address" when complete.

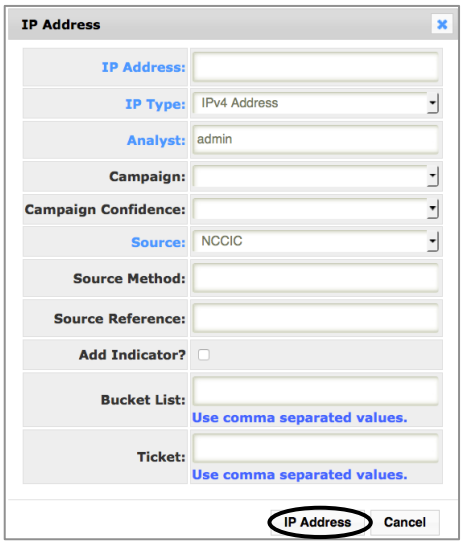

Figure 11. Add Information for a New IP Address

4. Click on the link created to view the new object. [Figure 12]

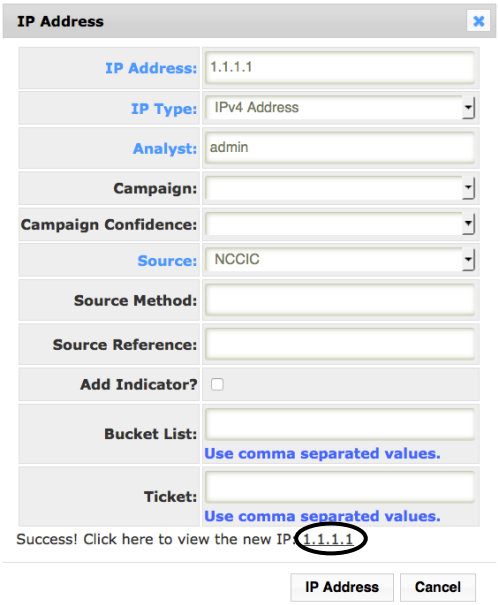

Figure 12. Select the Link to View the New Object

5. Enter additional information, such as the Critical Infrastructure Sectors affected by the indicator, the related Campaigns, and the Traffic Light Protocol value. [Figure 13]

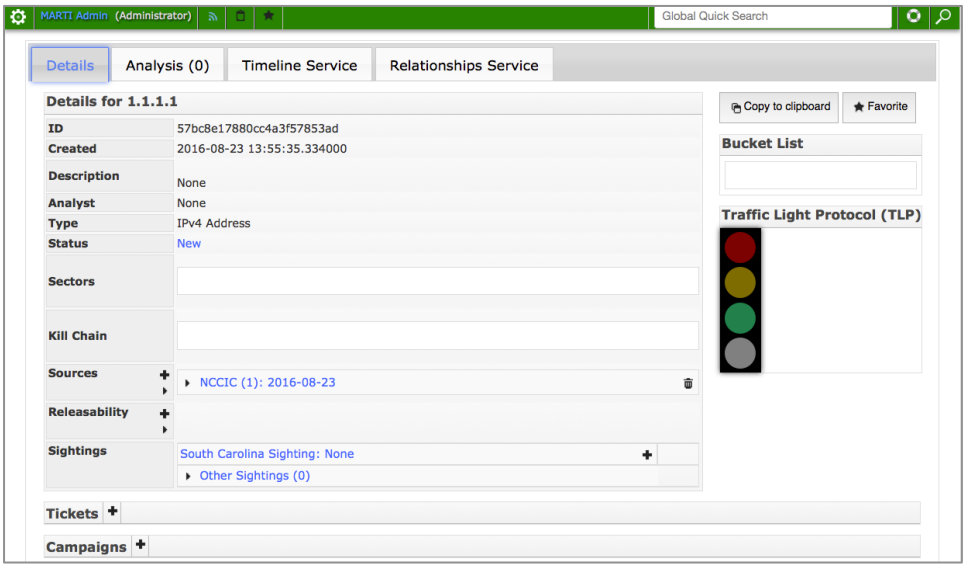

Figure 13. New IP Address Object

8 Appendix A – MARTI Object Fields

This appendix lists all fields contained in each object in MARTI.

- Campaign
  - Date Time Group
  - Reporting Organization
  - Analyst
  - Campaign Name
  - Aliases
  - Description
  - Tactics, Techniques, and Procedures (TTPs)
  - Status (New, In Progress, Analyzed, Deprecated)
  - Sectors (Critical Infrastructure Sectors)
  - Sources
  - Releasability
  - Email Activity
  - Tickets
  - Relationships
  - Comments (Private or Shared)
- IP Address
  - Date Time Group
  - Reporting Organization
  - Analyst
  - IP Address
  - Type (IPv4, IPv6)
  - Description
  - Status (New, In Progress, Analyzed, Deprecated)
  - Sectors (Critical Infrastructure Sectors)
  - Sources
  - Releasability
  - Tickets
  - Campaigns (Name, Confidence, Description)
  - Relationships
  - Comments (Private or Shared)
- Domain
  - Date Time Group
  - Reporting Organization
  - Analyst
  - Domain
  - Description
  - Status (New, In Progress, Analyzed, Deprecated)
  - Sectors (Critical Infrastructure Sectors)
  - Sources
  - Releasability
  - Tickets
  - Campaigns (Name, Confidence, Description)
  - Relationships
  - IP Address (IP, IP Type, Date/Time, Confidence)
  - Comments (Private or Shared)
- Email
  - Date Time Group

    - Reporting Organization
    - Analyst
    - Description
    - From (Email Address)
    - Sender
    - To
    - CC
    - Email Date
    - ISO Date
    - Email Subject
    - X-Mailer
    - Reply-To
    - Message ID
    - HELO
    - Boundary
    - Originating IP
    - X-Originating IP
    - Status (New, In Progress, Analyzed, Deprecated)
    - Sectors (Critical Infrastructure Sectors)
    - Sources
    - Releasability
    - Tickets
    - Campaigns (Name, Confidence, Description)
    - Relationships
    - Comments (Private or Shared)
- File Sample
    - Date Time Group
    - Reporting Organization
    - Analyst
    - Description
    - Filename
    - Additional Filenames
    - File Type
    - MIME Type
    - Size
    - MD5 Hash
    - SHA1 Hash
    - SHA256 Hash
    - SSDeep
    - Status (New, In Progress, Analyzed, Deprecated)
    - Sectors (Critical Infrastructure Sectors)
    - Sources
    - Releasability
    - Tickets
    - Campaigns (Name, Confidence, Description)
    - Relationships
    - Comments (Private or Shared)

9 Appendix B – Python Packages

This appendix list the Python packages installed on each VM.

Communication Server
apt-xapian-index (0.45)
argparse (1.2.1)
chardet (2.0.1)
colorama (0.2.5)
configobj (4.7.2)
Django (1.8.5)
django-solo (1.1.0)
django-sslserver (0.15)
html5lib (0.999)
Landscape-Client (14.12)
libtaxii (1.1.107)
lxml (3.4.4)
PAM (0.4.2)
pip (1.5.4)
pyOpenSSL (0.13)
pyserial (2.6)
python-apt (0.9.3.5ubuntu1)
python-dateutil (2.4.2)
python-debian (0.1.21-nmu2ubuntu2)
requests (2.2.1)
setuptools (3.3)
six (1.10.0)
ssh-import-id (3.21)
taxii-services (0.4)
Twisted-Core (13.2.0)
urllib3 (1.7.1)
wheel (0.24.0)
wsgiref (0.1.2)
zope.interface (4.0.5)

Team Server
amqp (1.4.7)
anyjson (0.3.3)
apt-xapian-index (0.45)
argparse (1.2.1)
billiard (3.3.0.20)
biplist (0.9)
celery (3.1.18)
chardet (2.0.1)
colorama (0.2.5)
configobj (4.7.2)
cybox (2.1.0.12)
defusedxml (0.4.1)
Django (1.6.11)
django-celery (3.1.17)
django-extensions (1.5.9)
django-secure (1.0.1)
django-sslserver (0.15)
django-tastypie (0.11.0)
django-tastypie-mongoengine (0.4.5)
django-wsgiserver (0.6.10)
html5lib (0.999)
kombu (3.0.27)
Landscape-Client (14.12)
libtaxii (1.1.107)
lxml (3.4.4)
M2Crypto (0.22.3)
mongoengine (0.8.7)
olefile (0.42.1)
ordereddict (1.1)
PAM (0.4.2)
Pillow (2.3.0)
pip (1.5.4)
pydeep (0.2)
pymongo (2.7.2)
pyOpenSSL (0.13)
pyparsing (2.0.3)
pyserial (2.6)
python-apt (0.9.3.5ubuntu1)
python-dateutil (2.4.2)
python-debian (0.1.21-nmu2ubuntu2)
python-ldap (2.4.21)
python-magic (0.4.6)
python-mimeparse (0.1.4)
pytz (2015.6)
PyYAML (3.11)
requests (2.2.1)
setuptools (3.3)
simplejson (3.8.0)
six (1.10.0)
ssh-import-id (3.21)
stix (1.2.0.0)
stix-validator (2.4.0)
Twisted-Core (13.2.0)
urllib3 (1.7.1)
ushlex (0.99)
Werkzeug (0.11.2)
wheel (0.24.0)
wsgiref (0.1.2)
xlrd (0.9.4)
yara (1.7.7)
zope.interface (4.0.5)

10 Appendix C – Required Users

The following credentials are required for MARTI. The MARTI Configuration Guide describes how to add and edit user accounts.

1. An administrator account is created during the installation process as described in the MARTI Deployment Guide. An administrator account has permissions to create, edit, and disable user accounts, configure services, and delete data from the database.
2. It is good security practice to create analyst accounts (i.e., user without administrator privileges) for standard operations. Analyst accounts can enter, modify, and share data in MARTI.
3. A reserved administrator account with username “taxii” must be created prior to configuring the TAXII™ service. This account is used for sending and receiving STIX™ messages via TAXII™. In addition, each MARTI instance must have a Source called “taxii” and each user must have the “taxii” Source in his/her list of enabled Sources as described in the MARTI Configuration Guide.