NIST SPECIAL PUBLICATION 1800-5C

IT Asset Management

Volume C:
How-To Guides

Michael Stone
Leah Kauffman, Editor-in-Chief
National Cybersecurity Center of Excellence
Information Technology Laboratory

Chinedum Irrechukwu
Harry Perper
Devin Wynne
The MITRE Corporation
McLean, VA

September 2018

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-5

The first draft of this publication is available free of charge from:
https://www.nccoe.nist.gov/sites/default/files/library/sp1800/fs-itam-nist-sp1800-5-draft.pdf

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST or NCCoE, nor is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 1800-5C, Natl. Inst. Stand. Technol. Spec. Publ. 1800-5C, 166 pages, (September 2018), CODEN: NSPUE2

FEEDBACK

As a private-public partnership, we are always seeking feedback on our Practice Guides. We are particularly interested in seeing how businesses apply NCCoE reference designs in the real world. If you have implemented the reference design, or have questions about applying it in your environment, please email us at financial_nccoe@nist.gov.

All comments are subject to release under the Freedom of Information Act (FOIA).

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive
Mailstop 2002
Gaithersburg, MD 20899
Email: nccoe@nist.gov

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity issues. This public-private partnership enables the creation of practical cybersecurity solutions for specific industries, as well as for broad, cross-sector technology challenges. Through consortia under Cooperative Research and Development Agreements (CRADAs), including technology partners—from Fortune 50 market leaders to smaller companies specializing in IT security—the NCCoE applies standards and best practices to develop modular, easily adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cyber Security Framework and details the steps needed for another entity to recreate the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Md.

To learn more about the NCCoE, visit https://www.nccoe.nist.gov/. To learn more about NIST, visit https://www.nist.gov.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication Series 1800) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align more easily with relevant standards and best practices, and provide users with the materials lists, configuration files, and other information they need to implement a similar approach.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. These documents do not describe regulations or mandatory practices, nor do they carry statutory authority.

ABSTRACT

While a physical asset management system can tell you the location of a computer, it cannot answer questions like, “What operating systems are our laptops running?” and “Which devices are vulnerable to the latest threat?” An effective IT asset management (ITAM) solution can tie together physical and virtual assets and provide management with a complete picture of what, where, and how assets are being used. ITAM enhances visibility for security analysts, which leads to better asset utilization and security.

KEYWORDS

asset management; financial sector; information technology asset management; ITAM; personnel security; physical security; operational security

ACKNOWLEDGMENTS

We are grateful to the following individuals for their generous contributions of expertise and time.

Name	Organization
FS-ISAC	Financial Services Information Sharing and Analysis Center
Gorrell Cheek	Western Union
Joe Buselmeier	American Express
Sean Franklin	American Express
Ron Ritchey	Bank of America
Sounil Yu	Bank of America
Joel Van Dyk	Depository Trust & Clearing Corporation
Dan Schutzer	Financial Services Roundtable
George Mattingly	Navy Federal Credit Union
Jimmie Owens	Navy Federal Credit Union
Mike Curry	State Street
Timothy Shea	RSA
Mark McGovern	MobileSystem7
Atul Shah	Microsoft
Leah Kauffman	NIST
Benham (Ben) Shariati	University of Maryland Baltimore County
Valerie Herrington	Herrington Technologies
Susan Symington	MITRE Corporation
Sallie Edwards	MITRE Corporation

Name	Organization
Sarah Weeks	MITRE Corporation
Lina Scorza	MITRE Corporation
Karen Scarfone	Scarfone Cybersecurity

The Technology Partners/Collaborators who participated in this build submitted their capabilities in response to a notice in the Federal Register. Respondents with relevant capabilities or product components were invited to sign a Cooperative Research and Development Agreement (CRADA) with NIST, allowing them to participate in a consortium to build this example solution. We worked with:

Technology Partner/Collaborator	Build Involvement
AlphaPoint Technology	AssetCentral
Belarc	BelManage, BelManage Analytics
Computer Associates	ITAM
Microsoft	WSUS, Server 2012R2 Certificate Authority
Peniel Solutions	Technology/Industry Expertise
PI Achievers	Penetration Testing Services
PuppetLabs	Puppet
RedJack	Fathom
Splunk	Splunk Enterprise
Tyco	iStar Edge
Vanguard Integrity Professionals	Security Manager

Contents

List of Figures

List of Tables

1 Introduction

The following volumes of this guide show information technology (IT) professionals and security engineers how we implemented this example solution. We cover all the products employed in this reference design. We do not recreate the product manufacturers' documentation, which is presumed to be widely available. Rather, these volumes show how we incorporated the products together in our environment.

Note: These are not comprehensive tutorials. There are many possible service and security configurations for these products that are out of scope for this reference design.

1.1 Practice Guide Structure

This National Institute of Standards and Technology (NIST) Cybersecurity Practice Guide demonstrates a standards-based reference design and provides users with the information they need to replicate all, or parts of the build created in the NCCoE ITAM Lab. This reference design is modular and can be deployed in whole or in part.

This guide contains three volumes:

- NIST SP 1800-5A: *Executive Summary*
- NIST SP 1800-5B: *Approach, Architecture, and Security Characteristics* – what we built and why
- NIST SP 1800-5C: *How-To Guides* – instructions for building the example solution **(you are here)**

Depending on your role in your organization, you might use this guide in different ways:

Business decision makers, including chief security and technology officers, will be interested in the *Executive Summary, NIST SP 1800-5A*, which describes the following topics:

- challenges enterprises face in implementing and using ITAM systems
- example solution built at the NCCoE
- benefits of adopting the example solution

Technology or security program managers who are concerned with how to identify, understand, assess, and mitigate risk will be interested in *NIST SP 1800-5B*, which describes what we did and why. The following sections will be of particular interest:

- Section 4.5, Risk Assessment and Mitigation, where we identify the steps we took to protect and monitor the ITAM system
- Section 4.5.1, Assessing Risk Posture, where we identify the security measures used in this implementation

- Section 4.5.2, Security Characteristics and Controls Mapping, where we map the security characteristics of this example solution to cybersecurity standards and best practices
- Section 4.6, Technologies, where we identify the products and technologies we used and map them to the relevant security controls

You might share the *Executive Summary, NIST SP 1800-5A*, with your leadership team members to help them understand the importance of adopting standards-based IT Asset Management.

IT professionals who want to implement an approach like this will find this whole practice guide useful. You can use this How-To portion of the guide, *NIST SP 1800-5C*, to replicate all or parts of the build created in our lab. This How-To portion of the guide provides specific product installation, configuration, and integration instructions for implementing the example solution. We do not recreate the product manufacturers' documentation, which is generally widely available. Rather, we show how we incorporated the products together in our environment to create an example solution.

This guide assumes that IT professionals have experience implementing security products within the enterprise. While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products. Your organization can adopt this solution or one that adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing parts of IT Asset Management. Your organization's security experts should identify the products that will best integrate with your existing tools and IT system infrastructure. We hope that you will seek products that are congruent with applicable standards and best practices. Section 4.6, Technologies, lists the products that we used and maps them to the cybersecurity controls provided by this reference solution.

A NIST Cybersecurity Practice Guide does not describe "the" solution, but a possible solution. Comments, suggestions, and success stories will improve subsequent versions of this guide. Please contribute your thoughts to financial_nccoe@nist.gov.

1.2 Build Overview

The NCCoE constructed the Information Technology Access Management (ITAM) build infrastructure using commercial off-the-shelf (COTS) hardware and software along with open source tools.

The lab network is connected to the public Internet through a virtual private network (VPN) appliance and firewall to enable secure Internet and remote access. The lab network is not connected to the NIST enterprise network. Table 1-1 lists the software and hardware components used in the build, as well the specific function each component contributes.

Table 1-1 Build Architecture Component List

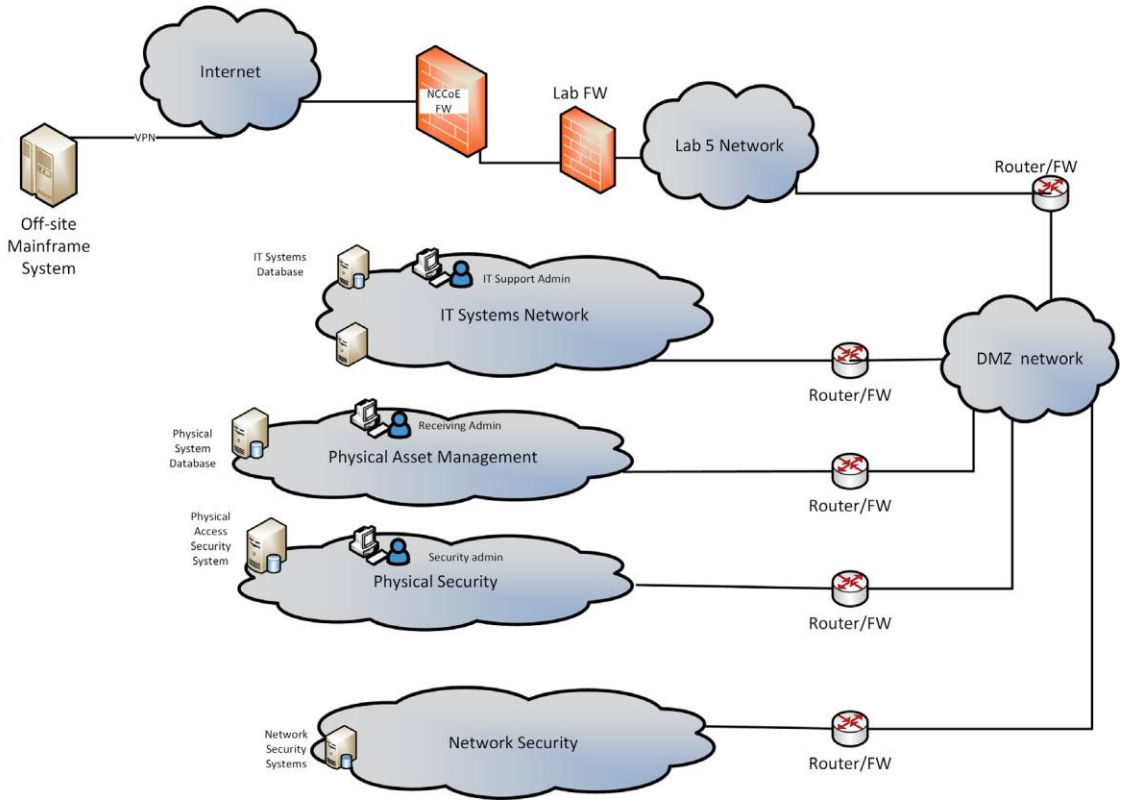
Host	Product	Function	Internet Protocol Address	Operating System
Demilitarized Zone				
Bro	Bro	Network security monitor	172.16.0.20	Ubuntu 14.04
FathomSensor	RedJack Fathom	Network analysis	172.16.0.50	CentOS 7
OpenSwan	OpenSwan	Virtual Private Network (VPN)	172.16.0.67	Ubuntu 14.04
Router0	pfSense	Router/firewall	172.16.0.11 10.33.5.9	BSD pfSense appliance
Snort	Cisco/Sourcefire Snort	Intrusion Detection System	172.16.0.40	Ubuntu 14.04
Apt-cacher0	Ubuntu apt-cacher	Patch management	172.16.0.77	Ubuntu 14.04
WSUS	Microsoft WSUS	Patch management	172.16.0.45	Server 2012R2
IT Systems				
AD1	Microsoft Active Directory	Directory manager, AAA, DNS	172.16.1.20	Server 2012R2
AD2	Microsoft Active Directory	Directory manager, AAA, DNS	172.16.1.21	Server 2012R2
CA server	Microsoft Certificate Authority	PKI certificate authority	172.16.1.41	Server 2012R2
Email Server	Postfix	Email server for the lab	172.16.1.50	Ubuntu 14.04
PE Master	Puppet Labs Puppet Enterprise	Configuration management	172.16.1.40	Ubuntu 14.04
Router1	pfSense	Router/firewall	172.16.0.12 172.16.1.1	BSD pfSense appliance
Ubuntu Client1	Ubuntu Desktop	Representative Linux client	DHCP	Ubuntu 14.04
Win7-Client1	Microsoft Windows7	Representative Windows client	DHCP	Windows 7 Enterprise
Win7-Client2	Microsoft Windows7	Representative Windows client	DHCP	Windows 7 Enterprise
Network Security				
Router2	pfSense	Router/firewall	172.16.0.13 172.16.2.11	BSD pfSense appliance
BelManage	Belarc BelManage	Software, hardware, configuration information	172.16.2.71	Windows Server 2012R2

Host	Product	Function	Internet Protocol Address	Operating System
BDA	Belarc BelManage Data Analytics	Analytic information for BelManage	172.16.2.72	Windows 7
OpenVAS	OpenVAS	Vulnerability analysis system	172.16.2.33	Ubuntu 14.04
Physical Asset Management				
Router3	pfSense	Router/firewall	172.16.0.14 172.16.3.11	BSD pfSense appliance
AssetCentral	AlphaPoint AssetCentral	IT and datacenter asset management system	172.16.3.103	CentOS7
CA ITAM	CA Technologies IT Asset Manager	Lifecycle asset manage-ment	172.16.3.92	Windows Server 2012R2
Physical Security				
Router4	pfSense	Router/firewall	172.16.0.15 192.168.1.11	BSD pfSense appliance
iStar Edge	Tyco iStar Edge	Security system with badge reader for door access	192.168.1.169	Embedded
NVR	Tyco/American Dynamics VideoEdge	Digital video recorder for IP security cameras	192.168.1.178	Suse Linux (JeOS)
Camera1	Illustra 600 IP camera	IP security camera	192.168.1.176	Embedded
Camera2	Illustra 600 IP camera	IP security camera	192.168.1.177	Embedded
CCure9000	CCure9000	Controller for iStar Edge and NVR	192.168.1.167	Windows 7
ITAM				
Router5	pfSense	Router/firewall	172.16.0.16 172.16.5.11	BSD pfSense appliance
Splunk	Splunk Enterprise	Data aggregation, storage, analysis and visualization	172.16.5.55	RHEL 7

1.2.1 Build Architecture Components Overview

The build architecture consists of multiple networks implemented to mirror the infrastructure of a typical financial industry corporation. The networks include a Demilitarized Zone (DMZ) network along with several subnets as shown in Figure 1-1. The DMZ network provides technologies that monitor and detect cybersecurity events, conduct patch management, and provide secure access to the mainframe computer. The Physical Asset Management Network provides management of identities and credentials for authorized devices and users. Network Security provides vulnerability scanning, along with a database for collection and analysis of data from hardware and software components. The IT Systems Network conducts configuration management and validation of client machines. Physical Security consists of management consoles for devices that operate and manage physical security. Such devices consist of badge readers and cameras. Firewalls are configured to limit access to and from the networks, blocking all traffic except required internetwork communications.

Figure 1-1 ITAM Build



1.2.2 Build Network Components

Internet – The public Internet is accessible by the lab environment to facilitate access for vendor software and NCCoE administrators. Internet access is not required to implement the build.

VPN Firewall – The VPN firewall is the access control point for vendors to support the installation and configuration of their components of the architecture. The NCCoE also used this access to facilitate product training. This firewall also blocks unauthorized traffic from the public Internet to the production networks. Additional firewalls are used to secure the multiple domain networks (ITAM, DMZ, Network Security, IT Systems, Physical Security, Physical Asset Management). Each network uses pfSense routers for all of its routing and firewall needs. The router is also performing duties as an NTP server and DHCP server on all subnets except the DMZ, which does not allow DHCP.

Demilitarized Zone – The DMZ provides a protected neutral network space that the other networks of the production network can use to route traffic to/from the Internet or each other. There is an external and internal facing subnet. The DMZ also provides technologies that monitor and detect cybersecurity events, conduct patch management, and issue secure access to the mainframe computer. DMZ devices consist of Router0, Ubuntu Apt-Cacher, Bro, Fathom Sensor, Snort and WSUS.

ITAM – The ITAM network contains the Splunk Enterprise server that serves as the IT asset management database. The Splunk Enterprise server gathers logging and status information from all machines in the environment. The ITAM network also contains Router5.

Network Security – The network security architecture is represented in Figure 1-1. Network security is where all devices pertaining to network security reside. These devices include Intrusion Detection System/Intrusion Prevention System (IDS/IPS), Security Event and Incident Management (SEIM), logging systems and vulnerability scanners. Devices within this network consist of Router2, OpenVAS, Belarc and Splunk Enterprise servers.

IT Systems – The IT systems network is dedicated to traditional IT systems. Examples of such systems are Domain Name System (DNS), Active Directory, email, certificate authority, internal Web servers and client machines. Devices included in this subnet are Router1, two Windows 7 clients, a Wiki and two Windows 2012 Active Directory servers. One serves as primary while the other serves as a backup. Puppet Enterprise Master enforces security and configuration baselines across all endpoints.

Physical Security – The physical security network houses the devices that operate and manage physical security, such as badge readers and cameras, along with their management consoles. The devices include Router4, iStar Edge, CCure controller, two badge readers and two Internet Protocol (IP) cameras.

Physical Asset Management – The physical asset management network contains devices that provide and collect information regarding physical assets. The devices include Router3, AssetCentral and CA Technologies IT Asset Manager. AssetCentral is a physical asset inventory and analysis system from AlphaPoint Technology. It allows users to view assets from multiple viewpoints, including building,

room, floor, rack, project, collection, or owner. AssetCentral is running on CentOS Linux. CA IT Asset Manager allows users to holistically manage IT hardware assets, from planning and requisition to retirement and disposal.

1.2.3 Operating Systems

All machines used in the build had either Windows 7 enterprise, Windows server 2012 R2, Ubuntu 14.04, RedHat Enterprise Linux 7.1 or CentOS 7 operating systems (OSs) installed.

1.2.3.1 Base Windows Installation and Hardening Details

The NCCoE base Windows OS images are Server 2012 R2 x86_64 and Windows 7 Enterprise x86_64 Department of Defense (DoD) Security Technical Implementation Guide (STIG) images. The installation of both Windows systems was performed using installation media provided by the Defense Information Systems Agency (DISA). These images were chosen because they are standardized, hardened and fully documented.

1.2.3.2 Base Linux Installation and Hardening Details

The NCCoE base Linux OS is CentOS 7. This OS is available as an open source image. The OS was configured to meet the DoD CentOS 6, STIG. No CentOS 7 STIG was available at the time the build was implemented.

1.3 Typographic Conventions

The following table presents typographic conventions used in this volume.

Typeface/Symbol	Meaning	Example
Italics	file names and path names; references to documents that are not hyperlinks; new terms; and placeholders	For detailed definitions of terms, see the *NCCoE Glossary*.
Bold	names of menus, options, command buttons, and fields	Choose **File** > **Edit**.
`Monospace`	command-line input, on-screen computer output, sample code examples, and status codes	`mkdir`
`Monospace Bold`	command-line user input contrasted with computer output	**`service sshd start`**
blue text	link to other parts of the document, a web URL, or an email address	All publications from NIST's NCCoE are available at https://www.nccoe.nist.gov.

2 Tier 1

2.1 Software Configurations

2.1.1 Splunk Enterprise

Splunk Enterprise is a software platform to search, analyze, and visualize the machine-generated data gathered from the websites, applications, sensors, and devices that comprise your IT infrastructure or business. Splunk Enterprise is comprised of a database, analytic engine, front-end and various ways of gathering data.

2.1.2 How It's Used

In the FS ITAM build Splunk Enterprise receives data from all of the sensors and IT asset management systems. Splunk Enterprise then indexes the data, analyzes it, and displays the results as both reports and graphical desktops.

Analysts can quickly view reports and dashboards to view commonly requested information. Analysts can also form ad-hoc queries on any of the data gathered and analyzed. Splunk Enterprise also provides the ability to alert on any security or performance event.

On the high-level architecture diagram Splunk Enterprise is the Tier 1 ITAM server. Splunk Enterprise is running its own syslog server and collecting syslog information from all hosts on the network (port 514 TCP/UDP). Splunk Enterprise utilizes several methods to acquire data from the ITAM systems which are shown in Table 2-1. The Splunk Enterprise server listens on TCP port 9997 for connections from Universal Forwarders.

Table 2-1 Splunk Enterprise Data Collection Methods

Product	Method
AssetCentral	Database Connection
Bro	Splunk Universal Forwarder
CA Technologies ITAM	Database Connection
Snort	Splunk Universal Forwarder
Fathom	Splunk Universal Forwarder
BelManage	Database Connection
Puppet	Splunk Universal Forwarder
Tyco	Files & Directories
WSUS	Splunk Universal Forwarder
OpenVAS	Splunk Universal Forwarder
Vanguard	Splunk Universal Forwarder

2.1.3 Installing Splunk Enterprise

1. Splunk Enterprise is installed on a hardened RedHat Enterprise Linux system. Please download the latest RPM file from Splunk and follow the instructions for installing from an RPM file. Installation was performed following the instruction from Splunk at http://docs.splunk.com/Documentation/Splunk/6.2.3/Installation/InstallonLinux#RedHat_RP% 20M_install.

2. After installing the RPM file (explained in the Splunk Enterprise installation instructions), the following steps are recommended to start Splunk Enterprise automatically at boot time:

   ```
   cd <splunk install_directory>/bin
   ```

   Commonly: `cd /opt/splunk/bin`

   ```
   ./splunk start --accept-license
   ```

```
./splunk enable boot-start

./splunk enable boot-start -user splunkuser

./splunk start
```

3. Splunk Enterprise also requires several ports to be opened through the firewall(s). To allow these ports through the built-in firewall on RHEL, enter the following commands:

```
sudo firewall-cmd -permanent --add-port =8000/tcp

sudo firewall-cmd -permanent --add-port =9997/tcp

sudo firewall-cmd -permanent --add-port =514/tcp

sudo firewall-cmd -permanent --add-port =514/udp

sudo firewall-cmd -reload

sudo firewall-cmd -list-ports
```

4. It is also recommended to increase the number of files that can be open simultaneously. This is done by editing the */etc/security/limits.conf* file. Please add the following lines to the end of */etc/security/limits.conf*:

```
soft nproc 8192

hard nproc 8192

soft nofile 8192

soft nofile 8192
```

   Note: These will not take effect until you log off and on again. You can issue the **`ulimit–a`** command to verify that it worked.

5. Splunk Enterprise can now be accessed by opening a web browser and going to

   http://localhost:8000

   Initial login = admin

   Initial password = changeme

2.1.3.1 Disable Transparent Huge Pages

Using Transparent Huge Pages causes performance degradation of up to 30% when using Splunk Enterprise. Splunk recommends disabling Huge Transparent Pages and details the issue at http://docs.splunk.com/Documentation/Splunk/6.2.3/ReleaseNotes/SplunkandTHP.

1. To disable Transparent Huge Pages, we added the following lines to the end of */etc/rc.d/rc.local*:

```
#disable THP at boot time
```

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then echo never >
/sys/kernel/mm/transparent_hugepage/enabled

fi

if test -f /sys/kernel/mm/transparent_hugepapge/defrag; then echo never >
sys/kernel/mm/transparent_hugepapge/defrag

fi
```

2. Ensure that rc.local is executable:

```
chmod +x /etc/rc.d/rc.local
```

3. Run the rc.local script to make the changes:

```
/etc/rc.d/rc.local
```

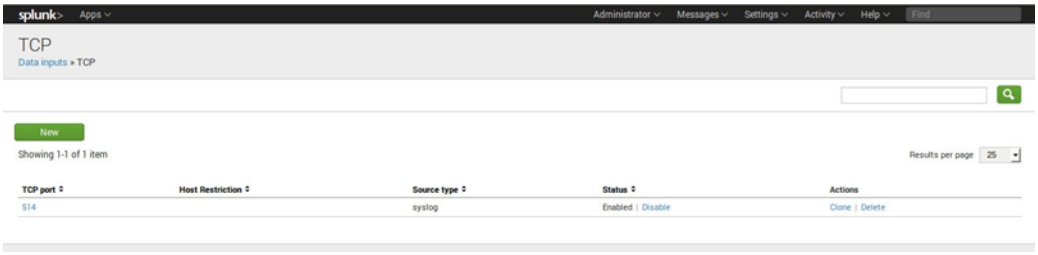
2.1.4 Configurations

2.1.4.1 Splunk Enterprise Data Inputs

2.1.4.1.1 Syslog TCP

1. Go to **Settings** > **Data Inputs** > **TCP**.

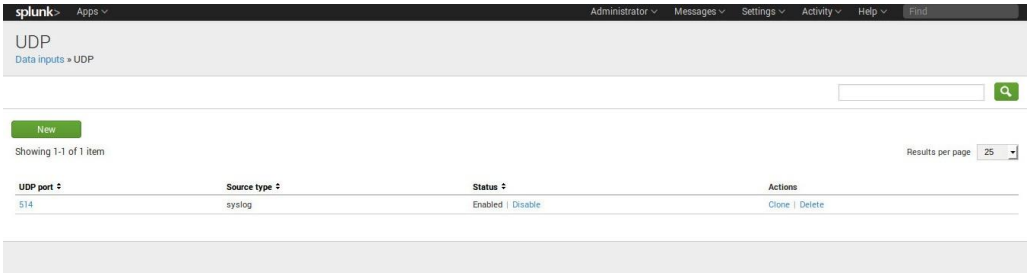
Figure 2-1 Splunk Enterprise Syslog TCP Input



2.1.4.1.2 Syslog UDP

1. Go to **Settings** > **Data Inputs** > **UDP**.

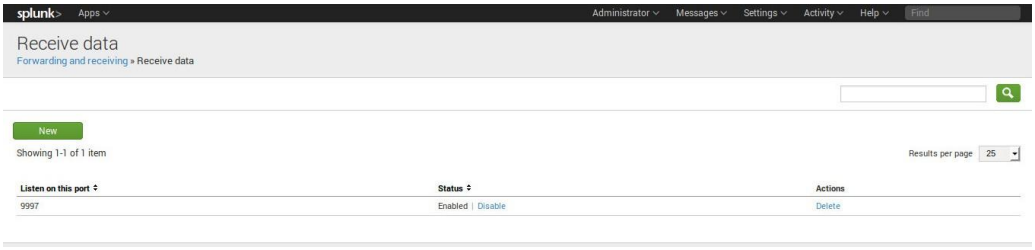
Figure 2-2 Splunk Enterprise Syslog UDP Input



2.1.4.1.3 Receive Data from Splunk Universal Forwarders

1. Go to **Settings** > **Forwarding and Receiving** > **Configure Receiving**.
2. Click the **New** button, and enter port **9997**.

Figure 2-3 Splunk Enterprise Receive from Splunk Universal Forwarder



2.1.4.2 Splunk Enterprise Indexes

Splunk Enterprise stores events in indexes. By default, the main index holds all events. However, using multiple indexes has several benefits including controlling user access to events, different retention policies for different events, and faster searches in certain situations. A separate index was created for each input type and stored in the data directory (*/data/splunk*). Table 2-2 contains the list of indexes that were created.

To create a new index, follow these steps.

1. On the web page for Splunk Enterprise (https://172.16.5.55:8000).
2. Navigate to **Settings > Indexes**. Then, click **New**.
3. Enter a **Name** for the index (see Table 1-1 for the list of names).
4. Ensure that the **Home Path** is set to */data/splunk*.

Follow the above steps for each index that you need to create. For additional information on indexes, go to: http://docs.splunk.com/Documentation/Splunk/6.2.3/Indexer/Setupmultipleindexes.

Table 2-2 Splunk Enterprise Indexes

Index Name
alerts
assetcentral
belmanage_computers
belmanage_hotfixesmissing
belmanage_hw_changes

Index Name
belmanage_sw_changes
belmanage_software
bro
ca_itam
fathom
firewall
mainframe
openvas
puppet
router_configs
snort
syslog
tyco
wsus

2.1.4.3 Splunk Enterprise Apps

Several Splunk Enterprise Apps were used in this project. The list of Splunk Enterprise Apps needed for the ITAM project can be found in Table 2-3. Splunk Enterprise Apps assist in processing, analyzing and displaying different types of data. To download Splunk Enterprise Apps you must have a valid Splunk account. You can install Splunk Enterprise Apps from https://splunkbase.splunk.com/.

To install Splunk Enterpise Apps, follow these steps:

1. Download App from https://splunkbase.splunk.com/.
2. On Splunk Enterprise web (https://172.16.5.55:8000).
    a. **Apps** (top left of web page) > **Manage Apps**
    b. Click **Install app from file**.

Table 2-3 Splunk Enterprise Apps

Splunk Add-On for Bro	Extracts information from Bro logs.
Splunk WebLog Add-On	Extracts information from web logs, such as those from an Apache server.
Splunk for Snort	Extracts information from Snort logs.

Splunk DB Connect v2	Run queries on external databases and stores the info in Splunk Enterprise indexes.
Splunk App for CEF	Extracts Common Event Format data
Technology Add-On for pfSense	Extracts information from pfSense router logs.
IP Reputation	Provides IP reputation information for Splunk Enterprise queries.
Google Maps	Provides geographic information and display for IP addresses.

The Splunk DB Connect v2 app requires the downloading and installation of specific database drivers. Database-specific drivers should be placed in the directory *$SPLUNK_HOME/etc/apps/splunk_app_db_connect/bin/lib*. This project required the installation of database drivers for Microsoft SQL and MySQL. The drivers must be obtained from the database manufacturers; in this case Microsoft and MySQL/Oracle. For more detailed information, please refer to **Install database drivers** at http://docs.splunk.com/Documentation/DBX/latest/DeployDBX/Installdatabasedrivers. The required drivers are listed in Table 2-4.

Table 2-4 Required Database Drivers

Database	Driver
Microsoft SQL	sqljdbc4.jar
MySQL	mysql-connector-java-5.1.36-bin.jar

2.1.4.4 Splunk Enterprise Connections

This section provides information about setting up connections that use the Splunk Enterprise DB Connect v2 app. The Splunk Enterprise DB Connect v2 app is used to connect to the following external databases: AssetCentral, BelManage and CA-ITAM.

To get data from an external database Splunk Enterprise DB Connect v2 requires 3 main steps:

1. Setup an identity. The identity is the username used to log into the database.
2. Setup a connection. The connection is the network and database information.
3. Setup an operation. The operation is what you want to do with the database (run an SQL query).

Table 2-5 provides the information needed to perform these steps.

Table 2-5 DB Connect v2 Identities

Identity	Used with
asset_query	AssetCentral
mike	BelManage
splunk	CA ITAM

2.1.4.4.1 Splunk Enterprise DB Connect v2 Connections

There should only be one database connection to each individual database. The database connections use the identities listed in Table 2-5. Please remember to select the **Enable** button when you configure each connection.

DB Connect V2 AssetCentral Connection:

- AssetCentral
- Status: Enabled
- Connection Name: assetcentral
- App: Splunk DB Connect v2
- Host: assetcentral
- Database Types: MySQL
- Default Database: assetcentral
- Identity: asset_query
- Port: 3306
- Enable SSL: NOT CHECKED
- Readonly: NOT CHECKED

DB Connect V2 BelManage Connection:

- BelManage
- Status: Enabled
- Connection Name: BelManage
- App: Splunk DB Connect v2
- Host: belmanage
- Database Types: MS-SQL Server Using MS Generic Driver
- Default Database: BelMonitor82_1

- Identity: mike
- Port: 1433
- Enable SSL: NOT CHECKED
- Readonly: NOT CHECKED

DB Connect V2 CA-ITAM Connection:

- CA-ITAM
- Status: Enabled
- Connection Name: ca-itam
- App: Splunk DB Connect v2
- Host: ca-itam
- Database Types: MS-SQL Server Using MS Generic Driver
- Default Database: mdb
- Identity: splunk
- Port: 1433
- Enable SSL: NOT CHECKED
- Readonly: NOT CHECKED

2.1.4.4.2 Splunk Enterprise DB Connect v2 Operations

Operations are the SQL operations performed on the database connections and the results are saved into Splunk Enterprise indexes. The operations can be run automatically, on a recurring basis, or when new data is detected.

Each operation has four components:

- Name Input
- Choose and Preview Table
- Set Parameters
- Metadata

The following subsections show the configurations for each operation.

AssetCentral:

DB Input: assetcentral

1. Name Input
    a. Status: Enabled
    b. Name: assetcentral
    c. Description: Assets from AssetCentral
    d. App: Splunk DB Connect v2
    e. Connection: assetcentral
    f. Click the **Continue** button.
2. Choose and Preview Table
    a. Make sure that **Simple Query Mode** is selected.
    b. Catalog: assetcentral
    c. Schema: NULL
    d. Table: assetview
    e. Max rows: 100
    f. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.
    g. Click the **Continue** button.
3. Set Parameters
    a. Type: Batch Input
    b. Max Rows to Retrieve: 100000
    c. Timestamp: Current Index Time
    d. Output Timestamp Format: YYYY-MM-dd HH:mm:ss
    e. Execution Frequency: 0 0 * * *
    f. Click the **Continue** button.

4. Metadata
    a. Source: assetcentral
    b. Sourcetype: assetcentral
    c. Index: assetcentral
    d. Select Resource Pool: local
    e. Click the **Save** button.

BelManage_Computers:

DB Input: BelManage_Computers

1. Name Input
    a. Status: Enabled
    b. Name: BelManage_Computers
    c. Description: Computer info from BelManage
    d. App: Splunk DB Connect v2
    e. Connection: BelManage
    f. Click the **Continue** button.
2. Choose and Preview Table
    a. Make sure that **Simple Query Mode** is selected.
    b. Catalog: BelMonitor82_1
    c. Schema: dbo
    d. Table: Computers
    e. Max rows: 100
    f. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.
    g. Click the **Continue** button.
3. Set Parameters
    a. Type: Rising Column
    b. Max Rows to Retrieve: 100000

c. Specify Rising Column: ProfileDate

d. Timestamp: Current Index Time

e. Output Timestamp Format: YYYY-MM-dd HH:mm:ss

f. Execution Frequency: * * * * *

g. Click the **Continue** button.

4. Metadata

a. Source: belmanage

b. Souretype: belmanage_computers

c. Index: belmanage_computers

d. Select Resource Pool: local

e. Click the **Save** button.

Belmanage_hotfixesmissing:

DB Input: belmanage_hotfixesmissing

1. Name Input

a. Status: Enabled

b. Name: belmanage_hotfixesmissing

c. Description: List of hotfixes/patches missing from each computer

d. App: Splunk DB Connect v2

e. Connection: BelManage

f. Click the **Continue** button.

2. Choose and Preview Table

a. Make sure that **Advanced Query Mode** is selected.

b. In the entry box type in the following SQL statement:

```
SELECT HotfixesMissing.*, Computers.ProfileName,
Comput-ers.NetworkIPAddress FROM HotfixesMissing INNER JOIN Computers on
HotfixesMissing.Id = Computers.Id
```

c. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.

d. Click the **Continue** button.

3. Set Parameters

    a. Type: Batch Input

    b. Max Rows to Retrieve: 100000

    c. Timestamp: Current Index Time

    d. Output Timestamp Format: YYYY-MM-dd HH:mm:ss

    e. Execution Frequency: 30 4 * * *

    f. Click the **Continue** button.

4. Metadata

    a. Source: belmanage

    b. Sourcetype: belmanage_hotfixesmissing

    c. Index: belmanage_hotfixesmissing

    d. Select Resource Pool: local

    e. Click the **Save** button.

Belmanage_hw_changes:

DB Input: belmanage_hw_changes

1. Name Input

    a. Status: Enabled

    b. Name: belmanage_hw_changes

    c. Description: BelManage hardware changes

    d. App: Splunk DB Connect v2

    e. Connection: BelManage

    f. Click the **Continue** button.

2. Choose and Preview Table

    a. Make sure that **Simple Query Mode** is selected.

b. Catalog: BelMonitor82_1

c. Schema: dbo

d. Table: HistoryReportAllHardware

e. Max rows: 100

f. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.

g. Click the **Continue** button.

3. Set Parameters

a. Type: Rising Column

b. Max Rows to Retrieve: 10000

c. Specify Rising Column: ActionDate

d. Timestamp: Current Index Time

e. Output Timestamp Format: YYYY-MM-dd HH:mm:ss

f. Execution Frequency: */15 * * * *

g. Click the **Continue** button.

4. Metadata

a. Source: belmanage

b. Sourcetype: belmanage_hw_changes

c. Index: belmanage_hw_changes

d. Select Resource Pool: local

e. Click the **Save** button.

Belmanage_software:

DB Input: belmanage_software

1. Name Input

a. Status: Enabled

b. Name: belmanage_software

c. Description: Software from BelManage

d. App: Splunk DB Connect v2

e. Connection: BelManage

f. Click the **Continue** button.

2. Choose and Preview Table

a. Make sure that **Advanced Query Mode** is selected.

b. In the entry box type in the following SQL statement:

```
SELECT

ProfileName,

Directory,

C.ProfileDate AS ProfileDate_soft, CAST(C.ProfileDate AS DATE) AS
ProfileDateDate_soft,

DATEDIFF (dd, ProfileDate, GETDATE() ) AS ProfileDateDaysAgo_soft,
DATEDIFF (mm, ProfileDate, GETDATE() ) AS ProfileDate-MonthsAgo_soft,

CASE WHEN CAST ( (CAST(GETDATE() AS FLOAT) - CAST(ProfileDate AS FLOAT))
AS INT) < 31 THEN 'yes' ELSE 'no' END AS

ProfileDateWithin-Last30Days_soft,

CASE WHEN CAST ( (CAST(GETDATE() AS FLOAT) - CAST(ProfileDate AS FLOAT))
AS INT) < 61 THEN 'yes' ELSE 'no' END AS

ProfileDateWithin-Last60Days_soft,

CASE WHEN CAST ( (CAST(GETDATE() AS FLOAT) - CAST(ProfileDate AS FLOAT))
AS INT) < 91 THEN 'yes' ELSE 'no' END AS

ProfileDateWithin-Last90Days_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN
LastUsedTime ELSE NULL END AS LastUsedTime_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN
CAST(LastUsedTime AS DATE) ELSE NULL END AS LastUsedDate_soft,

-- SS2005 compatible:CASE WHEN LastUsedTime > CAST('1971-01-01' AS
smalldatetime) THEN CAST(FLOOR(CAST(LastUsedTime AS FLOAT)) AS
smalldatetime) ELSE NULL END AS LastUsedDate_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN
DATEDIFF(dd,LastUsedTime, C.ProfileDate) ELSE NULL END AS

LastUsed-DaysAgo_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN
DATEDIFF(mm,LastUsedTime, C.ProfileDate) ELSE NULL END AS
```

```
LastUsed-MonthsAgo_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN CASE
WHEN CAST ( (CAST(C.ProfileDate AS FLOAT) - CAST(LastUsedTime AS FLOAT))
AS INT) < 31 THEN 'yes' ELSE 'no' END ELSE NULL END AS

LastUsedTimeWithinLast30Days_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN CASE
WHEN CAST ( (CAST(C.ProfileDate AS FLOAT) - CAST(LastUsedTime AS FLOAT))
AS INT) < 61 THEN 'yes' ELSE 'no' END ELSE NULL END AS

LastUsedTimeWithinLast60Days_soft,

CASE WHEN LastUsedTime > CAST('1971-01-01' AS smalldatetime) THEN CASE
WHEN CAST ( (CAST(C.ProfileDate AS FLOAT) - CAST(LastUsedTime AS FLOAT))
AS INT) < 91 THEN 'yes' ELSE 'no' END ELSE NULL END AS

LastUsedTimeWithinLast90Days_soft,

Company AS Company_soft, Product AS Product_soft, Version6Part AS
Version6Part_soft, Version AS Version_soft,

CAST(dbo.VersionMajor(Version6Part) AS varchar(6)) AS Ver-sionMajor_soft,

CAST(dbo.VersionMajor(Version6Part) AS varchar(6)) + '.' +
CAST(dbo.VersionMinor(Version6Part) AS varchar(6)) AS VersionMa-
jorMinor_soft,

CAST(dbo.VersionMajor(Version6Part) AS varchar(6)) + '.' +
CAST(dbo.VersionMinor(Version6Part) AS varchar(6)) + '.' +
CAST(dbo.VersionRev(Version6Part) AS varchar(6)) AS VersionMajorMi-
norRev_soft,

FileDescription, Filename, FileSize,

dbo.VersionFormat(dbo.VersionCompose (ProductVersionNoMS,
ProductVersionNoLS)) AS ProductVersionNo,

dbo.VersionFormat(dbo.VersionCompose (FileVersionNoMS, FileVer-sionNoLS))
AS FileVersionNo,

CASE StartUp WHEN 1 THEN 'auto' ELSE 'user' END AS StartUp,

CASE InUse WHEN 1 THEN 'yes' WHEN 0 THEN 'no' ELSE NULL END AS InUse,

CASE ServiceStatus WHEN 1 THEN 'running' WHEN 0 THEN 'stopped' ELSE NULL
END AS ServiceStatus,

CASE ServiceStartType WHEN 2 THEN 'auto' WHEN 3 THEN 'manual' WHEN 4 THEN
'disabled' ELSE NULL END AS ServiceStartType,

LastUserDomain, LastUser, LastUserFullName,

CASE WHEN Is64Bit = 1 THEN 'yes' ELSE 'no' END AS Is64Bit,
```

```
CASE WHEN IsNativeToOs = 1 THEN 'yes' ELSE 'no' END AS IsNativeToOs,
MachineType,

ExeHeaderTypeLong AS ExeHeaderType, LoginUser,

S.Language AS Language_soft, S.LanguageName AS LanguageName_soft FROM

Software S INNER JOIN Computers C ON S.Id = C.Id;
```

c. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.

d. Click the **Continue** button.

3. Set Parameters

    a. Type: Rising Column

    b. Max Rows to Retrieve: 10000

    c. Specify Rising Column: ProfileDate_soft

    d. Timestamp: Current Index Time

    e. Output Timestamp Format: YYYY-MM-dd HH:mm:ss

    f. Execution Frequency: * * * *

    g. Click the **Continue** button.

4. Metadata

    a. Source: belmanage

    b. Sourcetype: belmanage_software

    c. Index: belmanage_software

    d. Select Resource Pool: local

    e. Click the **Save** button.

Belmanage_sw_changes:

DB Input: belmanage_sw_changes

1. Name Input

    a. Status: Enabled

    b. Name: belmanage_sw_changes

    c. Description: Software changes from BelManage

d. App: Splunk DB Connect v2

e. Connection: BelManage

f. Click the **Continue** button.

2. Choose and Preview Table

a. Make sure that **Simple Query Mode** is selected.

b. Catalog: BelMonitor82_1

c. Schema: dbo

d. Table: SoftwareHistoryReport

e. Max rows: 100

f. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.

g. Click the **Continue** button.

3. Set Parameters

a. Type: Rising Column

b. Max Rows to Retrieve: 100000

c. Specify Rising Column: ActionDate

d. Timestamp: Current Index Time

e. Output Timestamp Format: YYYY-MM-dd HH:mm:ss

f. Execution Frequency: */30 * * * *

g. Click the **Continue** button.

4. Metadata

a. Source: belmanage

b. Sourcetype: belmanage_sw_changes

c. Index: belmanage_sw_changes

d. Select Resource Pool: local

e. Click the **Save** button.

CA ITAM:

DB Input: ca-itam

1. Name Input
    a. Status: Enabled
    b. Name: ca-itam
    c. Description: Asset from CA ITAM software
    d. App: Splunk DB Connect v2
    e. Connection: ca-itam
    f. Click the **Continue** button.
2. Choose and Preview Table
    a. Make sure that **Advanced Query Mode** is selected.
    b. In the entry box type in the following SQL statement:

```
SELECT DISTINCT

aud_ca_owned_resource.resource_name,audit_model_uuid,audit_resource_cl
ass, audit_resource_subclass,
ca_owned_resource.own_resource_id,ca_owned_resource.mac_address,ca_own
ed_resource.ip_address,ca_owned_resource.host_name,ca_owned_resource.s
erial_number,ca_owned_resource.asset_source_uuid,ca_owned_resource.cre
ation_user,ca_owned_resource.creation_date,
al_aud_contact_view.first_name, al_aud_contact_view.middle_name,
al_aud_contact_view.last_name, al_aud_contact_view.pri_phone_number,
ca_owned_resource.last_update_date

FROM aud_ca_owned_resource INNER JOIN ca_owned_resource

ON aud_ca_owned_resource.resource_name=ca_owned_resource.resource_name

INNER JOIN al_aud_contact_view

ON   ca_owned_resource.resource_contact_uuid   =
al_aud_contact_view.contact_uuid
```

    c. Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.
    d. Click the **Continue** button.
3. Set Parameters
    a. Type: Rising Column

b. Max Rows to Retrieve: 1000

c. Specify Rising Column: last_update_date

d. Timestamp: Current Index Time

e. Output Timestamp Format: YYYY-MM-dd HH:mm:ss

f. Execution Frequency: */5 * * * *

g. Click the **Continue** button.

4. Metadata

   a. Source: ca-itam

   b. Sourcetype: ca-itam

   c. Index: ca_itam

      Note: the index name is **ca_itam** with an underscore. Splunk Enterprise does not accept dashes in index names.

   d. Select Resource Pool: local

   e. Click the **Save** button.

2.1.5 Lookup Table Files

Several lookup table files are necessary for this project. The lookup table files are in comma separated value format and contain data generated by reports that are used in other reports and dash-boards.

To create a lookup table file:

1. Open the Splunk Enterprise web page (https://172.16.5.55:8000) and go to the **Lookup table files** page.
2. Select **Settings** > **Lookups**.
3. Click **Lookup table files**.
4. Click the **New** button.

Create the following lookup table files:

- */opt/splunk/etc/apps/search/lookups/AssetRisk_Alltime.csv*
- */opt/splunk/etc/apps/search/lookups/AssetRisk_Last7days.csv*
- */opt/splunk/etc/apps/search/lookups/AssetRisk_Last24hours.csv*

- */opt/splunk/etc/apps/search/lookups/asset_value_table.csv*
- */opt/splunk/etc/apps/search/lookups/license_table.csv*
- */opt/splunk/etc/apps/search/lookups/updown*
- */opt/splunk/etc/apps/search/lookups/vun_rating_table.csv*

2.1.5.1 Splunk Enterprise Configuration Files

Splunk Enterprise configuration files can be found in the external file titled Splunk_Configuration_Files.tar.gz.

Configuration files are stored on Splunk Enterprise in the *$SPLUNK_HOME/etc/system/local* directory.

2.1.5.2 Splunk Enterprise Dashboards

Splunk Enterprise stores dashboards in XML format. All of the dashboards can be found in the external file titled Splunk_Dashboards.tar.gz.

Splunk Enterprise dashboard files are stored on Splunk Enterprise in the *$SPLUNK_HOME/etc/apps/search/local/data/ui/views directory*.

2.1.5.3 Restarting Splunk Enterprise After Configuration File Changes

When you make changes to Splunk Enterprise using configuration files, you might need to restart Splunk Enterprise for the changes to take effect. See the following link for details: http://docs.splunk.com/Documentation/Splunk/6.2.3/Admin/Configurationfilechangesthatrequirerestart.

3 Tier 2

3.1 AssetCentral

AssetCentral is an IT infrastructure management system that stores and displays information related to physical assets including location, make, model, and serial number. AssetCentral can help run an entire data center by monitoring weight, utilization, available space, heat and power distribution. AssetCentral is installed on a CentOS7 system.

3.1.1 How It's Used

In the FS ITAM build AssetCentral is used to provide physical asset location. AssetCentral provides the building, room and rack of an asset.

3.1.2 Virtual Machine Configuration

The virtual machine is configured with 1 network interface cards, 4 GB of RAM and 1 CPU cores.

3.1.3 Network Configuration

The management network interface card is configured as such:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.1.50
- Netmask: 255.255.255.0
- Gateway: 172.16.1.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.1.4 Installing AssetCentral

AssetCentral is installed on a hardened CentOS7 Linux system. AssetCentral requires PHP, Web Server (Apache) and MySQL database to be installed.

Table 3-1 Recommended Versions for AssetCentral – Tier 2

Vendor	Product	Version
RedHat	Enterprise Linux Server	6.4 (Santiago) (x86_64)
Apache	Web Server	httpd-2.2.15-26.el6.x86_64
mysql	Server	5.1.66
php		5.33 or higher

3.1.5 Installing MySQL (MariaDB)

```
# yum -y install mariadb-server mariadb

#systemctl start mariadb.service

#systemctl enable mariadb.service

# mysql_secure_installation
```

1. Answer the questions with the default answers while performing the mysql_secure_installation.
2. Create a database – assetcentral.

3. Create a user – assetcentral.
4. Grant all privileges to assetcentral user.

3.1.6 Installing Apache

```
# yum -y install httpd

#systemctl start httpd.service

#systemctl enable httpd.service

#firewall-cmd --permanent --zone=public --add-service=http

#firewall-cmd --permanent --zone=public --add-service=https

#firewall-cmd -reload
```

3.1.6.1 HTTP Configuration

1. Go to HTTPD root; normally (*/etc/httpd*).
2. Under the modules directory, make sure *libphp5.so* exists.
3. Change document root (webroot) as per environment in *httpd.conf*.

3.1.7 Installing PHP5

```
#yum -y install php

#systemctl restart httpd.service

#yum search php

#yum -y install php-mysql

#yum -y install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap curl curl-devel
```

1. Restart Apache:

   ```
   #systemctl restart httpd.service
   ```

3.1.8 Post Installation Tasks

1. Copy AssetCentral files and folders from previous install to the new webroot.
2. Under the location (*../assetcentral/application/config*), make necessary changes as per environment.

3.1.8.1 Sample

```
<?php defined('ASSET_CENTRAL') or die(''); define('AC_URL_SUBDIR', '/acprod');
define('AC_URL_SCRIPT','/index.php'); define('AC_URL_PARAM','go');
define('AC_URL_PREFIX',AC_URL_SUBDIR . AC_URL_SCRIPT.'?'

. AC_URL_PARAM . '='); define('AC_ERROR_REPORTING',E_ERROR);

//no slash at the end of this url define('URL_SITE','http://10.1.xx.xxx');
define('OS','NIX'); // *NIX WIN BSD MAC

//default database (read) define('DB_TYPE_READ','MYSQL');
define('DB_HOST_READ','127.0.0.1');

//usually leave this blank for MYSQL define('DB_PORT_READ','');
define('DB_USER_READ','assetcentral'); define('DB_PASS_READ','xxxxx');
define('DB_DATA_READ','asset_prod'); define('DB_PREFIX_READ','');
```

3.1.9 Database Update – Add a View

A database view was created on AssetCentral to gather all of the information required by the ITAM project in one place. This database view is accessed directly from Splunk Enterprise.

1. On the AssetCentral machine, open a terminal window and type the following command to enter the MySQL client application (you will be asked for the root password of the MySQL database):

   ```
   mysql assetcentral -u root -p
   ```

2. The following command will create the assetview view (from inside of the MySQL client application):

   ```
   create view assetview as

   select a.asset_id, a.rack_id, a.system_id, a.contact_id, a.serial_number,
   a.asset_tag, a.asset_name, a.ip_addr, a.description, a.title,
   a.internal_number, rack.rack_name, rack.room_id, rack.rack_type,
   rack.rack_notes, s.system_name, s.system_description,

   c.contact_name, c.phone_number, c.email_address, room.room_name, room.floor_id,
   floor.floor_name

   from assets a

   left join racks rack on a.rack_id = rack.rack_id left join systems s on
   a.system_id = s.system_id left join contacts c on a.contact_id = c.contact_id
   left join rooms room on rack.room_id = room.room_id

   left join floors floor on room.floor_id = floor.floor_id where a.asset_deleted
   != 1;
   ```

3. Create a new database user and assign that user privilges on the assetview view (from inside of the MySQL client application):

```
create new users and privileges inside mysql/mariadb create user
'asset_query'@'localhost';

set password for 'asset_query'@'localhost' = password('password'); grant select
on assetcentral.assetview to 'asset_query'@'localhost'; grant file on *.* to
'asset_query'@'localhost';
```

4. Ensure that the MySQL network port is listening and is allowed through the firewall. You must be root to run these commands.
5. To verify that MySQL is listening:

```
netstat -l |grep mysql
```

6. To allow MySQL through the firewalld firewall:

```
firewall-cmd -permanent -add-service=mysql firewall-cmd -reload
```

7. To make sure the firewall rule was added correctly:

```
firewall-cmd -list-services
```

3.1.10 Add Assets into AssetCentral

For AssetCentral to be of use, the end user must populate the system with all of the IT hardware to be tracked.

AssetCentral provides a manual method of adding one or two assets as well as an automated method of adding numerous assets that have been saved in a spreadsheet.

3.2 BelManage

BelManage is installed on a Windows Server 2012R2 system. BelManage gathers hardware and software information from computers on the network. BelManage gathers, stores, analyzes and displays the hardware and software information in a Web application. The BelMonitor client is installed on all computers in the network and automatically sends the BelManage server information on hardware and software changes.

3.2.1 How It's Used

The ITAM system is using BelManage for its data gathering, analysis and reporting features. BelManage reports on all software installed and all hardware configurations for every machine on the network that is running the BelMonitor client.

Splunk Enterprise connects to the BelManage database to pull data and provide further analysis and correlation.

3.2.2 Virtual Machine Configuration

The BelManage virtual machine is configured with 1 network interface card, 8 gigabytes (GB) of random access memory (RAM) and one central processing unit (CPU) core.

3.2.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Disabled
- IP Address: 172.16.2.71
- Netmask: 255.255.255.0
- Gateway: 172.16.2.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.2.4 Installing BelManage

Before installing BelManage, verify that your Windows Server 2012R2 system is installed correctly, updated and that the network is correctly configured and working. Additionally, you may have to disable or modify some security services, such as AppLocker, during the installation process.

BelManage is installed by running the BelManage server installation program (BelManageServer8.1.31.exe). Documentation is provided by Belarc at https://www.belarc.com/en/products_belmanage.

3.2.4.1 Prerequisites

Internet Information Server (IIS) 4.0 or later must be installed. The website below has detailed instructions on installing IIS: http://www.iis.net/learn/install/installing-iis-85/installing-iis-85-on-windows-server-2012-r2.

BelManage requires the following options: Static Content, Default Document, ASP Application Development, IIS Management Scripts and Tools, IIS 6 Metabase Compatibility, IIS 6 WMI Compatibility, and IIS 6 Scripting Tools.

MS SQL Express will be installed as part of the normal BelManage installation process.

Microsoft (MS) Structured Query Language (SQL) Server Management Studio is not required but is highly recommended. MS SQL Server Management Studio will make it easy to work on the BelManage database. Make sure you run MS SQL Server Management Studio as administrator or you will get

permission errors. Additional information can be found at: https://msdn.microsoft.com/en-us/library/ms174173.aspx.

3.2.4.2 Installation Procedure

3.2.4.2.1 Installing the BelManage Server

1. Open Windows File Explorer and navigate to where your BelManage installer is located.
2. Right-click on the BelManage installer file and select **Run as Administrator**.
3. Choose the default selections.

   Note: You will need to enter your BelManage license number during the installation process.

3.2.4.2.2 Installing the BelManage Client

The BelMonitor client must be installed on all devices that you wish to monitor.

The BelMonitor client should also be installed on the BelManage server if you wish to monitor.

1. The BelMonitor client can be downloaded directly from the BelManage server that was just installed: Point your web browser to your BelManage server (172.16.2.71):

   http://172.16.2.71/BelManage
2. Enter your login and password.
3. Select the **Getting Started** option on the left side of the page.
4. Select **Download your installable BelMonitor client** from the middle of the page.
5. Select the appropriate download – Windows, Linux, Mac OSX or Solaris.
6. Follow the steps in the relevant section.
   a. For Windows machines:
      i. Right-click the BelMonitor client and select **Run as Administrator**.
      ii. Then accept the default settings. The BelMonitor client will be installed and set to autorun when the system boots. There should be an icon in your system tray (right-side) that looks like a little green eye with eyelashes.
   b. For Linux machines:

      The BelMonitor client must be installed as the root user.

      i. To install the BelMonitorLinux client on Linux machines you must first install the 32-bit compatibility libraries. On Ubuntu the process is as follows:

```
apt-get install lib32stdc++6
```

ii. The BelMonitor client uses RPM (RedHat Package Manager) which can be installed as follows:

```
apt-get install rpm
```

iii. Make the BelMonitorLinux executable.

```
chmod a+x BelMonitorLinux
```

iv. Start the installation.

```
./BelMonitorLinux
```

The BelMonitor client should now be running and reporting to the BelManage server every 15 minutes (default setting).

3.2.5 Integration and Final Steps

1. Use MS SQL Server Studio Manager to create a database user for the Splunk Enterprise database connection. A new user must be created and be added to the correct database for the Splunk Enterprise integration to work.
2. Right-click MS SQL Server Studio Manager and select **Run as Administrator**.
3. Click **Connect** as the default settings should be correct:

   Server type: **Database Engine**

   Server name: **BELARC\BELMANAGE**

   Authentication: **Windows Authentication**
4. Once MS SQL Server Management Studio has logged in and started, create a new database user.
   a. Select **Security** > **Logins**.
   b. Right-click **Logins** and select **New User**.
   c. Enter a **Login name**.
   d. Select SQL Server authentication.
   e. Enter a password.
   f. Enter the password again in the **Confirm password** box.
   g. The Enforce password policy, **Enforce password expiration** and **User must change password at next login** should all reflect your organization's security rules.

Default database = **BelMonitor82_1**

Default language = **English**

5. Add the new user that you created in the preceding steps to the **BelMonitor82_1** database.
    a. Select **Databases** > **BelMonitor82_1** > **Security** > **Users**.
    b. Right-click **Users** and select **New User**.
    c. Enter a user name for the new user in the **User Name** and **Login Name** fields. They should be identical.

       Default schema = **db_datareader**

       Schemas owned by this user = **none selected**
    d. Database role membership: **BelMonitorReader** and **db_datareader** should be checked.
6. Turn on or re-enable any security settings that you might have changed, such as AppLocker.

3.3 Bro

Bro is an open-source network security monitor. Bro efficiently analyzes all network traffic and provides insight into clear text password use, cryptographic certificate errors, traffic to known bad sites, network flow, and file transfers.

3.3.1 How It's Used

In the FS ITAM build, Bro monitors all traffic traversing the DMZ. Bro has a dedicated network interface in promiscuous mode for sniffing/capturing traffic. This interface does not have an IP address assigned. Bro has a second network interface for management that is assigned IP address 172.16.0.20. When configuring Bro, make sure that Bro is sniffing/capturing on the correct network interface.

On the high-level architecture diagram, Bro is in Tier 2. Bro uses the Splunk Universal Forwarder to send logs to Splunk Enterprise. Some of the logs include files, Hypertext Transfer Protocol (HTTP) traffic, Kerberos authentications, Secure Socket Layer (SSL) traffic, x509 certificates seen, known hosts, DNS traffic, all connections, notices, and intelligence alerts.

3.3.2 Virtual Machine Configuration

The Bro virtual machine is configured with two network interface cards, 16 GB of RAM and four CPU cores.

3.3.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.0.20
- Netmask: 255.255.255.0
- Gateway: 172.16.0.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.3.4 Installing Bro

Bro is installed on a hardened Ubuntu 14.04 Linux system. Please download the latest source package from Bro and follow the instructions for installing from source. Installation was performed following the instruction from Bro at: https://www.bro.org/sphinx/install/index.html.

3.3.4.1 Installation Prerequisites

Bro requires the following libraries and tools to be installed before you begin:

- Libpcap (http://www.tcpdump.org)
- OpenSSL libraries (http://www.openssl.org)
- BIND8 library
- Libz
- Bash (for BroControl)
- Python (for BroControl)

To build Bro from source, the following additional dependencies are required:

- CMake 2.8 or greater (http://www.cmake.org)
- Make
- C/C++ compiler
- SWIG (http://www.swig.org)
- Bison (GNU Parser Generator)
- Flex (Fast Lexical Analyzer)

- Libpcap headers (http://www.tcpdump.org)
- OpenSSL headers (http://www.openssl.org)
- zlib headers
- Perl

3.3.4.1.1 For Debian/Ubuntu Linux systems:

1. It is always best to make sure your system is up-to-date by performing:

```
sudo apt-get update sudo apt-get upgrade
```

2. Then install the prerequisites:

```
sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libssl-dev
python-dev swig zlib1g-dev

sudo apt-get install libgeoip-dev

sudo apt-get install libgoogle-perftools-dev sudo apt-get install curl

sudo apt-get install git
```

3. Download and install Bro (this will install in */usr/local/bro*):

   Note: You need to be root to install Bro.

```
cd /usr/local

git clone https://github.com/actor-framework/actor-framework.git cd
/usr/local/actor-framework

./configure make

make test

make install
```

3.3.4.2 Installation Procedure

```
cd /usr/local

git clone --recursive git://git.bro.org/bro cd /usr/local/bro

./configure make

make install
```

1. Add Bro bin directory to your runtime path:

```
Edit .bashrc
```

2. Add the following line to the end of .bashrc:

```
EXPORT PATH=/usr/local/bro/bin:$PATH
```

3. Then:

```
source .bashrc
```

4. To start Bro the first time:

```
broctl deploy
```

5. To check the status of Bro:

```
broctl status
```

3.3.5 Installing Intelligence Gathering Software

1. Uses the mal-dnssearch package from Jon Schipp, which must be installed. The compiled version will be installed into */usr/local/bin/mal-dnssearch*.

```
cd /opt
git clone https://github.com/jonschipp/mal-dnssearch cd /opt/mal-dnssearch
sudo make
sudo make install
mkdir  /usr/local/bro_intel
cd /usr/local/bro_intel
```

2. Copy the update_intel.sh script into */usr/local/bro_intel*.

```
cp update_intel.sh /usr/local/bro_intel
chmod 700 /usr/local/bro_intel/update_intel.sh cd /usr/local/bro_intel
./update_intel.sh
```

You should now have several files usable with the Bro Intelligence Framework, including tor.intel, mandiant.intel, and alienvault.intel.

3. To have the script run automatically every day, add a link inside */etc/cron.daily*.

```
ln -s /usr/local/bro_intel/update_intel.sh
/etc/cron.daily/update_intel
```

3.3.6 Configuring Bro

To implement all of the functionality in the FS-ITAM use case build, the default Bro configurations will need to be modified. Please follow these steps to gain the same functionality.

1. Stop Bro:

```
broctl stop
```

2. Copy and edit node.cfg:

   ```
   cp /usr/local/bro/etc/node.cfg /usr/local/bro/etc/node.cfg.orig cp
   <source_dir>/node.cfg /usr/local/bro/etc
   ```

   Edit **node.cfg**, making sure that **interface=eth0** is the correct interface on which you will be sniffing/capturing traffic (NOT your management interface).

3. Edit networks.cfg:

   The networks.cfg file identifies all of your internal networks, so please list them all here. Below is our example:

   List of local networks in CIDR notation, optionally followed by a descriptive tag. For example:

   `10.0.0.0/8` or `fe80::/64` are valid prefixes.

   `10.0.0.0/8` Private IP space

   `192.168.0.0/16` Private IP space

   `172.16.0.0/16` Private IP space

4. Edit the local.bro file to reflect the settings you want:

   ```
   cp /usr/local/bro/share/bro/site/local.bro
   ```

   ```
   /usr/local/bro/share/bro/site/local.bro.orig
   ```

   ```
   cp <source_dir>/local.bro /usr/local/bro/share/bro/site/
   ```

5. Check changes, install changes, and restart Bro:

   ```
   broctl check broctl install broctl start broctl status
   ```

If everything goes right, you should start seeing log files in */usr/local/bro/logs/current*.

3.3.7 Installing Splunk Universal Forwarder

Note: You will need a Splunk account to download the Splunk Universal Forwarder. The Splunk Universal Forwarder is free and can be downloaded from: https://www.splunk.com/page/sign_up.

1. Download the Splunk Universal Forwarder from: http://www.splunk.com/en_us/download/universal-forwarder.html.

2. You want the latest version for `OS version 2.6+ kernel Linux distributions (64-bit).` Since this is installing on Ubuntu, select the file that ends in `.deb`. An example is:

   ```
   splunkforwader-6.2.5-272645-linux-2.6-amd64.deb
   ```

Detailed installation instructions can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Installation/InstallonLinux.

3. An abridged version follows:

   ```
   dpkg -i <splunk_package_name.deb>
   ```

   Example: `dpkg -i splunkforwader-6.2.5-272645-linux-2.6-amd64.deb`

4. This will install in */opt/splunkforwarder*:

   ```
   cd /opt/splunkforwarder/bin
   ./splunk start --accept-license
   ./splunk enable boot-start
   ```

5. Add forwarder:

   More information about adding a forwarder can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Forwarding/Deployanixdfmanually.

   ```
   cd /opt/splunkforwarder/bin
   ./splunk add forward-server loghost:9997 -auth admin:changme
   ```

3.3.8 Configuring Splunk Universal Forwarder

Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You will also need a copy of your certificate authority's public certificate.

1. Create a directory to hold your certificates:

   ```
   mkdir /opt/splunkforwarder/etc/certs
   ```

2. Copy your certificates in PEM format to */opt/splunkforwarder/etc/certs*:

   ```
   cp CAServerCert.pem /opt/splunkforwarder/etc/certs
   cp bro_worker1.pem /opt/splunkforwarder/etc/certs
   ```

3. Copy the Splunk Universal Forwarder configuration files:

   ```
   cp <server.conf> /opt/splunkforwarder/etc/system/local
   cp <inputs.conf> /opt/splunkforwarder/etc/system/local
   cp <outputs.conf> /opt/splunkforwarder/etc/system/local
   ```

4. Modify **server.conf** so that:

   ServerName=Bro is your hostname.

```
sslKeysfilePassword = <password for your private key>
```

5. Modify **outputs.conf** so that:

   Server = loghost:9997 is your correct Splunk Enterprise server/indexer and port.

```
sslPassword = <password of your certificate private key>
```

Note: This will be hashed and not clear text after a restart.

Inputs.conf should work, but you are free to modify it to include the Bro logs that you are interested in.

Note: dns.log, conn.log and http.log generate a significant volume of messages for Splunk Enterprise to index. Depending on the size of your Splunk Enterprise license, this data volume might cause license warnings or violations. See http://docs.splunk.com/Documentation/Splunk/6.2.3/Admin/Aboutlicenseviolations for more information.

3.3.9 Configurations and Scripts

Update_intel.sh should be placed in */usr/local/bro_intel*.

```
#!/bin/sh

# This script downloads and formats reputation data from the Internet and formats it
so that Bro can use it as intel data.

# Good idea to restart bro every now and then: broctl restart

# /usr/local/bro/share/bro/site/local.bro      looks for the files in this directory.

#

# Uses the mal-dnssearch package from Jon Schipp

# git clone https://github.com/jonschipp/mal-dnssearch

# cd mal-dnssearch

# sudo make install

#


cd /usr/local/bro_intel


# download and format the Mandiant APT info

mal-dnssearch -M mandiant -p | mal-dns2bro -T dns -s mandiant -n true >

/usr/local/bro_intel/mandiant.intel
```

```
# download and format TOR info

mal-dnssearch -M tor -p | mal-dns2bro -T ip -s tor -n true -u
http://rules.emergingthreats.net/open/suricata/rules/tor.rules >

/usr/local/bro_intel/tor.intel


# download and format Alienvault reputation info

mal-dnssearch -M alienvault -p | mal-dns2bro -T ip -s alienvault -n true >
/usr/local/bro_intel/alienvault.intel
```

/usr/local/bro/etc/node.cfg

```
# Example BroControl node configuration.

#

# This example has a standalone node ready to go except for possibly changing

# the sniffing interface.


# This is a complete standalone configuration. Most likely you will

# only need to change the interface. [bro]

type=standalone host=localhost interface=eth1

## Below is an example clustered configuration. If you use this,

## remove the [bro] node above.

#[manager]

#type=manager

#host=host1

#

#[proxy-1]

#type=proxy

#host=host1

#

#[worker-1]

#type=worker
```

```
#host=host2
#interface=eth0
#
#[worker-2]
#type=worker
#host=host3
#interface=eth0
#
#[worker-3]
#type=worker
#host=host4
#interface=eth0
```

/usr/local/bro/share/bro/site/local.bro

```
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# Capture plaintext passwords
redef HTTP::default_capture_password=T; redef FTP::default_capture_password=T;
#Hash all HTTP - for APT script
#redef HTTP::generate_md5=/.*/;

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Apply the default tuning scripts for common tuning settings.
@load tuning/defaults

# Load the scan detection script.
```

```
@load misc/scan

# Log some information about web applications being used by users
# on your network.
@load misc/app-stats

# Detect traceroute being run on the network.
@load misc/detect-traceroute

# Generate notices when vulnerable versions of software are discovered.
# The default is to only monitor software found in the address space defined
# as "local". Refer to the software framework's documentation for more
# information.
@load frameworks/software/vulnerable

# Detect software changing (e.g. attacker installing hacked SSHD).
@load frameworks/software/version-changes

# This adds signatures to detect cleartext forward and reverse windows shells.
@load-sigs frameworks/signatures/detect-windows-shells

# Uncomment the following line to begin receiving (by default hourly) emails
# containing all of your notices.
# redef Notice::policy += { [$action = Notice::ACTION_ALARM, $priority
= 0] };

# Load all of the scripts that detect software in various protocols.
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
```

```
@load protocols/http/software
# The detect-webapps script could possibly cause performance trouble when
# running on live traffic. Enable it cautiously.
#@load protocols/http/detect-webapps

# This script detects DNS results pointing toward your Site::local_nets
# where the name is not part of your local DNS zone and is being hosted
# externally. Requires that the Site::local_zones variable is defined.
@load protocols/dns/detect-external-names

# Load dhcp script to log known devices
@load protocols/dhcp/known-devices-and-hostnames

# Script to detect various activity in FTP sessions.
@load protocols/ftp/detect
# Scripts that do asset tracking.
@load protocols/conn/known-hosts
@load protocols/conn/known-services
@load protocols/ssl/known-certs

# This script enables SSL/TLS certificate validation.
@load protocols/ssl/validate-certs

# Check for SSL Heartbleed attack
@load protocols/ssl/heartbleed

# Check for weak keys
@load protocols/ssl/weak-keys

# Check for expiring certs
```

```
@load protocols/ssl/expiring-certs

# Uncomment the following line to check each SSL certificate hash against the ICSI
# certificate notary service; see http://notary.icsi.berkeley.edu .
@load protocols/ssl/notary

# If you have libGeoIP support built in, do some geographic detections and
# logging for SSH traffic.
@load protocols/ssh/geo-data
# Detect hosts doing SSH bruteforce attacks.
@load protocols/ssh/detect-bruteforcing
# Detect logins using "interesting" hostnames.
@load protocols/ssh/interesting-hostnames

# Detect SQL injection attacks.
@load protocols/http/detect-sqli

const feed_directory = "/usr/local/bro_intel";

# Intelligence framework
#@load policy/frameworks/intel/seen
#@load policy/frameworks/intel/do_notice
@load frameworks/intel/seen
@load frameworks/intel/do_notice

#@load policy/integration/collective-intel
#redef Intel::read_files += {
# feed_directory + "/mandiant.intel",
# feed_directory + "/tor.intel",
# feed_directory + "/alienvault.intel",
```

```
##"/usr/local/bro/share/bro/site/bad_domains.txt",

##"/somewhere/yourdata1.txt",

#};

redef Intel::read_files += { "/usr/local/bro_intel/mandiant.intel",
"/usr/local/bro_intel/tor.intel", "/usr/local/bro_intel/alienvault.intel",

};


#### Network File Handling ####


# Enable MD5 and SHA1 hashing for all files.

@load frameworks/files/hash-all-files


# Detect SHA1 sums in Team Cymru's Malware Hash Registry.

@load frameworks/files/detect-MHR


# Extract collected files

#@load extract_files


# this is the original malware_detect using perl and clamavd

#@load malware_detect


# can define this stuff here or in the site specific .bro scripts

#redef Communication::listen_port = 47777/tcp;

#redef Communication::nodes += {

# ["broping"] = [$host = 127.0.0.1, $class="broping", $events = /ping/,

$connect = F, $ssl = F],

# ["malware_detect"] = [$host = 127.0.0.1, $class="malware_detect",

$events = /malware_message/, $connect= F, $ssl = F]

#};


#@load malware1
```

```
#@load broccoli

#@load whitelisting

#@load broping


event bro_init() { Analyzer::disable_analyzer(Analyzer::ANALYZER_SYSLOG);

}


#event bro_init()

# {

# local f = Log::get_filter(Notice::ALARM_LOG, "alarm-mail");

# f$interv = 1day;

# Log::add_filter(Notice::ALARM_LOG, f);

# }
```

/opt/splunkforwarder/etc/system/local/server.conf

```
[sslConfig]

sslKeysfilePassword = $1$2OJs1XSIp3Un


[lmpool:auto_generated_pool_forwarder] description = auto_generated_pool_forwarder
quota = MAX

slaves = *

stack_id = forwarder [lmpool:auto_generated_pool_free] description =
auto_generated_pool_free quota = MAX

slaves = * stack_id = free

[general]

pass4SymmKey = $1$j644iTHO7Ccn serverName = bro
```

/opt/splunkforwarder/etc/system/local/inputs.conf

```
[default] host = bro

sourcetype=BroLogs index=bro
```

```
[monitor:///usr/local/bro/logs/current/notice.log] sourcetype=bro_notice
[monitor:///usr/local/bro/logs/current/weird.log] sourcetype=bro_weird
[monitor:///usr/local/bro/logs/current/ssl.log] sourcetype=bro_ssl
[monitor:///usr/local/bro/logs/current/ssh.log] sourcetype=bro_ssh
[monitor:///usr/local/bro/logs/current/software.log] sourcetype=bro_software
[monitor:///usr/local/bro/logs/current/intel.log] sourcetype=bro_intel
[monitor:///usr/local/bro/logs/current/http.log] sourcetype=bro_http
[monitor:///usr/local/bro/logs/current/conn.log] sourcetype=bro_conn
[monitor:///usr/local/bro/logs/current/x509.log] sourcetype=bro_x509

[monitor:///usr/local/bro/logs/current/dns.log] sourcetype=bro_dns

#[monitor:///usr/local/bro/logs/current/*.log]

#host=bro-worker1

#sourcetype=BroLogs

#index=bro


#[monitor:///opt/splunkforwarder/var/log/splunk/splunkd.log]
```

/opt/splunkforwarder/etc/system/local/outputs.conf

```
[tcpout]

defaultGroup = splunkssl


[tcpout:splunkssl] server = loghost:9997 compressed = true

sslVerifyServerCert = false

sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem sslCertPath =
$SPLUNK_HOME/etc/certs/bro-worker1.pem sslPassword = $1$23DtXas9IZD8
```

3.4 CA Technologies IT Asset Manager

CA Technologies IT Asset Manager (CA ITAM) allows you to holistically manage IT hardware assets, from planning and requisition to retirement and disposal. This solution helps to rein in IT costs and boost return on investment by identifying underutilized hardware assets, improving hardware usage profiles, managing contracts and usage patterns, and giving you a thorough understanding of the true costs of your IT asset base.

3.4.1 How It's Used

In the FS ITAM build, CA ITAM is used to track hardware assets from requisition to disposal. Data collected during this task will be analyzed and used to notify an administrator of a change in the network

architecture. When a new hardware asset is received, an administrator will enter into the database information that includes, but is not limited to, the asset name, host name, operating system, serial number, owner, location, mac address and IP address. The data is then stored for retrieval by Splunk Enterprise. For this build, the CA ITAM database is pre-loaded with data from machines being used throughout the ITAM architecture. The Tier 1 ITAM server is connected to the CA ITAM database to query data stored in the CA ITAM resource tables.

3.4.2 Virtual Machine Configuration

The CA ITAM virtual machine is configured with one network interface cards, 16 GB of RAM, two CPU cores, a 40 GB hard drive, and another 100 GB hard drive. The 100 GB of hard drive space is very important for this machine.

3.4.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Disabled
- IP Address: 172.16.3.92
- Netmask: 255.255.255.0
- Gateway: 172.16.3.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.4.4 Installing CA ITAM

CA ITAM is installed on a clean 64-bit Windows Server 2012 R2 image with default Windows firewall configurations. Installation configurations are default for this build and are documented online by CA Technologies. CA Technologies installation guidelines can be found online at the following URL: https://support.ca.com/cadocs/0/CA%20IT%20Asset%20Manager%2012%208-ENU/Bookshelf_Files/PDF/APM_Impl_ENU.pdf.

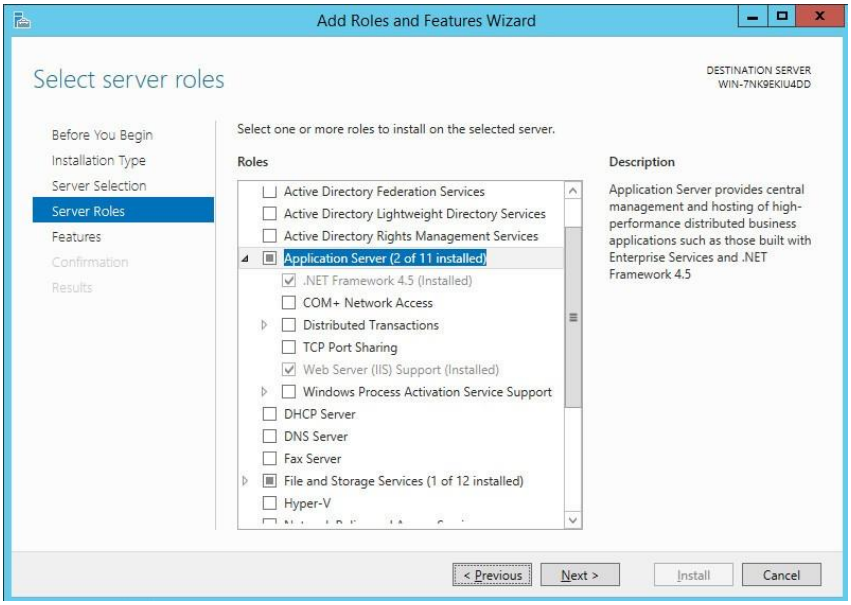
Prerequisites for this build are as follows:

- Java 7 JRE (32-bit)
  - Set the JAVA_HOME variable
- SQL Server 2012 with
  - Database Engine

- Backwards Compatibility
- Client Connectivity
- Management tools
- Used mixed authentication as the authentication method

- NET Framework 3.5
- NET Framework 4.5
  - Select ASP.NET
- IIS

  Note: Make sure the application server supports the IIS under add roles and features



- CA Business Intelligence Server
- CA Embedded Entitlements Manager

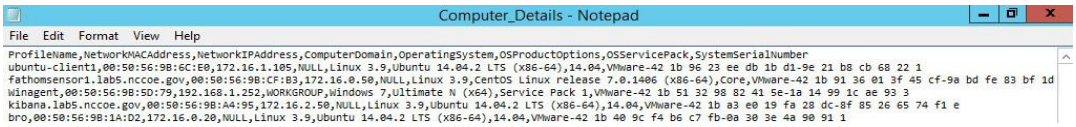
3.4.5 Configurations

Once installed, the data importer engine is used to import data from a .CSV file into the MDB. The file is obtained from the Belarc Server, which exports data into a .CSV file. Then the file is copied onto the CA ITAM Server.

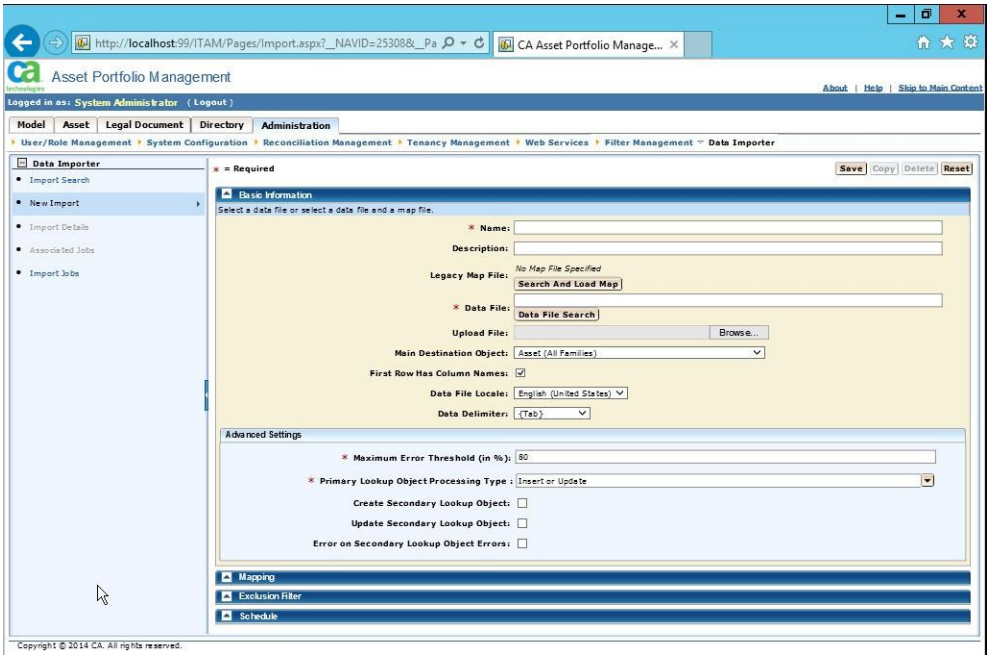
1. Save the .CSV file in *\CA\ITAM\Storage\Common Store\Import*.

The file contains data with the following field names: ProfileName, NetworkMACAddress, ComputerDomain, OperatingSystem, OSProductOptions, OSServicePack, SystemSerialNumber.

A snippet of the .CSV file is displayed in the following figure:



2. Open the CA Data Importer by logging into CA ITAM with administrator privileges and navigate to **Administration** > **Data Importer** > **New Import**.



3. In the **Administration** tab, specify these settings:
    a. **Name**: <Name>
    b. **Data File**: <filename>
    c. **Main Destination Object**: **Asset(Computer)**
    d. Select **First Row Has Column Names**
    e. **Data File Locale**: **English (United States)**
    f. **Data Delimiter**: **{Comma}**
4. In **Advanced Settings**, select all three check boxes.

5. Save the import.
6. Under **Mapping**, select **Load Source Fields**.
7. Map the **Source Fields** to the **Destination Fields** using the following rules:
    a. **Computer domain** = **Asset.Host Name**
    b. **NetworkIPAddress** = **Asset.IP Address**
    c. **NetworkMACAddress** = **Asset.MAC Address**
    d. **OperatingSystem** = **Asset.Model.Model Name**
    e. **OSProductOptions** = **Asset.Asset Type Hierarchy.Class.Value**
    f. **OSServicePack** = **Asset.Asset Type Hierarchy.Subclass.Value**
    g. **ProfileName** = **Asset.Asset Name**
    h. **SystemSerialNumber** = **Asset.Serial Number**
8. Under the **Schedule**, upload the .CSV data file again and **Submit**. Make sure that the data import service is running.
9. Check the status of the job under **Import Jobs**.
10. Use the data stored in the MDB to run a query through the Splunk DB Connection (See Section 2.1.1, Splunk Enterprise, to configure.).
11. Query is as follows:

```
SELECT DISTINCT

aud_ca_owned_resource.resource_name,audit_mode_uuid,audit_resource_class,au
dit_resource_subclass,ca_owned_resource.own_resource_id,ca_owned_resource.m
ac_address,ca_owned_resource.ip_address,ca_owned_resource.host_name,ca_owne
d_resource.serial_number,ca_owned_resource.asset_source_uuid,ca_owned_resou
rce.creation_user,ca_owned_resource.creation_date

FROM aud_ca_owned_resource INNER JOIN ca_owned_resource

ON aud_ca_owned_resource.resource_name = ca_owned_resource.resource_name
```

3.5 Fathom Sensor from RedJack

Fathom Sensor passively scans network traffic analyzing and reporting on netflow and cleartext banner information crossing the network. DNS and http traffic are also analyzed. Fathom Sensor detects anomalies on the network by analyzing these data streams.

3.5.1 How It's Used

Fathom Sensor passively monitors, captures, and optionally forwards summarized network traffic to its service running on the Amazon AWS cloud. The data on the Amazon server is then analyzed by RedJack to detect anomalies. The data is also aggregated with data from other organizations to detect attack trends.

3.5.2 Virtual Machine Configuration

The FathomSensor1 virtual machine is configured with 2 network interface cards (1 card for access and 1 for sniffing traffic), 16 GB of RAM, 1 CPU cores and 16 GB of hard drive space.

3.5.3 Network Configuration

The management network interface card is configured as such:

- IPv4 Manual
- IPv6 Disabled
- IP Address: 172.16.0.50
- No IP address for the second network interface card Netmask: 255.255.255.0
- Gateway: 172.16.0.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.5.4 Installing Fathom Sensor

3.5.4.1 VM Deployments

This document will track the best-practices for provisioning, installing, and deploying the fathom-sensor in a virtual machine (VM).

3.5.4.2 Requirements

Fathom Sensor VM requirements vary based on the size, traffic volume, and complexity of the network. The most important factor for performance is RAM. A small business network of <50 devices might be safe on a VM with **16GB RAM**, where as a large enterprise gateway may require **32-64GB RAM** and dedicated hardware.

Fathom Sensor will continue to operate in a degraded state if it becomes resource starved, but it is best to start high.

3.5.4.3 Configure the VM

When creating the virtual machine, create two network interfaces, one for management, and one for monitoring. The monitoring interface must be set to promiscuous mode.

Instructions vary by VM platform and host, but this is covered here:

- ESX – [KB: 1004099](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1004099)
- Linux – [KB: 287](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=287)
- Fusion – Password prompt can be disabled under Preferences > Network.

3.5.4.4 Install CentOS 7 Minimal

Our reference platform is CentOS 7 x64. Install (using USB or ISO or whatever) a minimal install.

3.5.4.5 Configure OS

Note: The following is based on the aforementioned VM with 2 NICs, one management NIC (eno1...) and one monitoring NIC (eno2...).

Before beginning the configuration, you should collect the following information: IP/Netmask/Gateway for management interface. This will need Internet access on port **80** and **443**. Optionally, you can use DHCP.

172.16.0.50

DNS server. This can be a local (to the customer) DNS server, or public (8.8.8.8, 4.2.2.4), however the latter will require firewall rules. Optionally, DHCP can configure this, however it needs to be set as above.

172.16.1.20, 172.16.1.21

NTP Server. This can be a local (to the customer), or a public (0.centos.pool.ntp.org) server, however the latter will require firewall rules.

172.16.0.11

NICs can be obscurely named, especially in VM environments. List all interfaces with: `# ip addr`.

3.5.4.6 Configure the Management Network with a Static IP

```
# /etc/sysconfig/network-scripts/ifcfg-eno1
```

```
BOOTPROTO=static IPADDR=172.16.0.50 NETMASK=255.255.255.0 ONBOOT=yes
```

3.5.4.7 Configure the Monitoring Interface Without an IP

1. Configure the monitoring interface without an IP:

```
# /etc/sysconfig/network-scripts/ifcfg-eno2
BOOTPROTO=static ONBOOT=yes
```

2. Disable IPv6 autoconfiguration on the monitoring interface:

```
# sysctl -w net.ipv6.conf.eno2.disable_ipv6=1
```

3.5.4.8 Configure DNS

```
# vi /etc/resolv.conf
search lab5.nccoe.gov
nameserver 172.16.1.20
nameserver 172.16.1.21
```

3.5.4.9 Set the Hostname

```
# hostnamectl set-hostname fathomsensor1
# vi /etc/hosts
127.0.0.1 localhost
172.16.0.50 fathomsensor1
```

3.5.4.10 Adjust the Packages

1. Not required, but if you are planning to install VMWare Tools, you need

```
$ yum install perl net-tools gcc kernel-devel
```

2. Install basic tools

```
$ yum install ntp bash-completion net-tools wget curl lsof tcpdump psmisc
```

3.5.4.11 Remove Unnecessary Packages

```
$ systemctl stop postfix chronyd avahi-daemon.socket avahi-daemon.service
$ systemctl disable avahi-daemon.socket avahi-daemon.service
$ yum remove postfix chronyd avahi-autoipd avahi-libs avahi
```

3.5.4.12 Disable SELinux

```
# vi /etc/selinux/config
```

```
SELINUX=permissive
```

3.5.4.13 Limit SSH

```
# vi /etc/ssh/sshd_config
```

```
ListenAddress 172.16.0.50
```

3.5.4.14 NTP

Some VM platforms or configurations will provide a synchronized system clock. If you know this is the case, you can skip this section.

```
#vi /etc/ntp.conf
```

```
driftfile /var/lib/ntp/drift

restrict default nomodify notrap nopeer noquery

server 0.centos.pool.ntp.org iburst

server 1.centos.pool.ntp.org iburst

server 2.centos.pool.ntp.org iburst

server 3.centos.pool.ntp.org iburst

includefile  /etc/ntp/crypto/pw

keys /etc/ntp/keys

disable monitor
```

1. Limit NTP to only listening on the management interface:

   ```
   #vi    /etc/sysconfig/ntpd
   ```

   ```
   OPTIONS="-g -I eno1 -I 172.16.0.50"
   ```

2. Before deployment, make sure the hardware clock is set to something reasonably correct:

   ```
   $ ntpdate 172.16.0.11
   ```

   ```
   $ hwclock -w
   ```

3. Set NTP to start:

   ```
   $ systemctl enable ntpd
   ```

   ```
   $ systemctl start ntpd
   ```

3.5.4.15 CollectD

We use collectd to keep track of system (and fathom metrics) and report those metrics back to customer-metrics.redjack.com every 60 seconds.

1. First, we need to install it from EPEL (version number will change):

```
#yum install
http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarc h.rpm
#yum install collectd collectd-netlink
```

2. Then install the collectd config file, which will have a URL specific for this sensor, which we've been using as the sensor UUID.

3. Then enable collectd:

```
$ systemctl enable collectd
$ systemctl start collectd
```

3.5.4.16 Install Fathom-Sensor

1. First install all the sensor RPMs:

```
$ sudo yum install *.rpm
```

2. Assuming that you have built a sensor config with `fathom-admin`:

```
$ cp fathom-sensor1.conf /etc/fathom/fathom-sensor.conf
$ chown fathom:fathom /etc/fathom/fathom-sensor.conf
$ chmod 600 /etc/fathom/fathom-sensor.conf
```

3. Edit the sensor config to make sure that it is listening to the correct device:

```
# vi /etc/fathom/fathom-sensor.conf
FATHOM_SENSOR_NETWORK_DEVICE=eno2
```

3.5.4.17 Update Dynamic Run-Time Bindings

1. Update dynamic run-time bindings because sometimes it needs it:

```
$ ldconfig
```

2. Then enable the “dedicated” version of the sensor. This has some hardcore properties in it that will reboot if there are continual problems:

```
$ systemctl enable fathom-sensor-dedicated
$ systemctl start fathom-sensor-dedicated
```

3.5.4.18 Install and Configure Amazon S3 Command Line Tools Using PIP

1. Go to http://docs.aws.amazon.com/cli/latest/userguide/installing.html.
2. Verify that you have at least Python 2.7:

   ```
   $ python -version
   ```

3. Download the pip installation script:

   ```
   $ curl -O https://bootstrap.pypa.io/get-pip.py
   ```

4. Run the pip installation script:

   ```
   $ sudo python get-pip.py
   ```

5. Install the AWS CLI:

   ```
   $ sudo pip install awscli
   ```

3.5.4.19 Configure AWS CLI

1. Configure AWS CLI:

   ```
   #aws configure
   ```

2. You will get the data to configure AWS CLI from the fathom-sensor.conf file. We want the data in JSON format.

   AWS Access Key ID = FATHOM_SENSOR_AWS_ACCESS_KEY

   AWS Secret Access Key = FATHOM_SENSOR_AWS_SECRET_KEY Default region Name = None

   Default output format = json

3. Create a directory to save the files gathered from Amazon AWS:

   ```
   #mkdir /opt/fathom-sync
   ```

4. Create a script to sync data with the Amazon AWS:

   ```
   #vi /usr/local/bin/fathom-sync.sh
   ```

5. Copy the following lines into fathom-sync.sh. Replace <SENSOR ID> with your individual sensor ID.

   #!/bin/sh

   /bin/aws s3 sync s3://fathom-pipeline/json/nccoe/<SENSOR ID>/ /opt/fathom-sync

6. Make the script executable:

   ```
   #chmod +x /usr/local/bin/fathom-sync
   ```

7. Make the script run every hour by placing a link in */etc/cron.hourly*:

```
#cd /etc/cron.hourly

#ln -s /usr/local/bin/fathom-sync.sh /etc/cron.hourly/fathom-sync
```

3.5.5 Installing Splunk Universal Forwarder

Note: You will need a Splunk account to download the Splunk Universal Forwarder. It is free and can be setup at: https://www.splunk.com/page/sign_up.

1. Download the Splunk Universal Forwarder from: http://www.splunk.com/en_us/download/universal-forwarder.html.
2. Use the latest version for `OS version 2.6+ kernel Linux distributions (64-bit)`. Since this is installing on Ubuntu select the file that ends in `.deb`. An example is:

   ```
   splunkforwader-6.2.5-272645-linux-2.6-amd64.deb
   ```

   Detailed installation instructions can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Installation/InstallonLinux.
3. An abridged version follows:

   ```
   rpm -i <splunk_package_name.deb>
   ```

   Example: `rpm -i splunkforwader-6.2.4-271043-linux-2.6-x86_64.rpm`
4. This will install in */opt/splunkforwarder*:

   ```
   cd /opt/splunkforwarder/bin

   ./splunk start --accept-license

   ./splunk enable boot-start
   ```
5. Add forwarder:

   More info about adding a forwarder can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Forwarding/Deployanixdfmanually.

   ```
   cd /opt/splunkforwarder/bin

   ./splunk add forward-server loghost:9997 -auth admin:changme
   ```

3.5.6 Configuring Splunk Universal Forwarder

Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You will also need a copy of your certificate authority's public certificate.

1. Create a directory to hold your certificates:

   ```
   mkdir /opt/splunkforwarder/etc/certs
   ```

2. Copy your certificates in PEM format to */opt/splunkforwarder/etc/certs*:

   ```
   cp CAServerCert.pem /opt/splunkforwarder/etc/certs
   ```

   ```
   cp fathomsensor1.lab5.nccoe.pem /opt/splunkforwarder/etc/certs
   ```

3. Copy Splunk Universal Forwarder configuration files:

   ```
   cp <server.conf> /opt/splunkforwarder/etc/system/local
   ```

   ```
   cp <inputs.conf> /opt/splunkforwarder/etc/system/local
   ```

   ```
   cp <outputs.conf> /opt/splunkforwarder/etc/system/local
   ```

4. Modify **server.conf** so that:

   ServerName=Bro is your hostname.

   ```
   sslKeysfilePassword = <password for your private key>
   ```

5. Modify **outputs.conf** so that:

   Server = loghost:9997 is your correct Splunk Enterprise server/indexer and port.

   ```
   sslPassword = <password of your certificate private key>
   ```

Note: this will be hashed and not clear text after a restart.

3.5.7 Helpful Commands and Information

The following commands could prove useful when working with Amazon Web Servers S3. Replace <SENSOR ID> with your individual sensor ID.

1. List your sensor(s):

   ```
   aws s3 ls s3://fathom-pipeline/json/nccoe/
   ```

2. List data types for a sensor:

   ```
   aws s3 ls s3://fathom-pipeline/json/nccoe/<SENSOR ID>/
   ```

3. List dates for the client-banner data type:

```
aws s3 ls s3://fathom-pipeline/json/nccoe/<SENSOR ID>/client-banner/
```

4. List individual JSON files on that date:

```
aws s3 ls
s3://fathom-pipeline/json/nccoe/<SENSOR ID>/client-banner/20150604/
```

5. The following command will convert from a certificate in PKCS12 format to PEM format:

```
openssl pkcs12 -in certificate.pfx -out certificate.cer -nodes
```

3.5.8 Configurations and Scripts

/opt/splunkforwarder/etc/system/local/server.conf

```
[sslConfig]

sslKeysfilePassword = $1$2OJs1XSIp3Un


[lmpool:auto_generated_pool_forwarder] description = auto_generated_pool_forwarder
quota = MAX

slaves = *

stack_id = forwarder


[lmpool:auto_generated_pool_free] description = auto_generated_pool_free quota = MAX

slaves = * stack_id = free

[general]

pass4SymmKey  =  $1$j644iTHO7Ccn serverName = fathomsensor1.lab5.nccoe.gov
```

/opt/splunkforwarder/etc/system/local/inputs.conf

```
[default]

host = fathomsensor1.lab5.nccoe.gov sourcetype=fathomsensor index=fathom

[monitor:///opt/fathom-sync/*/client-banner*]

/opt/splunkforwarder/etc/system/local/outputs.conf [tcpout]

defaultGroup = splunkssl
```

```
[tcpout:splunkssl] server = loghost:9997 compressed = true

sslVerifyServerCert = false

sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem

sslCertPath = $SPLUNK_HOME/etc/certs/fathomsensor1.lab5.nccoe.gov.pem sslPassword =
$1$23DtXas9IZD8
```

3.6 OpenVAS

OpenVAS is an open-source network vulnerability scanner and manager. OpenVAS runs customizable scans and generates reports in multiple formats. OpenVAS is also a framework, and additional tools can be added to it.

3.6.1 How It's Used

In the FS ITAM build, OpenVAS automatically runs vulnerability scans on all systems connected to the network. Every machine is scanned at least once a week. OpenVAS collects the information, stores it in a database, and creates reports. OpenVAS can also download the latest vulnerabilities along with their CVE and NVT information.

On the high-level architecture diagram, OpenVAS is in Tier 2. OpenVAS utilizes the Splunk Universal Forwarder to send reports to Splunk Enterprise. Information is extracted from the OpenVAS database every hour, and any new records are forwarded to Splunk Enterprise. Splunk Enterprise uses the information from OpenVAS to provide context to analysts regarding the security of individual systems as well as aggregating statistics to show the overall organizational security posture.

3.6.2 Virtual Machine Configuration

The OpenVAS virtual machine is configured with one network interface card, 16 GB of RAM and four CPU cores.

3.6.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.2.33
- Netmask: 255.255.255.0
- Gateway: 172.16.2.11

- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

https://www.digitalocean.com/community/tutorials/how-to-use-openvas-to-audit-the-securit y-of-remote-systems-on-ubuntu-12-04

3.6.4 Installation Prerequisites

```
sudo apt-get update

sudo apt-get install python-software-properties

sudo apt-get install sqlite3 xsltproc texlive-latex-base

texlive-latex-extra texlive-latex-recommended htmldoc alien rpm nsis fakeroot
```

3.6.5 Installing OpenVAS

OpenVAS is installed on a hardened Ubuntu 14.04 Linux system. Please download the latest source package from OpenVAS and follow the instructions for installing from source.

Installation was performed following the instructions gathered from the following web sites:

- http://www.openvas.org/
- https://www.digitalocean.com/community/tutorials/how-to-use-openvas-to-audit-the-securit y-of-remote-systems-on-ubuntu-12-04
- https://launchpad.net/~openvas/+archive/ubuntu/openvas6

1. Add new file in /etc/apt/sources.list.d/openvas-openvas6-trusty.list:

   ```
   deb http://ppa.launchpad.net/openvas/openvas6/ubuntu precise main

   deb-src http://ppa.launchpad.net/openvas/openvas6/ubuntu precise main

   sudo apt-get install openvas-manager openvas-scanner

   openvas-administrator openvas-cli greenbone-security-assistant sudo openvas-
   mkcert
   ```

2. Answer the questions for the new certificate:

   ```
   sudo openvas-mkcert-client -n om -i
   ```

3. Download and build the vulnerability database:

   ```
   sudo openvas-nvt-sync
   ```

4. Stop the services:

```
sudo service openvas-manager stop

sudo service openvas-scanner stop
```

5. Start the scanner application (this will download and sync a lot of data):

```
sudo openvassd
```

6. Rebuild the database:

```
sudo openvasmd --rebuild
```

7. Download and sync SCAP data:

```
sudo openvas-scapdata-sync
```

8. Download and sync cert data:

```
sudo openvas-certdata-sync
```

Note: You will most likely get an error because the Ubuntu package is missing some files.

9. The following commands will get the files from the Fedora package and install them in the correct location:

```
cd

wget http://www6.atomicorp.com/channels/atomic/fedora/18/i386/RPMS/openvas-
manager-5.0.8-27.fc18.art.i686.rpm

sudo apt-get install rpm2cpio

rpm2cpio openvas* | cpio -div

sudo mkdir /usr/share/openvas/cert

sudo cp ./usr/share/openvas/cert/* /usr/share/openvas/cert
```

10. Now sync the certs, and everything should work:

```
sudo openvas-certdata-sync
```

11. Add user and permissions:

```
sudo openvasad -c add_user -n admin -r Admin
```

12. Edit the following file and insert your OpenVAS IP address:

```
sudo nano /etc/default/greenbone-security-assistant
```

13. Start up the services:

```
sudo killall openvassd
```

```
sudo service openvas-scanner start sudo service openvas-manager start
sudo service openvas-administrator restart
sudo service greenbone-security-assistant restart
```

14. Enable start up at boot time:

```
sudo update-rc.d openvas-scanner enable 2 3 4 5
sudo update-rc.d openvas-manager enable 2 3 4 5
sudo update-rc.d openvas-administrator enable 2 3 4 5
sudo update-rc.d greenbone-security-assistant enable 2 3 4 5
```

15. Try it out. Point your web browser to:

    https://localhost:9392

    https://172.16.2.33:9292

    Note: It must be https.

3.6.6 Configuring OpenVAS

Full user documentation can be found at: http://docs.greenbone.net/index.html#user_documentation.

OpenVAS supports immediate scans and scheduled scans. Scheduled scans enable full automation of scanning and reporting.

1. Set up schedules:
    a. **Configuration** > **Schedules**.
    b. Click the **Star** icon to create a new schedule.
    c. Create a schedule for every day of the week. Example: Monday scans - every day at 21:00.
    d. Do the same for the other 6 days of the week.
2. Setup targets:

    A target is an individual system to scan or a range of systems to scan. In the FS-ITAM lab a separate target was configured for each subnet.

    a. **Configuration** > **Targets**.

b. Click the **Star** icon to create a new target.

Example:

Name: Network Security.

Hosts: 172.16.2.1-172.16.2.254.

Comment: Network Security systems.

c. Click **Create Target** button to save.

3. Set up tasks:

A task is something that is done to a target. So we need to setup a scan on each target.

a. **Scan Management** > **New Task**.

Name: **Scan DMZ**

Comment: **Scan the DMZ systems**

Scan Config: **Full and fast**

Scan Targets: **DMZ** (this is why the target must exist before the task).

Schedule: **Tuesday scan** (this is why the schedule must exist before the task).

b. Click the **Create Task** button to save.

c. Continue adding all of the tasks that you need - one for each target.

3.6.6.1 Openvas_results.py

The *openvas_results.py* is a Python script that accesses the OpenVAS Sqlite3 database, extracts interesting values and then writes those to files in CSV and JSON formats.

The *openvas_results.py* is run by cron every hour to check for new results from OpenVAS scans.

The Splunk Universal Forwarder checks the CSV file written by openvas_results.py for any changes and sends those to the Splunk Enterprise server/indexer.

1. Place *openvas_results.py* in /root and make sure that it is executable:

```
cp <openvas_results.py> /root
chmod +x /root/openvas_results.py
```

2. Create a symbolic link in /etc/cron.hourly so that *openvas_results.py* runs every hour:

```
ln -s /root/openvas_results.py /etc/cron.daily/openvas_results
```

3.6.7 Installing Splunk Universal Forwarder

Note: You will need a Splunk account to download the Splunk Universal Forwarder. It is free and can be set up at: https://www.splunk.com/page/sign_up.

1. Download the Splunk Universal Forwarder from: http://www.splunk.com/en_us/download/universal-forwarder.html.
2. You want the latest version for `OS version 2.6+ kernel Linux distributions (64-bit)`. Since this is installing on Ubuntu, select the file that ends in `.deb`. An example is:

   ```
   splunkforwader-6.2.5-272645-linux-2.6-amd64.deb
   ```

   Detailed installation instructions can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Installation/InstallonLinux.
3. An abridged version follows:

   ```
   dpkg -i <splunk_package_name.deb>
   ```

   Example: `dpkg -i splunkforwader-6.2.5-272645-linux-2.6-amd64.deb`
4. This will install in */opt/splunkforwarder*:

   ```
   cd /opt/splunkforwarder/bin
   ./splunk start --accept-license
   ./splunk enable boot-start
   ```
5. Add forwarder:

   More information about adding a forwarder can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Forwarding/Deployanixdfmanually.

   ```
   cd /opt/splunkforwarder/bin
   ./splunk add forward-server loghost:9997 -auth admin:changme
   ```

3.6.8 Configuring Splunk Universal Forwarder

Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You will also need a copy of your certificate authority's public certificate.

1. Create a directory to hold your certificates:

   ```
   mkdir /opt/splunkforwarder/etc/certs
   ```

2. Copy your certificates in PEM format to */opt/splunkforwarder/etc/certs*:

```
cp CAServerCert.pem /opt/splunkforwarder/etc/certs

cp bro_worker1.pem /opt/splunkforwarder/etc/certs
```

3. Copy Splunk Universal Forwarder configuration files:

```
cp <server.conf> /opt/splunkforwarder/etc/system/local

cp <inputs.conf> /opt/splunkforwarder/etc/system/local

cp <outputs.conf> /opt/splunkforwarder/etc/system/local
```

4. Modify **server.conf** so that:

   ServerName=openvascd is your hostname.

```
sslKeysfilePassword = <password for your private key>
```

5. Modify **outputs.conf** so that:

   Server = loghost:9997 is your correct Splunk Enterprise server/indexer and port.

```
sslPassword = <password of your certificate private key>
```

Note: This will be hashed and not clear text after a restart.

Inputs.conf should work, but you are free to modify it to include the OpenVAS logs that you are interested in.

3.6.9 Configurations and Scripts

/root/openvas_results.py

```
#! /usr/bin/env python

#

# Gathers info from OpenVAS database and writes it to a CSV and JSON for
SplunkForwarder

#

import os import os.path import sys

from time import sleep

from datetime import datetime import ntpath

import errno import sqlite3 import csv import json

# Global variables and configs
```

```
# SQLITE3 database file

file_db = "/var/lib/openvas/mgr/tasks.db"

# JSON file to write results to

json_file = "/home/mike/openvas_results.json"


# CSV file to write results to - actually tab delimited csv_file =
"/home/mike/openvas_results.csv"

# last_id is how we keep track of the last item added. This keeps us from re-
processing old items. This value is kept in the openvas_state.txt file

last_id = 0


#openvas_state.txt - change this to 0 if you want to start over openvas_state_file =
"/home/mike/openvas_state.txt"

# this is just a status of how many records have be processed. new_record_count = 0

print "Getting OpenVAS reports"


if os.path.isfile(openvas_state_file) and os.access(openvas_state_file, os.W_OK):

openvas_state = open(openvas_state_file, 'r+') last_id = openvas_state.read()

else:

print "File %s does not exist, creating" % openvas_state_file

#sys.exit()

openvas_state = open(openvas_state_file, 'w') openvas_state.write('0')

print "Last ID = ", last_id


# stripped removes non-printable characters def stripped(x):

return "".join([i for i in x if 31 < ord(i) < 127])


try:

db_conn = sqlite3.connect(file_db, check_same_thread=False) except:

print "Cannot connect to %s" % file_db sys.exit()

db_cursor = db_conn.cursor()
```

```
#query = """SELECT id, task, subnet, host, port, nvt, type, description, report from
results"""


query = """SELECT results.id, results.task, results.subnet, results.host,
results.port, results.nvt, results.type, results.description, results.report,
nvts.name, nvts.description,

nvts.cve, nvts.cvss_base, nvts.risk_factor from results LEFT JOIN nvts ON results.nvt
= nvts.uuid ORDER BY results.id"""


#field_names = ['id', 'task', 'subnet', 'host', 'port', 'nvt', 'type',
'results_description', 'report', 'nvts_name', 'nvts_description', 'cve', 'cvss_base',
'risk_factor']


csvfile = open(csv_file, 'a')

csv_writer = csv.writer(csvfile, delimiter='\t', quotechar='|',
quoting=csv.QUOTE_MINIMAL)


jsonfile = open(json_file, 'a')


for row in db_cursor.execute(query):

#print row

id = row[0] #this needs to be a number task = stripped(str(row[1]))

subnet = stripped(str(row[2])) host = stripped(str(row[3])) port =
stripped(str(row[4])) nvt = stripped(str(row[5])) type = stripped(str(row[6]))

results_description = stripped(str(row[7])) report = stripped(str(row[8]))

nvts_name = stripped(str(row[9])) nvts_description = stripped(str(row[10])) cve =
stripped(str(row[11]))

cvss_base = stripped(str(row[12])) risk_factor = stripped(str(row[13]))


if int(id) > int(last_id):

#print "Greater!" last_id = id openvas_state.seek(0,0)

openvas_state.write(str(last_id)) new_record_count = new_record_count + 1

csv_writer.writerow([id, task, subnet, host, port, nvt, type, results_description,
report, nvts_name, nvts_description, cve, cvss_base, risk_factor])
```

```
json_dict = {'id': id, 'task': task, 'subnet': subnet, 'host': host, 'port': port,
'nvt': nvt, 'type': type, 'results_description': results_description, 'report':
report, 'nvts_name': nvts_name, 'nvts_description': nvts_description, 'cve': cve,
'cvss_base': cvss_base, 'risk_factor': risk_factor}

json.dump(json_dict, jsonfile, sort_keys = True, indent = 4, ensure_ascii = False)

#print "ID: %s      LAST: %s" % (id, last_id), print "\n"

db_conn.close() csvfile.close() jsonfile.close()

print "Wrote %s new records." % new_record_count
```

/opt/splunkforwarder/etc/system/local/server.conf

```
[sslConfig]

sslKeysfilePassword = $1$JnofjmZL66ZH


[lmpool:auto_generated_pool_forwarder] description = auto_generated_pool_forwarder
quota = MAX

slaves = *

stack_id = forwarder


[lmpool:auto_generated_pool_free] description = auto_generated_pool_free quota = MAX

slaves = * stack_id = free

[general]

pass4SymmKey = $1$cTZL0iMNoPRH serverName = openvas
```

/opt/splunkforwarder/etc/system/local/outputs.conf

```
[tcpout]

defaultGroup = splunkssl


[tcpout:splunkssl] compressed = true server = loghost:9997

sslCertPath = $SPLUNK_HOME/etc/certs/openvas.lab5.nccoe.gov.pem sslPassword =
$1$JnofjmZL66ZH

sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem sslVerifyServerCert = true
```

/opt/splunkforwarder/etc/system/local/inputs.conf

```
[default]  host = openvas

index = openvas sourcetype = openvas

[monitor:///home/mike/openvas_results.csv]
```

3.7 Puppet Enterprise

Puppet Enterprise enforces a configuration baseline on servers and workstations. Puppet agents installed on the hosts will run periodically, download a list of instructions referred to as a configuration catalog from the Master, and then execute it on the hosts. A successful Puppet Enterprise agent run can make configuration changes, install new software, remove unwanted software and send reports to the Master.

3.7.1 How It's Used

In the Financial Services ITAM solution, Puppet Enterprise is used to enforce a base configuration for all endpoints and to enforce basic security configurations. On the endpoints, it ensures that anti-virus software is installed, firewalls are enabled, IP forwarding is disabled, and the software asset management agent is installed.

Reporting is also a feature that was extended in this solution. With the inclusion of customized scripts, Puppet Enterprise sends very valuable reports to the ITAM analysis engine. The reports include which endpoint has successfully uploaded reports to the Puppet Enterprise master.

Failure to upload a report within a certain interval would indicate an anomaly with the endpoint or an off-line endpoint. Puppet Enterprise's functionality was extended to remove blacklisted software listed in a file made available from an analyst. A script was written to parse the file on a daily basis and inject the appropriate Puppet Enterprise code to remove such listed software. After successful removal, Puppet Enterprise writes a report identifying the offending endpoint, the uninstalled software and the time of removal.

3.7.2 Prerequisites

Puppet Enterprise Server requires the following:

- at least a four core CPU, 6 GB of RAM and 100 GB of hard drive space
- network-wide name resolution via DNS
- network-wide time synchronization using NTP

3.7.3 Installing Puppet Enterprise Server

Instructions for installing Puppet Enterprise can be found at http://docs.puppetlabs.com/pe/latest/install_pe_mono.html.

1. Download the Puppet Enterprise tarball from the Puppet Labs web site. Use the instructions referenced in the preceding link to locate and download the file.
2. Run `tar -xf <PuppetEnterpriseTarball>` to unpack its contents.
3. List directory with `ls` to view current directory contents.
4. Change into the directory with name `puppet-enterprise-<version>-<OSversion>`.
5. Execute `sudo ./puppet-enterprise-installer`.
6. Connect to Puppet Enterprise Server console by going to: **https://YourPuppetServerFQDN:3000**.
7. Accept the untrusted connection and make an exception to this site by storing it in your trusted list.
8. Confirm the security exception.
9. From Installation Web page, select **Let's get started**.
10. Select **Monolithic Installation**.
11. Choose **Install on this Server**.
12. Do not enable the Puppet 4 language parser if your existing Puppet code was developed in Puppet 3.xx.
13. Choose to install PostGreSQL on the same server.
14. Supply a console password when prompted.

3.7.4 Puppet Enterprise Linux Agent Installation

To install Puppet Enterprise agent on the same platform as the server:

1. Enter `curl -k https://YourPuppetServerFQDN:8140/packages/current/install.bash |sudo bash` at the agent terminal.
2. Request a certificate by typing `puppet agent -t` from the client node.
3. Go to the Puppet Enterprise server Web console and log in.

4. Accept node requests by clicking on the **Node** link.
5. Click **Accept** to sign the Certificate.

To install Puppet Enterprise agent on a different platform from the server:

1. Go to the Puppet Enterprise Web console.
2. Click on **Classification**.
3. Select the **PE Master Group**.
4. Click the **Classes** tab.
5. Select your platform from the new class textbox dropdown.
6. Click **Add Class**.
7. Click **Commit 1 Change**.
8. Run `puppet agent -t` to configure the newly assigned class.
9. To install the agent, enter `curl -k https://<YourPuppetServerFQDN>:8140/packages/current/install.bash | sudo bash`.

3.7.5 Puppet Enterprise Windows Agent Installation

To install Puppet Enterprise agent on a Windows computer:

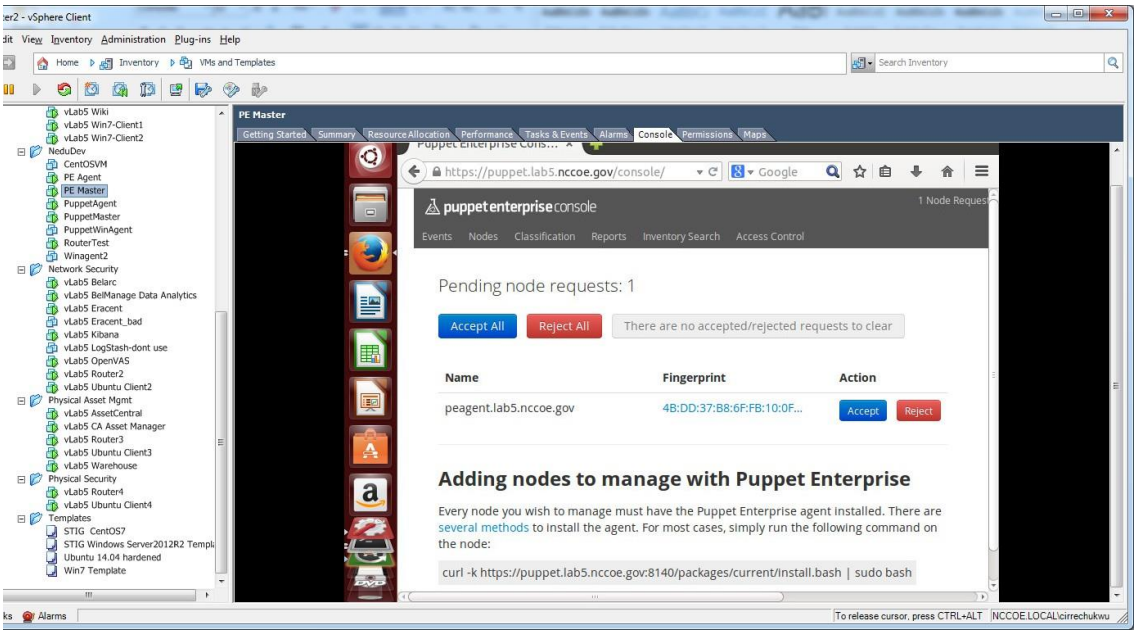
1. Make sure to start the installation file or log in to the system with an administrator account.
2. Double-click the Puppet Enterprise executable file.
3. Accept the default options.

3.7.6 Puppet Enterprise Agent Configuration

1. Agents need to obtain certificates from the Puppet Enterprise Server/Master. Connect to the Puppet Enterprise Server console at https://PuppetEnterpriseServerFQDN.
2. Log in to the console with your configured username and password.
3. Click on **Nodes**.
4. Accept Node requests from each agent you have configured. The agent's fully qualified domain name (FQDN) will be displayed.
5. A certificate request can be generated if you do not see one by typing `puppet agent -t` from the agent terminal.

6. Certificate requests can be viewed from the Web console of Puppet Enterprise Server.

7. Windows agents offer the option of using the graphical user interface by clicking on **Start Programs** > **Puppet Enterprise** > **Run Puppet Agent**.



8. Puppet agents fetch and apply configurations retrieved from the Puppet Enterprise Master Server. This agent run occurs every 30 minutes. You can change this interval by adding an entry to the */etc/puppetlabs/puppet/puppet.conf* file.

    a. On Linux, add the entry `runinterval = 12` to the main section of the */etc/puppetlabs/puppet/puppet.conf* file to have the agent run every 12 hours.

    b. On Windows, add the entry `runinterval = 12` to the main section of the *C:\ProgramData\PuppetLabs\puppet\etc\puppet.conf* file to have the agent run every 12 hours.

3.7.7 Puppet Enterprise Manifest Files and Modules

The main configuration file, also called a manifest file in Puppet Enterprise, is */etc/puppetlabs/puppet/environments/production/manifests/site.pp*. You can place all the Puppet Enterprise code here for agents to run. In our solution, we created modules, declared classes, and called those modules from within the site.pp file.

A module consists of a parent directory that contains a file's subdirectory and a manifest's subdirectory. Within the manifests subdirectory will be another file called init.pp that contains the Puppet Enterprise

code for that module. The init.pp file must have a class declaration statement. The files subdirectory can be empty or can contain files that need to be copied over to endpoints that will execute code in that module. All modules reside in the directory */etc/puppetlabs/puppet/modules*. We have the following modules:

- */etc/puppetlabs/puppet/modules/windowsnodes*
- */etc/puppetlabs/puppet/modules/ubuntubase*
- */etc/puppetlabs/puppet/modules/redhatbase*
- */etc/puppetlabs/puppet/modules/clamav*
- */etc/puppetlabs/puppet/modules/blacklist*

Each has a files directory */etc/puppetlabs/puppet/modules/<modulename>/files* and a manifests directory with the */etc/puppetlabs/puppet/modules/<modulename>/manifests/init.pp* file.

3.7.7.1 Module: windowsnodes

This module configures a baseline for Windows endpoints. Execution of this module copies a number of executable files and the baseline.bat script over to the endpoints from the Puppet Enterprise Server. Once baseline.bat is executed on the endpoint, it will look for and install the copied over executable programs, which consist of the belmonitor.exe asset management software agent and an anti-virus software. The text of the */etc/puppetlabs/puppet/modules/windowsnodes/init.pp* manifest file is shown in the code and scripts section.

3.7.7.2 Module: ubuntubase

This module configures a baseline for Ubuntu endpoints. It installs software, disables IP forwarding, installs clamav anti-virus, and copies over files including a script *dailyscript* that runs daily and is placed in the */etc/cron.daily* directory. You can use the same technique to ensure that your scripts remain where you want them.

3.7.7.3 Module: redhatbase

This module configures a baseline for RedHat or CentOS based endpoints. It disables IP forwarding on endpoints, copies over files including scripts that run periodically, ensures that the belmonitor asset management software is installed, and configures the logging to the appropriate logging server.

3.7.7.4 Module: clamav

This module installs clamav anti-virus on Ubuntu endpoints and ensures that the clamav-daemon service is running.

```
class clamav{


package{'clamav-daemon': ensure=>installed,

}


service{'clamav-daemon': ensure=>running, require=>Package['clamav-daemon'],

}

}
```

3.7.7.5 Module: blacklist

This module removes blacklisted software from endpoints and reports success if the software package is removed. Its *init.pp* file is constantly being updated with new software slated for removal. A python script called *blacklistenforcer.py* is used to populate the module's */etc/puppetlabs/puppet/modules/blacklist/manifests/init.pp* file. Another python script is used to read reports from the */var/opt/lib/pe-puppet/reports/<HostFQDN>* subdirectories to identify successfully removed blacklisted software.

3.7.7.6 Software Blacklist Removal

Puppet Enterprise Server is configured to remove blacklisted software from agent nodes. A python script placed in */etc/cron.daily* directory runs daily, checking a blacklisted software. The python script will extract the software list from the file */etc/splunkreport/fakeblacklist.csv*, write new Puppet code such that Puppet Enterprise catalog includes the blacklisted software, and identifies it to Puppet for removal.

3.7.8 Reporting

Puppet agents forward reports of their runs to the Puppet Enterprise server. To ensure reporting is enabled, go to */etc/puppetlabs/puppet/puppet.conf* and verify that an entry such as `reports = console, puppetdb, store` exists under master section of the file.

Agents upload reports in the form of YAML files to */var/opt/lib/pe-puppet/reports/<agent_hostname>*.

In this solution, the Puppet Enterprise Server machine was set up to forward two basic reports to the ITAM server. Both were done with scripts. The first reporting function forwarded checked the fully qualified hostnames of endpoints that failed to upload reports to the server within two reporting cycles.

If a reporting interval or cycle is 30 minutes, then failure to upload a report for more than an hour would indicate that an endpoint is offline and would trigger the forwarding of a syslog message to the ITAM server declaring the endpoint absent. Other endpoints that successfully upload reports without missing two cycles are declared present and send an appropriate message to the ITAM server. The script written that accomplishes this is written in BASH and is in the code and scripts section.

The second reporting function reports on the successful removal of blacklisted software. It scans through the report files from all the nodes in Puppet Enterprise Server, identifies successfully removed software and updates the CSV file */etc/splunkreport/reporttosplunk.csv* with information that identifies the endpoint, the successfully removed software and the time of removal. The Splunk Universal Forwarder agent monitors this file and forwards changes to the ITAM server, which uses Splunk Enterprise as its analysis engine.

3.7.9 Report Directory Cleanup

Thousands of files could be uploaded to the reports directory in a short time. Therefore, it is important to delete files that are no longer needed. We used a python script that ran hourly to delete files modification times more than 12 hours old. In this solution, that is equivalent to files that are more than 12 hours old. This script was placed in the */etc/cron.hourly*.

3.7.10 Puppet Code and Scripts

3.7.10.1 Main Manifest Configuration File

/etc/puppetlabs/puppet/environments/production/manifests/site.pp

```
## site.pp ##


# This file (/etc/puppetlabs/puppet/manifests/site.pp) is the main

# entry point used when an agent connects to a master and asks for an # updated
configuration.

#

# Global objects like filebuckets and resource defaults should go in

# this file,as should the default node definition. (The default node

# can be omitted

# if you use the console and don't define any other nodes in site.pp. # See
http://docs.puppetlabs.com/guides/language_guide.html#nodes for # more on node
definitions.)
```

```
## Active Configurations ##

# PRIMARY FILEBUCKET
# This configures puppet agent and puppet inspect to back up file
# contents when they run. The Puppet Enterprise console needs this to # display file
contents and differences.

# Define filebucket 'main': filebucket { 'main':
server => 'puppet.lab5.nccoe.gov', path => false,
}

# Make filebucket 'main' the default backup location for all File resources:
File { backup => 'main' }

# DEFAULT NODE
# Node definitions in this file are merged with node data from the console. See
# http://docs.puppetlabs.com/guides/language_guide.html#nodes for more
# on node definitions.

# The default node definition matches any node lacking a more specific
# node definition. If there are no other nodes in this file, classes
# declared here will be included in every node's catalog, *in
# addition* to any classes specified in the console for that node.

node default {
# This is where you can declare classes for all nodes.
# Example:
#      class { 'my_class': }

}
```

```
#Changes to the site.pp file were made below this line.
#Nodes were specified with the modules that would execute
#on them
node 'centos1', 'fathomsensor1'{ include redhatbase
include blacklist
}
node 'ubuntu-client1', 'kibana', 'openvas', 'sensu', 'ubuntu-client2', 'wiki'{
include blacklist include ubuntubase package{'curl':
ensure => installed,
}
}

node 'ubuntu-template', 'jumpbox', 'bro', 'snort', 'apt-cache', 'warehouse'{
include blacklist include ubuntubase package{'curl':
ensure => installed,
}
}

node 'win7-client1', 'win7-client2', 'ad2', 'ad1', 'Belarc', 'eracent'{ include
blacklist
include windowsnodes
}

node 'asset-manager'{ include blacklist include windowsnodes
}
```

3.7.10.2 Windowsnodes Configuration File and Script

/etc/puppetlabs/puppet/modules/windowsnodes/manifests/init.pp

```
#This manifest file declares a class called windowsnodes, creates a
#C:\software directory, copies a number of files to the agent including the
baseline.bat
```

```
#script and executes the baseline.bat. When executed baseline.bat batch file installs
#some programs and turns on the firewall and ensures the guest account is disabled
class windowsnodes{ file{'C:\software':
ensure=>"directory",
}

file{'C:\software\baseline.bat':
source => "puppet:///modules/windowsnodes/baseline.bat", source_permissions=>ignore,
require => File['C:\software'],
}

file{'C:\software\belmonitor.exe':
source => "puppet:///modules/windowsnodes/belmonitor.exe", source_permissions=>ignore,
require => File['C:\software'],
}

file{'C:\software\mbamsetup.exe':
source => "puppet:///modules/windowsnodes/mbamsetup.exe", source_permissions=>ignore,
require => File['C:\software'],
}

exec{'win_baseline':
command=>'C:\windows\system32\cmd.exe /c C:\software\baseline.bat', require =>
File['C:\software\belmonitor.exe'],
}

file{'C:\Program Files (x86)\nxlog\conf\nxlog.conf': source =>
"puppet:///modules/windowsnodes/nxlog.conf", source_permissions=>ignore,
}

}
```

/etc/puppetlabs/puppet/modules/windowsnodes/files/baseline.bat

```
REM Install new user called newuser net user newuser /add

REM Disable newuser

net user newuser /active:no


REM Disable the guest account net user guest /active:no

REM Turn on firewall

netsh advfirewall set allprofiles state on


REM Use puppet to check if Malwarebytes is installed puppet resource package |find
"Malwarebytes"

REM Install Malwarebytes silently if not installed

if %errorlevel% neq 0 C:\software\mbamsetup.exe /verysilent /norestart sc query |find
"BelMonitorService"

REM Install Belmonitor if the service is not running if %errorlevel% neq 0
C:\software\belmonitor.exe
```

3.7.10.3 Ubuntubase Configuration File and Script

/etc/puppetlabs/puppet/modules/ubuntubase/manifests/init.pp

```
#This module configures a baseline for Ubuntu endpoints class ubuntubase{

#Copy over the CA certificate

file{'/usr/local/share/ca-certificates/CAServerCert.crt': source =>
"puppet:///modules/ubuntubase/CAServerCert.crt",

}


# Add CA certificate to Ubuntu endpoint's repository of certificates exec{'update-ca-
certificates':

command=>'/usr/sbin/update-ca-certificates',

}
```

```
#Ensure the /etc/ufw directory is present or create it file{'/etc/ufw':

ensure=>"directory",

}


#Copy over the sysctl.conf file to each endpoint. IP forwarding will be

#disabled file{'/etc/ufw/sysctl.conf':

source => "puppet:///modules/ubuntubase/sysctl.conf", require => File['/etc/ufw'],

}


#Run the clamav module include clamav

file{'/etc/cron.daily': ensure=>"directory",

}


file{'/etc/rsyslog.d': ensure=>"directory",

}


#Copy over this script to endpoint with associated permissions
file{'/etc/cron.daily/dailyscript':

source => "puppet:///modules/ubuntubase/dailyscript", mode => 754,

require => File['/etc/cron.daily'],

}


#Copy over the 50-default.conf file with specified content file{'/etc/rsyslog.d/50-
default.conf':

content => "*.* @@loghost\n *.* /var/log/syslog", require => File['/etc/rsyslog.d'],

}


#Copy over Belmonitor Linux installation file file{'/opt/BelMonitorLinux':

source => "puppet:///modules/ubuntubase/BelMonitorLinux",

}
```

```
#Make the BelMonitorLinux file executable exec{'belmonitor_executable':
command=>'/bin/chmod a+x /opt/BelMonitorLinux', require=>File['/opt/BelMonitorLinux'],

}


exec{'install_rpm':

command=>'/usr/bin/apt-get install -y rpm', require=>File['/opt/BelMonitorLinux']

}


##Install 32 bit library exec{'install_32bitlibrary':

command=>'/usr/bin/apt-get install -y gcc-multilib', require=>Exec['install_rpm'],

}


##install 32 bit library exec{'install_second_32bit_library':

command=> '/usr/bin/apt-get install -y lib32stdc++6',

}


exec{'install_belmonitor': command=>'/opt/BelMonitorLinux',
require=>Exec['install_32bitlibrary'],

}


service{'BelMonitor': ensure=>'running',

}

}
```

/etc/puppetlabs/puppet/modules/ubuntubase/files/dailyscript

```
#!/bin/bash df -kh mount

netstat -nult ifconfig -a iptables -L

/usr/bin/freshclam

cat /var/lib/apt/extended_states apt-get update
```

3.7.10.4 Redhatbase Module Configuration File and Script

/etc/puppetlabs/puppet/modules/redhatbase/manifests/init.pp

```
class redhatbase{


#Copies over a customized sysctl.conf that disables IP forwarding
file{'/etc/sysctl.conf':

source => "puppet:///modules/redhatbase/sysctl.conf",

}


#Ensures that cron.daily directory is present or creates it file{'/etc/cron.daily':

ensure=>"directory",

}


file{'/etc/rsyslog.d': ensure=>"directory",

}




#Copies over the a script that runs daily called dailyscript
file{'/etc/cron.daily/dailyscript':

source => "puppet:///modules/redhatbase/dailyscript", mode => 754,

require => File['/etc/cron.daily'],

}


#Ensures that log messages are forwarded to loghost and

/var/log/messages file{'/etc/rsyslog.d/50-default.conf':

content => "*.* @@loghost:514\n *.* /var/log/messages", require =>
File['/etc/rsyslog.d'],

}

#Copies over the a script that installs clamav if not installed
file{'/etc/cron.daily/claminstall':

source => "puppet:///modules/redhatbase/claminstall", mode => 754,
```

```
require => File['/etc/cron.daily'],

}


##Ensure the opt dir is present, copy the BelMonitorLinux script file

## Copy the belmonitor_install script to the /opt dir

## Check that the BelMonitor file is present before belmonitor_install

## executes


file{'/opt': ensure=>"directory",

}

file{'/opt/BelMonitorLinux':

source => "puppet:///modules/redhatbase/BelMonitorLinux",

}


##Make BelMonitorLinux executable exec{'make_executable':

command=>'/bin/chmod a+x /opt/BelMonitorLinux', require =>
File['/opt/BelMonitorLinux'],

}


##Install dependencies exec{'upgrade_dep1':

command=>'/usr/bin/yum -y upgrade libstdc++',


}


exec{'install_dep2':

command=>'/usr/bin/yum -y install libstdc++.i686',

}


exec{'upgrade_dep3': command=>'/usr/bin/yum -y upgrade zlib',

}
```

```
exec{'install_dep4':

command=>'/usr/bin/yum -y install zlib.i686',

}


exec{'install_belmonitor': command=>'/opt/BelMonitorLinux',

}

file{'/opt/belmonitor_install':

source => "puppet:///modules/redhatbase/belmonitor_install",

}


}
```

/etc/puppetlabs/puppet/modules/redhatbase/files/claminstall

```
#!/bin/bash

# /etc/puppetlabs/puppet/modules/redhatbase/files/claminstall#

# Script installs clamav if not already installed when run


if rpm -qa clamav; then

echo "Clamav is installed" else

yum install -y epel-release

yum --enablerepo=epel -y install clamav clamav-update sed -i -e "s/^Example/#Example/"
/etc/freshclam.conf
```

3.7.10.5 Clamav Puppet Module Configuration File

/etc/puppetlabs/puppet/modules/clamav/manifests/init.pp class clamav{

```
package{'clamav-daemon': ensure=>installed,

}


service{'clamav-daemon': ensure=>running, require=>Package['clamav-daemon'],

}
```

```
}
```

3.7.10.6 Blacklisted Software Removal Script

/etc/puppetlabs/puppet/modules/blacklist/manifests/init.pp

```
#!/usr/bin/python3

#--------------------------------readreport.py-----------------------#

#Script will search through the Puppet reports directory and subdirectories, and
identify blacklisted

#packages within the yaml files that have been confirmed as removed. It will retrieve
the software

#package, host and time of removal and write this to a file called reporttosplunk.csv


import os

#List directories in /var/opt/lib/pe-puppet/reports report_list =
os.listdir('/var/opt/lib/pe-puppet/reports')

#Make the path to reports a string

origdir_path = '/var/opt/lib/pe-puppet/reports'


action_term = "file:

/etc/puppetlabs/puppet/modules/blacklist/manifests/init.pp" outfile =
open('/etc/splunkreport/reporttosplunk.csv', 'a')

#For loop iterates through report_list (or the reports directory) for sub_dirs in
report_list:

hostname = sub_dirs print(hostname)

#Concatenation creates the full path to subdirectories (it remains a string)

subdir_path = origdir_path+'/'+sub_dirs

#Creates the list of files in the variable (the variable in this case would be a sub
directory)

#At the end of this block, infile contains a list of line elements in each file

sub_dirs_list = os.listdir(subdir_path) for files in sub_dirs_list:

files_path = subdir_path+'/'+files reportfile = open(files_path, "r") infile =
reportfile.readlines() reportfile.close()

#line_counter used in keeping track of the index for the line elements in each file
```

```
line_counter = 0


for line in infile:

if action_term in line:

if "source" in infile[line_counter + 3]: bad_package = infile[line_counter + 3]
bad_package = bad_package.replace('\n',',') if "removed" in infile[line_counter + 2]:

message_var = infile[line_counter + 2] message_var = message_var.replace('\n',',') if
"time" in infile[line_counter + 1]:

time_var = infile[line_counter + 1] time_var = time_var.replace('\n',',')
refined_bad_pkg = bad_package.split('/') bad_pkg = refined_bad_pkg[3]

bad_pkg = bad_pkg + ","

print(hostname+","+bad_pkg+message_var+time_var+'\n')


outfile.write(hostname+','+bad_pkg+message_var+time_var+'\n') line_counter =
line_counter + 1
```

3.7.10.7 Reports Directory Cleanup Script

/etc/cron.hourly/cleanreportdir.py

```
#!/usr/bin/python3


#-------------------cleanreportdir.py----------------------------#

#Script removes files with mtimes older than 12 hours to keep the number of files to a
manageable size

#Files removed are from the reports subdirectory within Puppet import os

import time

#List directories in /var/opt/lib/pe-puppet/reports report_list =
os.listdir('/var/opt/lib/pe-puppet/reports')

#Make the path to reports a string

origdir_path = '/var/opt/lib/pe-puppet/reports'

#For loop iterates through report_list for sub_dirs in report_list:

#Concatenation creates the full path to subdirectories (it remains a string)

subdir_path = origdir_path+'/'+sub_dirs
```

```
print('Old files are being removed from ',subdir_path)

#Creates the list of files in the variable sub_dirs_list sub_dirs_list =
os.listdir(subdir_path)

for files in sub_dirs_list:

files_path = subdir_path+'/'+files mtime = os.path.getmtime(files_path) current_time =
time.time()

time_diff = current_time - mtime

#Removes files with mtimes older than 12 hours if time_diff > 43200:

print(files_path, " will be deleted") os.remove(files_path)
```

3.7.10.8 Reporting Section Script

```
#!/bin/bash

#/etc/cron.hourly/nodereport

#Time in seconds before declaring an agent that has not checked in absent

#Change the time to suit your needs

let "desired_interval=3600"


for node in $(ls /var/opt/lib/pe-puppet/yaml/node) do

#Strip out the yaml extension from the node name node=${node%.*}

#Get time of most recent agent run or check in

#This time will be reported without formatting node_report_time=$(date -r

/var/opt/lib/pe-puppet/yaml/facts/$node.yaml)

#Get epoch time of agent facter yaml file, assign time to variable node_time=$(date
+%s -r

/var/opt/lib/pe-puppet/yaml/facts/$node.yaml)

#Assign current epoch_time to variable current_time=$(date +%s)

#Subtract node most recent report time from current time and

#assign to variable node_interval=$((current_time-node_time))

#Nodes that have not reported in the given interval are

#declared absent, otherwise they are declared present if (("$node_interval" >
"$desired_interval"))

then

echo $node "is absent with a last run time of "
```

```
$node_report_time
logger $node "is absent. Last run is " $node_report_time


else
echo $node "is present with a last run time of "
$node_report_time
logger $node "is present. Last run is " $node_report_time
fi
done
```

3.8 Snort

Snort is an open-source intrusion detection system. Snort efficiently analyzes all network traffic and matches it with signatures of know bad traffic. An alert is generated if a signature is matched.

3.8.1 How It's Used

In the FS ITAM build, Snort monitors all traffic traversing the DMZ.

On the high-level architecture diagram, Snort is in Tier 2. Snort utilizes the Splunk Universal Forwarder to send alerts to Splunk Enterprise.

3.8.2 Virtual Machine Configuration

The Snort virtual machine is configured with one network interface card, 2 GB of RAM and one CPU core.

3.8.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.0.40
- Netmask: 255.255.255.0
- Gateway: 172.16.0.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.8.4 Installing Snort

Snort is installed on a hardened Ubuntu 14.04 Linux system. Complete installation instructions can be found at: https://www.snort.org/.

This installation utilized the Snort IDS and Barnyard2 to interpret binary Snort alerts into readable text.

3.8.5 Installing Snort

1. For Debian/Ubuntu Linux systems, it is always best to make sure your system is up-to-date by performing:

   sudo apt-get update

   sudo apt-get upgrade

   sudo apt-get install snort

2. You will be asked to input your local networks. For the FS-ITAM lab this is **172.16.0.0/16**.

3. Configure */etc/snort/snort.debian.conf*.

4. Make sure that the correct HOME_NET and INTERFACE are specified in */etc/snort/snort.debian.conf*.

   DEBIAN_SNORT_HOME_NET="172.16.0.0/16"

   DEBIAN_SNORT_INTERFACE="eth0"

5. Configure */etc/snort/snort.conf*.

6. Comment out all output configuration lines and add the following:

   output unified2: filename /var/log/snort/snort.log, limit 128, mpls_event_types, vlan_event_types

   The preceding line is important for Barnyard2 to work correctly.

3.8.6 Get Updated Community Rules

```
cd /opt

wget https://snort.org/downloads/community/community-rules.tar.gz tar xzvf
community.rules.tar.gz -C /etc/snort/rules
```

These community rules contain the **sid-msg.map** file that Barnyard2 needs.

```
mkdir /etc/snort/etc

cp /etc/snort/rules/community-rules/sid-msg.map /etc/snort/etc
```

Note: In a production environment, it is advisable to install an automatic rule updater such as PulledPork. PulledPork requires obtaining an account at Snort.org which results in an Oinkcode.

3.8.7 Installing Barnyard2

1. Install the prerequisites:

```
sudo apt-get install build-essential libtool autoconf git nmap
sudo apt-get install libpcap-dev libmysqld-dev libpcre3-dev libdumbnet-dev
sudo apt-get install flex bison ldconfig
```

2. Barnyard2 requires the <dnet.h> header. Unfortunately, Ubuntu names this header <dumb-net.h> so we must create a symbolic link for Barnyard2 to compile.

```
cd /usr/include
ln -s /usr/include/dumbnet.h dnet.h
```

   Note: You need to be root to install Barnyard2.

```
cd /opt
Need the Daq libraries from Snort
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
tar xzvf daq-2.0.6.tar.gz
cd /opt/daq-2.0.6
./configure make
make install
git clone https://github.com/firnsy/barnyard2.git
cd /opt/barnyard2
./autogen.sh
./configure make
make install
```

3. Copy the provided **barnyard2.conf** file to */usr/local/etc*:

```
cp /usr/local/etc/barnyard2.conf /usr/local/etc/barnyard2.conf.orig
cp <barnyard2.conf> /usr/local/etc
```

4. Create a link inside /etc/snort to this file:

```
ln -s /usr/local/etc/barnyard2 /etc/snort/barnyard.conf
```

5. Copy the provided **barnyard2** init script to */etc/init.d* and make it executable:

```
cp <barnyard2> /etc/init.d chmod 755 /etc/init.d/barnyard2

sudo update-rc.d barnyard2 defaults sudo update-rc.d barnyard2 enable
```

6. Start up Barnyard2:

```
/etc/init.d/barnyard2 start
```

Error messages can be found in */var/log/syslog*.

3.8.8 Testing

Performing these steps will let you know that Snort and Barnyard2 are working.

1. Add a local rule.
2. Edit */etc/snort/rules/local.rules* by adding the following line at the bottom that will generate alerts for any ICMP/Ping traffic:

   alert icmp any any -> any any (msg: "ICMP Detected";classtype:unknown; sid:1000001; rev:1;)

   Note: the sid must be greater than 1 million.
3. Restart Snort:

```
service snort restart
```

4. Verify that Snort is running:

```
ps -ef |grep snort
```

5. Verify that Barnyard2 is running:

```
ps -ef |grep barnyard2
```

6. Check the logs in */var/log/snort*. The snort.log and alert files should both be growing fast.
7. You can view the alert file:

```
tail -f /var/log/snort/alert
```

   Note: Do not leave this test running. If you do, it will fill your hard drive.
8. If everything is good, just comment out the line that you created in local.rules and restart Snort.

3.8.9 Installing Splunk Universal Forwarder

Note: You will need a Splunk account to download the Splunk Universal Forwarder. It is free and can be set up at: https://www.splunk.com/page/sign_up.

1. Download the Splunk Universal Forwarder from: http://www.splunk.com/en_us/download/universal-forwarder.html.
2. You want the latest version for `OS version 2.6+ kernel Linux distributions (64-bit)`. Since this is installing on Ubuntu, select the file that ends in `.deb`. An example is:

   ```
   splunkforwader-6.2.5-272645-linux-2.6-amd64.deb
   ```

   Detailed installation instructions can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Installation/InstallonLinux.
3. An abridged version follows:

   ```
   dpkg -i <splunk_package_name.deb>
   ```

   Example: `dpkg -i splunkforwader-6.2.5-272645-linux-2.6-amd64.deb`
4. This will install in */opt/splunkforwarder*:

   ```
   cd /opt/splunkforwarder/bin
   ./splunk start --accept-license
   ./splunk enable boot-start
   ```
5. Add forwarder:

   More information about adding a forwarder can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Forwarding/Deployanixdfmanually.

   ```
   cd /opt/splunkforwarder/bin
   ./splunk add forward-server loghost:9997 -auth admin:changme
   ```

3.8.10 Configuring Splunk Universal Forwarder

Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You will also need a copy of your certificate authority's public certificate.

1. Create a directory to hold your certificates:

   ```
   mkdir /opt/splunkforwarder/etc/certs
   ```

2. Copy your certificates in PEM format to */opt/splunkforwarder/etc/certs*:

   ```
   cp CAServerCert.pem /opt/splunkforwarder/etc/certs
   cp bro_worker1.pem /opt/splunkforwarder/etc/certs
   ```

3. Copy Splunk Universal Forwarder configuration files:

   ```
   cp <server.conf> /opt/splunkforwarder/etc/system/local
   cp <inputs.conf> /opt/splunkforwarder/etc/system/local
   cp <outputs.conf> /opt/splunkforwarder/etc/system/local
   ```

4. Modify **server.conf** so that:

   ServerName=snort is your hostname.

   ```
   sslKeysfilePassword = <password for your private key>
   ```

5. Modify **outputs.conf** so that:

   Server = loghost:9997 is your correct Splunk Enterprise server/indexer and port.

   ```
   sslPassword = <password of your certificate private key>
   ```

Note: This will be hashed and not clear text after a restart.

Inputs.conf should work, but you are free to modify it to include the Bro logs that you are interested in.

3.8.11 Configurations and Scripts

/etc/default/barnyard2

```
# Config file for /etc/init.d/barnyard2
#LOG_FILE="snort_unified.log" LOG_FILE="snort.log"
# You probably don't want to change this, but in case you do SNORTDIR="/var/log/snort"
INTERFACES="eth0"

# Probably not this either CONF=/etc/snort/barnyard2.conf
EXTRA_ARGS="
```

/etc/snort/snort.conf

```
#--------------------------------------------------
```

```
#      VRT Rule Packages Snort.conf
#
#      For more information visit us at:
#      http://www.snort.org       Snort Website
#      http://vrt-blog.snort.org/ Sourcefire VRT Blog
#
#      Mailing list Contact:      snort-sigs@lists.sourceforge.net
#      False Positive reports:    fp@sourcefire.com
#      Snort bugs:  bugs@snort.org
#
#      Compatible with Snort Versions:
#      VERSIONS : 2.9.6.0
#
#      Snort build options:
#      OPTIONS : --enable-gre --enable-mpls --enable-targetbased
--enable-ppm --enable-perfprofiling --enable-zlib
--enable-active-response --enable-normalizer --enable-reload
--enable-react --enable-flexresp3
#
#      Additional information:
#      This configuration file enables active response, to run snort in
#      test mode -T you are required to supply an interface -i
<interface>
#      or test mode will fail to fully validate the configuration and
#      exit with a FATAL error
#--------------------------------------------------
##################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
```

```
#      1) Set the network variables.
#      2) Configure the decoder
#      3) Configure the base detection engine
#      4) Configure dynamic loaded libraries
#      5) Configure preprocessors
#      6) Configure output plugins
#      7) Customize your rule set
#      8) Customize preprocessor and decoder rule set
#      9) Customize shared object rule set
##################################################

##################################################
# Step #1: Set the network variables. For more information, see README.variables
##################################################

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET
```

```
# List of DNS servers on your network ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network ipvar SQL_SERVERS $HOME_NET

# List of telnet servers on your network ipvar TELNET_SERVERS $HOME_NET

# List of ssh servers on your network ipvar SSH_SERVERS $HOME_NET

# List of ftp servers on your network ipvar FTP_SERVERS $HOME_NET

# List of sip servers on your network ipvar SIP_SERVERS $HOME_NET

# List of ports you run web servers on portvar HTTP_PORTS

[36,80,81,82,83,84,85,86,87,88,89,90,311,383,555,591,593,631,801,808,8

18,901,972,1158,1220,1414,1533,1741,1830,2231,2301,2381,2809,3029,3037

,3057,3128,3443,3702,4000,4343,4848,5117,5250,6080,6173,6988,7000,7001

,7144,7145,7510,7770,7777,7779,8000,8008,8014,8028,8080,8081,8082,8085

,8088,8090,8118,8123,8180,8181,8222,8243,8280,8300,8500,8509,8800,8888

,8899,9000,9060,9080,9090,9091,9111,9443,9999,10000,11371,12601,15489,

29991,33300,34412,34443,34444,41080,44449,50000,50002,51423,53331,5525

2,55555,56712]

# List of ports you want to look for SHELLCODE on. portvar SHELLCODE_PORTS !80

# List of ports you might see oracle attacks on portvar ORACLE_PORTS 1024:

# List of ports you want to look for SSH connections on: portvar SSH_PORTS 22

# List of ports you run ftp servers on portvar FTP_PORTS [21,2100,3535]

# List of ports you run SIP servers on portvar SIP_PORTS [5060,5061,5600]

# List of file data ports for file inspection portvar FILE_DATA_PORTS
[$HTTP_PORTS,110,143]

# List of GTP ports for GTP preprocessor portvar GTP_PORTS [2123,2152,3386]

# other variables, these should not be modified ipvar AIM_SERVERS

[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0

/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.18
8.153.0/24,205.188.179.0/24,205.188.248.0/24]
```

```
# Path to your rules files (this can be a relative path)

# Note for Windows users:  You are advised to make this an absolute path,

# such as:   c:\snort\rules

#var RULE_PATH /etc/snort/rules var RULE_PATH rules

var SO_RULE_PATH /etc/snort/so_rules

var PREPROC_RULE_PATH /etc/snort/preproc_rules


# If you are using reputation preprocessor set these

# Currently there is a bug with relative paths, they are relative to where snort is

# not relative to snort.conf like the above variables

# This is completely inconsistent with how other vars work, BUG 89986

# Set the absolute path appropriately var WHITE_LIST_PATH /etc/snort/rules var
BLACK_LIST_PATH /etc/snort/rules

##################################################

# Step #2: Configure the decoder. For more information, see README.decode

##################################################


# Stop generic decode events: config disable_decode_alerts

# Stop Alerts on experimental TCP options config disable_tcpopt_experimental_alerts

# Stop Alerts on obsolete TCP options config disable_tcpopt_obsolete_alerts

# Stop Alerts on T/TCP alerts config disable_tcpopt_ttcp_alerts

# Stop Alerts on all other TCPOption type events: config disable_tcpopt_alerts

# Stop Alerts on invalid ip options config disable_ipopt_alerts

# Alert if value in length field (IP, TCP, UDP) is greater th elength of the packet

# config enable_decode_oversized_alerts


# Same as above, but drop packet if in Inline mode (requires
enable_decode_oversized_alerts)

# config enable_decode_oversized_drops


# Configure IP / TCP checksum mode config checksum_mode: all
```

```
# Configure maximum number of flowbit references. For more information, see
README.flowbits

# config flowbits_size: 64


# Configure ports to ignore

# config ignore_ports: tcp 21 6667:6671 1356

# config ignore_ports: udp 1:17 53


# Configure active response for non inline operation. For more information, see
REAMDE.active

# config response: eth0 attempts 2

# Configure DAQ related options for inline operation. For more information, see
README.daq

#

# config daq: <type>

# config daq_dir: <dir>

# config daq_mode: <mode>

# config daq_var: <var>

#

# <type> ::= pcap | afpacket | dump | nfq | ipq | ipfw

# <mode> ::= read-file | passive | inline

# <var> ::= arbitrary <name>=<value passed to DAQ

# <dir> ::= path as to where to look for DAQ module so's


# Configure specific UID and GID to run snort as after dropping privs. For more
information see snort -h command line options

#

# config set_gid:

# config set_uid:


# Configure default snaplen. Snort defaults to MTU of in use interface. For more
information see README
```

```
#
# config snaplen:
#


# Configure default bpf_file to use for filtering what traffic reaches snort. For more
information see snort -h command line options (-F)
#
# config bpf_file:
#


# Configure default log directory for snort to log to. For more information see snort
-h command line options (-l)
#
# config logdir:


##################################################
# Step #3: Configure the base detection engine. For more information, see README.decode
##################################################


# Configure PCRE match limitations config pcre_match_limit: 3500
config pcre_match_limit_recursion: 1500


# Configure the detection engine  See the Snort Manual, Configuring Snort - Includes -
Config
config detection: search-method ac-split search-optimize max-pattern-len 20


# Configure the event queue.      For more information, see README.event_queue
config event_queue: max_queue 8 log 5 order_events content_length
##################################################
## Configure GTP if it is to be used.
## For more information, see README.GTP
##################################################
```

```
# config enable_gtp

##################################################
# Per packet and rule latency enforcement
# For more information see README.ppm
##################################################

# Per Packet latency configuration
#config ppm: max-pkt-time 250, \
#      fastpath-expensive-packets, \
#      pkt-log

# Per Rule latency configuration
#config ppm: max-rule-time 200, \
#      threshold 3, \
#      suspend-expensive-rules, \
#      suspend-timeout 20, \
#      rule-log alert

##################################################
# Configure Perf Profiling for debugging
# For more information see README.PerfProfiling
##################################################

#config profile_rules: print all, sort avg_ticks
#config profile_preprocs: print all, sort avg_ticks

##################################################
# Configure protocol aware flushing
```

```
# For more information see README.stream5
##################################################
config paf_max: 16000

##################################################
# Step #4: Configure dynamic loaded libraries.
# For more information, see Snort Manual, Configuring Snort - Dynamic Modules
##################################################
# path to dynamic preprocessor libraries
dynamicpreprocessor directory /usr/lib/snort_dynamicpreprocessor/

# path to base preprocessor engine
dynamicengine /usr/lib/snort_dynamicengine/libsf_engine.so

# path to dynamic rules libraries
dynamicdetection directory /usr/lib/snort_dynamicrules

##################################################
# Step #5: Configure preprocessors
# For more information, see the Snort Manual, Configuring Snort - Preprocessors
##################################################

# GTP Control Channle Preprocessor. For more information, see README.GTP
# preprocessor gtp: ports { 2123 3386 2152 }

# Inline packet normalization. For more information, see README.normalize
# Does nothing in IDS mode preprocessor normalize_ip4
preprocessor normalize_tcp: ips ecn stream preprocessor normalize_icmp4
preprocessor normalize_ip6 preprocessor normalize_icmp6
```

```
# Target-based IP defragmentation.      For more inforation, see README.frag3

preprocessor frag3_global: max_frags 65536

preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10
min_fragment_length 100 timeout 180


# Target-Based stateful inspection/stream reassembly. For more inforation, see
README.stream5

preprocessor stream5_global: track_tcp yes, \ track_udp yes, \

track_icmp no, \ max_tcp 262144, \

max_udp 131072, \

max_active_responses 2, \

min_response_seconds 5

preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, \

overlap_limit 10, small_segments 3 bytes 150, timeout 180, \

ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137

139 143 \

161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665

6666 6667 6668 6669 \

7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778

32779, \

ports both 36 80 81 82 83 84 85 86 87 88 89 90 110 311 383 443 465

563 555 591 593 631 636 801 808 818 901 972 989 992 993 994 995 1158

1220 1414 1533 1741 1830 2231 2301 2381 2809 3029 3037 3057 3128 3443

3702 4000 4343 4848 5117 5250 6080 6173 6988 7907 7000 7001 7144 7145

7510 7802 7770 7777 7779 \

7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912

7913 7914 7915 7916 \

7917 7918 7919 7920 8000 8008 8014 8028 8080 8081 8082 8085 8088

8090 8118 8123 8180 8181 8222 8243 8280 8300 8500 8509 8800 8888 8899

9000 9060 9080 9090 9091 9111 9443 9999 10000 11371 12601 15489 29991

33300 34412 34443 34444 41080 44449 50000 50002 51423 53331 55252 55555
```

```
56712

preprocessor stream5_udp: timeout 180


# performance statistics.  For more information, see the Snort Manual, Configuring
Snort - Preprocessors - Performance Monitor

# preprocessor perfmonitor: time 300 file /var/snort/snort.stats pktcnt 10000


# HTTP normalization and anomaly detection.    For more information, see
README.http_inspect

preprocessor http_inspect: global iis_unicode_map unicode.map 1252 compress_depth
65535 decompress_depth 65535 max_gzip_mem 104857600

preprocessor http_inspect_server: server default \

http_methods { GET POST PUT SEARCH MKCOL COPY MOVE LOCK UNLOCK NOTIFY POLL BCOPY
BDELETE BMOVE LINK UNLINK OPTIONS HEAD DELETE TRACE TRACK CONNECT SOURCE SUBSCRIBE
UNSUBSCRIBE PROPFIND PROPPATCH BPROPFIND BPROPPATCH RPC_CONNECT PROXY_SUCCESS
BITS_POST CCM_POST SMS_POST RPC_IN_DATA RPC_OUT_DATA RPC_ECHO_DATA } \

chunk_length 500000 \

server_flow_depth 0 \

client_flow_depth 0 \

post_depth 65495 \

oversize_dir_length 500 \

max_header_length 750 \

max_headers 100 \

max_spaces 200 \

small_chunk_length { 10 5 } \

ports { 36 80 81 82 83 84 85 86 87 88 89 90 311 383 555 591 593 631

801 808 818 901 972 1158 1220 1414 1741 1830 2231 2301 2381 2809 3029

3037 3057 3128 3443 3702 4000 4343 4848 5117 5250 6080 6173 6988 7000

7001 7144 7145 7510 7770 7777 7779 8000 8008 8014 8028 8080 8081 8082

8085 8088 8090 8118 8123 8180 8181 8222 8243 8280 8300 8500 8509 8800

8888 8899 9000 9060 9080 9090 9091 9111 9443 9999 10000 11371 12601

15489 29991 33300 34412 34443 34444 41080 44449 50000 50002 51423 53331

55252 55555 56712 } \
```

```
non_rfc_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \ enable_cookie \

extended_response_inspection \ inspect_gzip \

normalize_utf \ unlimited_decompress \ normalize_javascript \ apache_whitespace no \
ascii no \

bare_byte no \ directory no \ double_decode no \ iis_backslash no \ iis_delimiter no \
iis_unicode no \ multi_slash no \ utf_8 no \ u_encode yes \ webroot no

# ONC-RPC normalization and anomaly detection. For more information, see the Snort
Manual, Configuring Snort - Preprocessors - RPC Decode

preprocessor rpc_decode: 111 32770 32771 32772 32773 32774 32775 32776

32777 32778 32779 no_alert_multiple_requests no_alert_large_fragments
no_alert_incomplete


# Back Orifice detection. preprocessor bo

# FTP / Telnet normalization and anomaly detection.  For more information, see
README.ftptelnet

preprocessor ftp_telnet: global inspection_type stateful encrypted_traffic no
check_encrypted

preprocessor ftp_telnet_protocol: telnet \ ayt_attack_thresh 20 \

normalize ports { 23 } \ detect_anomalies

preprocessor ftp_telnet_protocol: ftp server default \ def_max_param_len 100 \

ports { 21 2100 3535 } \ telnet_cmds yes \ ignore_telnet_erase_cmds yes \

ftp_cmds { ABOR ACCT ADAT ALLO APPE AUTH CCC CDUP } \ ftp_cmds { CEL CLNT CMD CONF CWD
DELE ENC EPRT } \ ftp_cmds { EPSV ESTA ESTP FEAT HELP LANG LIST LPRT } \ ftp_cmds {
LPSV MACB MAIL MDTM MIC MKD MLSD MLST } \ ftp_cmds { MODE NLST NOOP OPTS PASS PASV
PBSZ PORT } \ ftp_cmds { PROT PWD QUIT REIN REST RETR RMD RNFR } \ ftp_cmds { RNTO
SDUP SITE SIZE SMNT STAT STOR STOU } \ ftp_cmds { STRU SYST TEST TYPE USER XCUP XCRC
XCWD } \ ftp_cmds { XMAS XMD5 XMKD XPWD XRCP XRMD XRSQ XSEM } \

ftp_cmds { XSEN XSHA1 XSHA256 } \

alt_max_param_len 0 { ABOR CCC CDUP ESTA FEAT LPSV NOOP PASV PWD QUIT REIN STOU SYST
XCUP XPWD } \

alt_max_param_len 200 { ALLO APPE CMD HELP NLST RETR RNFR STOR STOU XMKD } \

alt_max_param_len 256 { CWD RNTO } \ alt_max_param_len 400 { PORT } \
alt_max_param_len 512 { SIZE } \

chk_str_fmt { ACCT ADAT ALLO APPE AUTH CEL CLNT CMD } \ chk_str_fmt { CONF CWD DELE
ENC EPRT EPSV ESTP HELP } \ chk_str_fmt { LANG LIST LPRT MACB MAIL MDTM MIC MKD } \
chk_str_fmt { MLSD MLST MODE NLST OPTS PASS PBSZ PORT } \ chk_str_fmt { PROT REST RETR
```

```
RMD RNFR RNTO SDUP SITE } \ chk_str_fmt { SIZE SMNT STAT STOR STRU TEST TYPE USER } \
chk_str_fmt { XCRC XCWD XMAS XMD5 XMKD XRCP XRMD XRSQ } \

chk_str_fmt { XSEM XSEN XSHA1 XSHA256 } \ cmd_validity ALLO < int [ char R int ] > \

cmd_validity EPSV < [ { char 12 | char A char L char L } ] > \ cmd_validity MACB <
string > \

cmd_validity MDTM < [ date nnnnnnnnnnnnnn[.n[n[n]]] ] string > \ cmd_validity MODE <
char ASBCZ > \

cmd_validity PORT < host_port > \ cmd_validity PROT < char CSEP > \ cmd_validity STRU
< char FRPO [ string ] > \

cmd_validity TYPE < { char AE [ char NTC ] | char I | char L [ number ] } >

preprocessor ftp_telnet_protocol: ftp client default \ max_resp_len 256 \

bounce yes \ ignore_telnet_erase_cmds yes \ telnet_cmds yes

# SMTP normalization and anomaly detection.    For more information, see README.SMTP

preprocessor smtp: ports { 25 465 587 691 } \ inspection_type stateful \
b64_decode_depth 0 \

qp_decode_depth 0 \

bitenc_decode_depth 0 \

uu_decode_depth 0 \ log_mailfrom \ log_rcptto \ log_filename \ log_email_hdrs \
normalize cmds \

normalize_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY
} \

normalize_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND
SOML } \

normalize_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP X-
EXCH50 } \

normalize_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA
XTRN XUSR } \

max_command_line_len 512 \

max_header_line_len 1000 \

max_response_line_len 512 \ alt_max_command_line_len 260 { MAIL } \
alt_max_command_line_len 300 { RCPT } \

alt_max_command_line_len 500 { HELP HELO ETRN EHLO } \

alt_max_command_line_len 255 { EXPN VRFY ATRN SIZE BDAT DEBUG EMAL ESAM ESND ESOM EVFY
IDENT NOOP RSET } \
```

```
alt_max_command_line_len 246 { SEND SAML SOML AUTH TURN ETRN DATA RSET QUIT ONEX QUEU
STARTTLS TICK TIME TURNME VERB X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN
XLICENSE XQUE XSTA XTRN XUSR } \

valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY } \

valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML }
\

valid_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP X-EXCH50 }
\

valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN
XUSR } \

xlink2state { enabled }


# Portscan detection.      For more information, see README.sfportscan

# preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low }

# ARP spoof detection.     For more information, see the Snort Manual - Configuring
Snort - Preprocessors - ARP Spoof Preprocessor

# preprocessor arpspoof

# preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00


# SSH anomaly detection.   For more information, see README.ssh preprocessor ssh:
server_ports { 22 } \

autodetect \ max_client_bytes 19600 \

max_encrypted_packets 20 \

max_server_version_len 100 \ enable_respoverflow enable_ssh1crc32 \ enable_srvoverflow
enable_protomismatch

# SMB / DCE-RPC normalization and anomaly detection.  For more information, see
README.dcerpc2

preprocessor dcerpc2: memcap 102400, events [co ] preprocessor dcerpc2_server:
default, policy WinXP, \

detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593],

\

autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \ smb_max_chain 3,
smb_invalid_shares ["C$", "D$", "ADMIN$"]

# DNS anomaly detection.   For more information, see README.dns preprocessor dns:
ports { 53 } enable_rdata_overflow
```

```
# SSL anomaly detection and traffic bypass.    For more information, see README.ssl
preprocessor ssl: ports { 443 465 563 636 989 992 993 994 995 7801 7802
7900 7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913
7914 7915 7916 7917 7918 7919 7920 }, trustservers, noinspect_encrypted

# SDF sensitive data preprocessor.      For more information see README.sensitive_data
preprocessor sensitive_data: alert_threshold 25

# SIP Session Initiation Protocol preprocessor. For more information see README.sip
preprocessor sip: max_sessions 40000, \ ports { 5060 5061 5600 }, \
methods { invite \
cancel \ ack \
bye \ register \ options \ refer \
subscribe \ update \ join \
info \ message \ notify \ benotify \ do \
qauth \ sprack \ publish \ service \ unsubscribe \ prack }, \
max_uri_len 512, \
max_call_id_len 80, \
max_requestName_len 20, \
max_from_len 256, \
max_to_len 256, \
max_via_len 1024, \
max_contact_len 512, \
max_content_len 2048

# IMAP preprocessor. For more information see README.imap preprocessor imap: \
ports { 143 } \ b64_decode_depth 0 \
qp_decode_depth 0 \
bitenc_decode_depth 0 \
uu_decode_depth 0
```

```
# POP preprocessor. For more information see README.pop preprocessor pop: \
ports { 110 } \ b64_decode_depth 0 \
qp_decode_depth 0 \
bitenc_decode_depth 0 \
uu_decode_depth 0


# Modbus preprocessor. For more information see README.modbus preprocessor modbus:
ports { 502 }
# DNP3 preprocessor. For more information see README.dnp3 preprocessor dnp3: ports {
20000 } \
memcap 262144 \
check_crc


#
# Note to Debian users: this is disabled since it is an experimental
# preprocessor. If you want to use it you have to create the rules files
# referenced below in the /etc/snort/rules directory
#
# Reputation preprocessor. For more information see README.reputation
#preprocessor reputation: \
#      memcap 500, \
#      priority whitelist, \
#      nested_ip inner, \
#      whitelist $WHITE_LIST_PATH/white_list.rules, \
#      blacklist $BLACK_LIST_PATH/black_list.rules


##################################################
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
##################################################
```

```
# unified2

# Recommended for most installs

# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types,
vlan_event_types

#output unified2: filename snort.log, limit 128, nostamp, mpls_event_types,
vlan_event_types

output unified2: filename /var/log/snort/snort.log, limit 128, mpls_event_types,
vlan_event_types


# Additional configuration for specific types of installs

# output alert_unified2: filename snort.alert, limit 128, nostamp

# output log_unified2: filename snort.log, limit 128, nostamp


# syslog

# output alert_syslog: LOG_AUTH LOG_ALERT


# pcap

# output log_tcpdump: tcpdump.log


# metadata reference data. do not modify these lines include classification.config

include reference.config


##################################################

# Step #7: Customize your rule set

# For more information, see Snort Manual, Writing Snort Rules

#

# NOTE: All categories are enabled in this conf file

##################################################


# Note to Debian users: The rules preinstalled in the system

# can be *very* out of date. For more information please read
```

```
# the /usr/share/doc/snort-rules-default/README.Debian file

#
# If you install the official VRT Sourcefire rules please review this
# configuration file and re-enable (remove the comment in the first line) those
# rules files that are available in your system (in the
/etc/snort/rules
# directory)

# site specific rules
include $RULE_PATH/local.rules

#include $RULE_PATH/app-detect.rules include $RULE_PATH/attack-responses.rules include
$RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules include $RULE_PATH/dos.rules
include $RULE_PATH/experimental.rules
#include $RULE_PATH/exploit-kit.rules include $RULE_PATH/exploit.rules
#include $RULE_PATH/file-executable.rules
#include $RULE_PATH/file-flash.rules
#include $RULE_PATH/file-identify.rules
#include $RULE_PATH/file-image.rules
```

```
#include $RULE_PATH/file-java.rules

#include $RULE_PATH/file-multimedia.rules

#include $RULE_PATH/file-office.rules

#include $RULE_PATH/file-other.rules

#include $RULE_PATH/file-pdf.rules include $RULE_PATH/finger.rules include
$RULE_PATH/ftp.rules include $RULE_PATH/icmp-info.rules include $RULE_PATH/icmp.rules
include $RULE_PATH/imap.rules

#include $RULE_PATH/indicator-compromise.rules

#include $RULE_PATH/indicator-obfuscation.rules

#include $RULE_PATH/indicator-scan.rules

#include $RULE_PATH/indicator-shellcode.rules include $RULE_PATH/info.rules

#include $RULE_PATH/malware-backdoor.rules

#include $RULE_PATH/malware-cnc.rules

#include $RULE_PATH/malware-other.rules

#include $RULE_PATH/malware-tools.rules include $RULE_PATH/misc.rules

include $RULE_PATH/multimedia.rules include $RULE_PATH/mysql.rules include
$RULE_PATH/netbios.rules include $RULE_PATH/nntp.rules include $RULE_PATH/oracle.rules

#include $RULE_PATH/os-linux.rules

#include $RULE_PATH/os-mobile.rules

#include $RULE_PATH/os-other.rules

#include $RULE_PATH/os-solaris.rules

#include $RULE_PATH/os-windows.rules include $RULE_PATH/other-ids.rules include
$RULE_PATH/p2p.rules

#include $RULE_PATH/phishing-spam.rules

#include $RULE_PATH/policy-multimedia.rules

#include $RULE_PATH/policy-other.rules include $RULE_PATH/policy.rules

#include $RULE_PATH/policy-social.rules

#include $RULE_PATH/policy-spam.rules include  $RULE_PATH/pop2.rules include
$RULE_PATH/pop3.rules

#include $RULE_PATH/protocol-dns.rules

#include $RULE_PATH/protocol-finger.rules

#include $RULE_PATH/protocol-ftp.rules
```

```
#include $RULE_PATH/protocol-icmp.rules
#include $RULE_PATH/protocol-imap.rules
#include $RULE_PATH/protocol-nntp.rules
#include $RULE_PATH/protocol-pop.rules
#include $RULE_PATH/protocol-rpc.rules
#include $RULE_PATH/protocol-scada.rules
#include $RULE_PATH/protocol-services.rules
#include $RULE_PATH/protocol-snmp.rules
#include $RULE_PATH/protocol-telnet.rules
#include $RULE_PATH/protocol-tftp.rules
#include $RULE_PATH/protocol-voip.rules
#include $RULE_PATH/pua-adware.rules
#include $RULE_PATH/pua-other.rules
#include $RULE_PATH/pua-p2p.rules
#include $RULE_PATH/pua-toolbars.rules include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
#include $RULE_PATH/scada.rules include $RULE_PATH/scan.rules
#include $RULE_PATH/server-apache.rules
#include $RULE_PATH/server-iis.rules
#include $RULE_PATH/server-mail.rules
#include $RULE_PATH/server-mssql.rules
#include $RULE_PATH/server-mysql.rules
#include $RULE_PATH/server-oracle.rules
#include $RULE_PATH/server-other.rules
#include $RULE_PATH/server-samba.rules
#include $RULE_PATH/server-webapp.rules
#
# Note: These rules are disable by default as they are
# too coarse grained. Enabling them causes a large
# performance impact
```

```
#include $RULE_PATH/shellcode.rules include $RULE_PATH/smtp.rules include
$RULE_PATH/snmp.rules

#include $RULE_PATH/specific-threats.rules

#include $RULE_PATH/spyware-put.rules include $RULE_PATH/sql.rules

include $RULE_PATH/telnet.rules include $RULE_PATH/tftp.rules include
$RULE_PATH/virus.rules

#include $RULE_PATH/voip.rules

#include $RULE_PATH/web-activex.rules include $RULE_PATH/web-attacks.rules

include $RULE_PATH/web-cgi.rules include $RULE_PATH/web-client.rules include
$RULE_PATH/web-coldfusion.rules include $RULE_PATH/web-frontpage.rules include
$RULE_PATH/web-iis.rules include $RULE_PATH/web-misc.rules include $RULE_PATH/web-
php.rules include $RULE_PATH/x11.rules

include $RULE_PATH/community-sql-injection.rules include $RULE_PATH/community-web-
client.rules include $RULE_PATH/community-web-dos.rules include $RULE_PATH/community-
web-iis.rules include $RULE_PATH/community-web-misc.rules include
$RULE_PATH/community-web-php.rules include $RULE_PATH/community-sql-injection.rules
include $RULE_PATH/community-web-client.rules include $RULE_PATH/community-web-
dos.rules include $RULE_PATH/community-web-iis.rules include $RULE_PATH/community-web-
misc.rules include $RULE_PATH/community-web-php.rules

##################################################

# Step #8: Customize your preprocessor and decoder alerts

# For more information, see README.decoder_preproc_rules

##################################################


# decoder and preprocessor event rules

# include $PREPROC_RULE_PATH/preprocessor.rules

# include $PREPROC_RULE_PATH/decoder.rules

# include $PREPROC_RULE_PATH/sensitive-data.rules


##################################################

# Step #9: Customize your Shared Object Snort Rules

# For more information, see

http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-ru les.html

##################################################
```

```
# dynamic library rules
# include $SO_RULE_PATH/bad-traffic.rules
# include $SO_RULE_PATH/chat.rules
# include $SO_RULE_PATH/dos.rules
# include $SO_RULE_PATH/exploit.rules
# include $SO_RULE_PATH/icmp.rules
# include $SO_RULE_PATH/imap.rules
# include $SO_RULE_PATH/misc.rules
# include $SO_RULE_PATH/multimedia.rules
# include $SO_RULE_PATH/netbios.rules
# include $SO_RULE_PATH/nntp.rules
# include $SO_RULE_PATH/p2p.rules
# include $SO_RULE_PATH/smtp.rules
# include $SO_RULE_PATH/snmp.rules
# include $SO_RULE_PATH/specific-threats.rules
# include $SO_RULE_PATH/web-activex.rules
# include $SO_RULE_PATH/web-client.rules
# include $SO_RULE_PATH/web-iis.rules
# include $SO_RULE_PATH/web-misc.rules


# Event thresholding or suppression commands. See threshold.conf include
threshold.conf
```

/etc/snort/snort.debian.conf

```
# snort.debian.config (Debian Snort configuration file)
#
# This file was generated by the post-installation script of the snort
# package using values from the debconf database.
#
# It is used for options that are changed by Debian to leave
```

```
# the original configuration files untouched.
#
# This file is automatically updated on upgrades of the snort package
# *only* if it has not been modified since the last upgrade of that package.
#
# If you have edited this file but would like it to be automatically updated
# again, run the following command as root:
#      dpkg-reconfigure snort


DEBIAN_SNORT_STARTUP="boot" DEBIAN_SNORT_HOME_NET="172.16.0.0/16"
DEBIAN_SNORT_OPTIONS="" DEBIAN_SNORT_INTERFACE="eth0" DEBIAN_SNORT_SEND_STATS="true"
DEBIAN_SNORT_STATS_RCPT="root" DEBIAN_SNORT_STATS_THRESHOLD="1"
```

/usr/local/etc/barnyard2.conf

Also linked from */etc/snort/barnyard.conf*.

```
#
#      Barnyard2 example configuration file
#
#
# This file contains a sample barnyard2 configuration.
# You can take the following steps to create your own custom configuration:
#
#      1) Configure the variable declarations
#      2) Setup the input plugins
#      3) Setup the output plugins
#


#
# Step 1: configure the variable declarations
#
```

```
# in order to keep from having a commandline that uses every letter in the

# alphabet most configuration options are set here.


# use UTC for timestamps

#

#config utc


# set the appropriate paths to the file(s) your Snort process is using.

#

config reference_file:     /etc/snort/etc/reference.config config classification_file:
/etc/snort/etc/classification.config config gen_file: /etc/snort/gen-msg.map

config sid_file:    /etc/snort/etc/sid-msg.map


# Configure signature suppression at the spooler level see doc/README.sig_suppress

#

#

#config sig_suppress: 1:10


# Set the event cache size to defined max value before recycling of event occur.

#

#

#config event_cache_size: 4096

# define dedicated references similar to that of snort.

#

#config reference: mybugs http://www.mybugs.com/?s=


# define explicit classifications similar to that of snort.

#

#config classification: shortname, short description, priority


# set the directory for any output logging
```

```
#
config logdir: /var/log/barnyard2

# to ensure that any plugins requiring some level of uniqueness in their output
# the alert_with_interface_name, interface and hostname directives are provided.
# An example of usage would be to configure them to the values of the associated
# snort process whose unified files you are reading.
#
# Example:
#      For a snort process as follows:
#      snort -i eth0 -c /etc/snort.conf
#
#      Typical options would be:
#      config hostname:    thor
#      config interface: eth0
#      config alert_with_interface_name
#
config hostname:    snort config interface:    eth0
# enable printing of the interface name when alerting.
#
#config alert_with_interface_name

# at times snort will alert on a packet within a stream and dump that stream to
# the unified output. barnyard2 can generate output on each packet of that
# stream or the first packet only.
#
#config alert_on_each_packet_in_stream

# enable daemon mode
#
```

```
config daemon

# make barnyard2 process chroot to directory after initialisation.
#
#config chroot: /var/spool/barnyard2

# specifiy the group or GID for barnyard2 to run as after initialisation.
#
#config set_gid: 999

# specifiy the user or UID for barnyard2 to run as after initialisation.
#
#config set_uid: 999

# specify the directory for the barnyard2 PID file.
#
#config pidpath: /var/run/by2.pid

# enable decoding of the data link (or second level headers).
#
#config decode_data_link

# dump the application data
#
#config dump_payload

# dump the application data as chars only
#
#config dump_chars_only
```

```
# enable verbose dumping of payload information in log style output plugins.
#
#config dump_payload_verbose

# enable obfuscation of logged IP addresses.
#
#config obfuscate

# enable the year being shown in timestamps
#
config show_year

# set the umask for all files created by the barnyard2 process (eg. log files).
#
#config umask: 066

# enable verbose logging
#
#config verbose

# quiet down some of the output
#
#config quiet

# define the full waldo filepath.
#
config waldo_file: /tmp/waldo

# specificy the maximum length of the MPLS label chain
#
```

```
#config max_mpls_labelchain_len: 64

# specify the protocol (ie ipv4, ipv6, ethernet) that is encapsulated by MPLS.
#
#config mpls_payload_type: ipv4

# set the reference network or homenet which is predominantly used by the
# log_ascii plugin.
#
#config reference_net: 192.168.0.0/24

#
# CONTINOUS MODE
#

# set the archive directory for use with continous mode
#
#config archivedir: /tmp

# when in operating in continous mode, only process new records and ignore any
# existing unified files
#
#config process_new_records_only

#
# Step 2: setup the input plugins
#

# this is not hard, only unified2 is supported ;)
input unified2
```

```
#
# Step 3: setup the output plugins#


# alert_cef
#
----------------------------------------------------------------------
#
# Purpose:
#      This output module provides the abilty to output alert information to a
# remote network host as well as the local host using the open standard
# Common Event Format (CEF).
#
# Arguments: host=hostname[:port], severity facility
#      arguments should be comma delimited.
#      host   - specify a remote hostname or IP with optional port number
#      this is only specific to WIN32 (and is not yet fully supported)
#      severity      - as defined in RFC 3164 (eg. LOG_WARN, LOG_INFO)
#      facility      - as defined in RFC 3164 (eg. LOG_AUTH, LOG_LOCAL0)
#
# Examples:
#      output alert_cef
#      output alert_cef: host=192.168.10.1
#      output alert_cef: host=sysserver.com:1001
#      output alert_cef: LOG_AUTH LOG_INFO
#


# alert_bro
#
----------------------------------------------------------------------
```

```
#
# Purpose: Send alerts to a Bro-IDS instance.
#
# Arguments: hostname:port
#
# Examples:
#      output alert_bro: 127.0.0.1:47757


# alert_fast
#
------------------------------------------------------------------------
# Purpose: Converts data to an approximation of Snort's "fast alert" mode.
#
# Arguments: file <file>, stdout
#      arguments should be comma delimited.
#      file - specifiy alert file
#      stdout - no alert file, just print to screen
#
# Examples:
#      output alert_fast
#      output alert_fast: stdout
#
#output alert_fast: stdout
output alert_fast: /var/log/snort/alert


# prelude: log to the Prelude Hybrid IDS system
#
------------------------------------------------------------------------
#
# Purpose:
```

```
#      This output module provides logging to the Prelude Hybrid IDS system
#
# Arguments: profile=snort-profile
#      snort-profile - name of the Prelude profile to use (default is snort).
#
# Snort priority to IDMEF severity mappings:
# high < medium < low < info
#
# These are the default mapped from classification.config:
# info = 4
# low  = 3
# medium = 2
# high = anything below medium
#
# Examples:
#      output alert_prelude
#      output alert_prelude: profile=snort-profile-name
#


# alert_syslog
#
--------------------------------------------------------------------
#
# Purpose:
#      This output module provides the abilty to output alert information to local
syslog
#
#      severity     - as defined in RFC 3164 (eg. LOG_WARN, LOG_INFO)
#      facility     - as defined in RFC 3164 (eg. LOG_AUTH, LOG_LOCAL0)
#
# Examples:
```

```
#      output alert_syslog

#      output alert_syslog: LOG_AUTH LOG_INFO

#

output alert_syslog: LOG_AUTH LOG_INFO


# syslog_full

#--------------------------------

# Available as both a log and alert output plugin. Used to output data via TCP/UDP or
LOCAL ie(syslog())

# Arguments:

#      sensor_name $sensor_name   - unique sensor name

#      server $server      - server the device will report to

#      local  - if defined, ignore all remote information and use syslog() to send
message.

#      protocol $protocol  - protocol device will report over (tcp/udp)

#      port $port    - destination port device will report to (default: 514)

#      delimiters $delimiters           - define a character that will delimit message
sections ex: "|", will use | as message section delimiters. (default: |)

#    separators $separators      - define field separator included in each message ex:
" " , will use space as field separator. (default: [:space:])

#      operation_mode $operaion_mode     - default | complete : default mode is
compatible with default snort syslog message, complete prints more information such as
the raw packet (hexed)

#      log_priority  $log_priority - used by local option for syslog priority call. (man
syslog(3) for supported options) (default: LOG_INFO)

#      log_facility  $log_facility - used by local option for syslog facility call. (man
syslog(3) for supported options) (default: LOG_USER)

#      payload_encoding    - (default: hex)    support hex/ascii/base64 for
log_syslog_full using operation_mode complete only.


# Usage Examples:

# output alert_syslog_full: sensor_name snortIds1-eth2, server xxx.xxx.xxx.xxx,
protocol udp, port 514, operation_mode default

# output alert_syslog_full: sensor_name snortIds1-eth2, server xxx.xxx.xxx.xxx,
protocol udp, port 514, operation_mode complete
```

```
# output log_syslog_full: sensor_name snortIds1-eth2, server xxx.xxx.xxx.xxx, protocol
udp, port 514, operation_mode default

# output log_syslog_full: sensor_name snortIds1-eth2, server xxx.xxx.xxx.xxx, protocol
udp, port 514, operation_mode complete

# output alert_syslog_full: sensor_name snortIds1-eth2, server xxx.xxx.xxx.xxx,
protocol udp, port 514

# output log_syslog_full: sensor_name snortIds1-eth2, server xxx.xxx.xxx.xxx, protocol
udp, port 514

# output alert_syslog_full: sensor_name snortIds1-eth2, local

# output log_syslog_full: sensor_name snortIds1-eth2, local, log_priority
LOG_CRIT,log_facility LOG_CRON


# log_ascii

#

--------------------------------------------------------------------

#

# Purpose: This output module provides the default packet logging funtionality

#

# Arguments: None.

#

# Examples:

#      output log_ascii

#

output log_ascii


# log_tcpdump

#

--------------------------------------------------------------------

#

# Purpose

#      This output module logs packets in binary tcpdump format

#
```

```
# Arguments:
#      The only argument is the output file name.
#
# Examples:
#      output log_tcpdump: tcpdump.log
#
output log_tcpdump: /var/log/snort/tcpdump.log
# sguil
#
--------------------------------------------------------------------
#
# Purpose: This output module provides logging ability for the sguil interface
# See doc/README.sguil
#
# Arguments: agent_port <port>, sensor_name <name>
#      arguments should be comma delimited.
#      agent_port    - explicitly set the sguil agent listening port
#      (default: 7736)
#      sensor_name - explicitly set the sensor name
#      (default: machine hostname)
#
# Examples:
#      output sguil
#      output sguil: agent_port=7000
#      output sguil: sensor_name=argyle
#      output sguil: agent_port=7000, sensor_name=argyle
#


# database: log to a variety of databases
#
```

```
------------------------------------------------------------------
#
# Purpose: This output module provides logging ability to a variety of databases
# See doc/README.database for additional information.
#
# Examples:
#      output database: log, mysql, user=root password=test dbname=db host=localhost
#      output database: alert, postgresql, user=snort dbname=snort
#      output database: log, odbc, user=snort dbname=snort
#      output database: log, mssql, dbname=snort user=snort password=test
#      output database: log, oracle, dbname=snort user=snort password=test
#
#output database: log, mysql, user=root password=1Password! dbname=snortdb

# alert_fwsam: allow blocking of IP's through remote services
#
------------------------------------------------------------------
# output alert_fwsam: <SnortSam Station>:<port>/<key>
#
#      <FW Mgmt Station>:  IP address or host name of the host running SnortSam.
#      <port>:       Port the remote SnortSam service listens on (default 898).
#      <key>: Key used for authentication (encryption really)
#      of the communication to the remote service.
#
# Examples:
#
# output alert_fwsam: snortsambox/idspassword
# output alert_fwsam: fw1.domain.tld:898/mykey
# output alert_fwsam: 192.168.0.1/borderfw     192.168.1.254/wanfw
#
```

/opt/splunkforwarder/etc/system/local/server.conf

```
[sslConfig]

sslKeysfilePassword = $1$A0zU/599eO4g


[lmpool:auto_generated_pool_forwarder] description = auto_generated_pool_forwarder
quota = MAX

slaves = *

stack_id = forwarder


[lmpool:auto_generated_pool_free] description = auto_generated_pool_free quota = MAX

slaves = * stack_id = free

[general]

pass4SymmKey = $1$VACAo9o7M7wg serverName = snort
```

/opt/splunkforwarder/etc/system/local/inputs.conf

Note: The **sourcetype=snort_alert_full** is important if you are using the Splunk TA_Snort app.

```
[default] host=snort

sourcetype=snort_alert_full index=snort

[monitor:///var/log/snort/alert] sourcetype=snort_alert_full
```

/opt/splunkforwarder/etc/system/local/outputs.conf

```
[tcpout]

defaultGroup = splunkssl


[tcpout:splunkssl] server = loghost:9997 compressed = true

sslVerifyServerCert = false

sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem sslCertPath =
$SPLUNK_HOME/etc/certs/snort.lab5.nccoe.gov.pem sslPassword = $1$cw==
```

3.9 Tyco Security Products

Tyco Security Products are used to integrate personnel access management into the FS ITAM build. The CCURE 9000 security and event management system allows integration with a variety of intrusion devices, allowing admins to monitor and perform intrusion detection within facilities to stop incidents of malicious activity or violation of policy. For the ITAM build, the focal point of the CCURE 9000 product is personnel and visitor management. The iSTAR Edge Door Controller provides features to secure any door, including clustering, door monitoring, and anti-passback.

3.9.1 Installing Tyco Security Products

Tyco Security Products hardware is received with pre-installed software. Hardware components received for this build include the following:

- host laptop
- iSTAR Edge Door Controller
- two badge readers
- three badges
- American Dynamics Video Edge Network Video Recorder (NVR)
- one camera
- NETGEAR ProSAFE switch
- Ethernet cables

Directions for connecting components will be included in the packaging on the iSTAR Edge Installation Reference disc. The host laptop will have the iSTAR Configuration Utility, CCURE 9000, License Manager, KeyCodeGenerator, and Victor Management Software installed and pre-configured. The iSTAR Configuration Utility can be used to confirm IP addresses.

3.9.2 Configurations

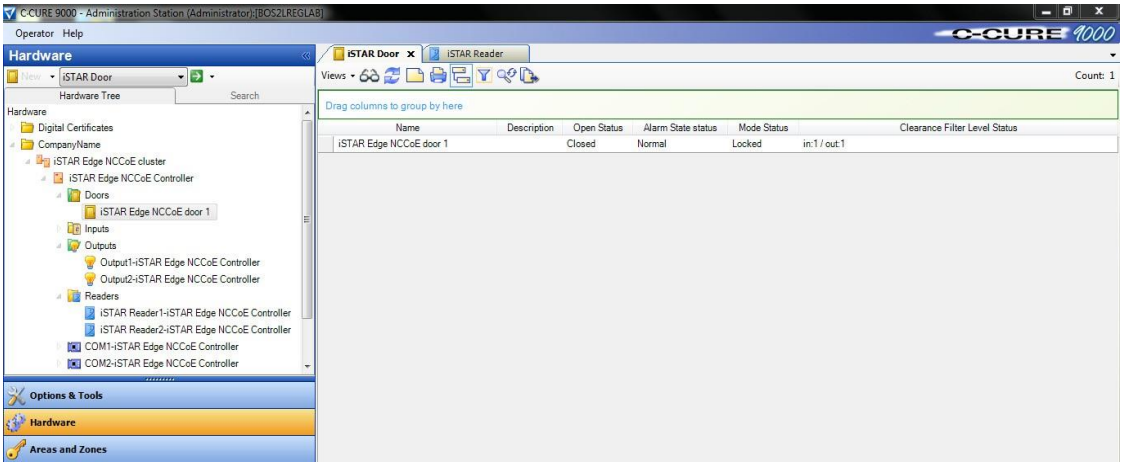
All components included with Tyco Security Products will be pre-configured. Configuration manuals are documented at the Tyco Security Products website as well as on the iSTAR Edge Installation Reference disc. In addition, the security product suite will be accompanied by a list of all static IP addresses to confirm or correct any configurations. Static IP addresses for the ITAM build are as follows:

- laptop (host): 192.168.1.167
- NVR: 192.168.1.178
- camera: 192.168.1.177
- iSTAR: 192.168.1.169

The three badges received are configured for the ITAM build. Two badges contain access rights, with a clearance, while one badge does not. Two door readers are configured as door controllers for one door. One reader is configured as the **IN** reader while the second is configured as the **OUT** reader. Badges must have a clearance to be admitted into the door.

Configurations for badges, doors and readers can be viewed and managed using CCURE 9000 software shown in Figure 3-1.

Figure 3-1 CCURE 9000 Overview



The host machine should then be connected to the ITAM network to integrate with the ITAM build. To prepare the host machine for integration with ITAM, SQL Server Management Studio must be installed. For the ITAM build, a query to the journal table is called by Splunk Enterprise to retrieve information, including the Cardholder Name, Door Name, Journal Log Message Type, Message Text and Message Date/Time. The information produced from CCURE is shown in Figure 3-2.

Figure 3-2 CCURE 9000 Messages

C•CURE 9000 **SWH13 - Personnel Admitted at Doors Report**

Journal

Cardholder Name	Door Name	Journal Log Message Type	Message Text	Message Date/Time
good, guy	iSTAR Edge NCCoE door 1	Card Admitted	Admitted 'good, guy' (Card: 16053) at 'iSTAR Edge NCCoE door 1' (IN) ([Unused]).	8/20/2015 12:55:14 PM
good, guy	iSTAR Edge NCCoE door 1	Card Admitted	Admitted 'good, guy' (Card: 16053) at 'iSTAR Edge NCCoE door 1' (OUT) ([Unused]).	8/20/2015 12:55:24 PM
good, guy II	iSTAR Edge NCCoE door 1	Card Admitted	Admitted 'good, guy II' (Card: 608) at 'iSTAR Edge NCCoE door 1' (IN) ([Unused]).	8/20/2015 12:56:06 PM
good, guy II	iSTAR Edge NCCoE door 1	Card Admitted	Admitted 'good, guy II' (Card: 608) at 'iSTAR Edge NCCoE door 1' (OUT)	8/20/2015 12:56:15 PM

The query ran for Splunk Enterprise to retrieve the information from the journal is as follows:

```
SELECT MessageType, MessageUTC, REPLACE(PrimaryObjectName,',',' ') AS
PrimaryObjectName, XmlMessage

FROM JournalLog WHERE MessageType='CardAdmitted' OR MessageType='CardRejected'
```

3.10 Windows Server Update Services (WSUS)

WSUS is integrated into Windows Server 2012 as a server role. WSUS enables IT administrators to deploy the latest Microsoft product updates to computers that are running the Windows operating system. Using WSUS, an administrator can fully manage the distribution of updates that are released through Microsoft Update to computers in their network.

3.10.1 How It's Used

The ITAM system is using WSUS for its reporting features. WSUS reports on the volume and status of software updates from Microsoft Update. ITAM uses this information to provide insight to administrators for analysis of which Windows machines in the network are not in compliance with the latest vulnerability patches and software updates.

3.10.2 Virtual Machine Configuration

The WSUS virtual machine is configured with one network interface card, 8 GB of RAM, one CPU core and 100 GB of hard drive space. The 100 GB of hard drive space is very important for this machine.

3.10.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Disabled
- IP Address: 172.16.0.45
- Netmask: 255.255.255.0
- Gateway: 172.16.0.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

3.10.4 Installing WSUS

WSUS is installed through the add roles and features wizard in Server Manager. Documentation is provided by Microsoft at: https://technet.microsoft.com/en-us/windowsserver/bb332157.aspx. WSUS should NOT be a member of your domain.

3.10.5 Configurations

You configure WSUS using the WSUS Server Configuration Wizard. When the wizard prompts you, set these options as follows:

- **Update Source and Proxy Server** – **Synchronize form Microsoft Update**
- **Products and Classifications** – **Microsoft SQL Server 2012, Microsoft SQL Server 2014, SQL Server 2008 R2, SQL Server 2008, SQL Server 2012 Product Updates for Setup, SQL server Feature Pack, Windows 7, Windows Server 2012 R2 and later drivers, Windows Server 2012 R2**
- **Update Files and Languages** – **Store update files locally on this server < Download update files to this server only when updates are approved, Download updates only in English**
- **Synchronization Schedule** – **Automatically > 1 per day**
- **Automatic Approvals** – Default
- **Computers** – **Use the Update Services console**
- **Reporting Rollup** – N/A
- **E-mail Notifications** – N/A
- **Personalization** – N/A

3.10.6 Configure Active Directory Server to Require WSUS

Clients are configured to get their Windows updates and patches through Group Policy on the Active Directory server.

Full documentation can be found at: https://technet.microsoft.com/en-us/library/Cc720539%28v=WS.10%29.aspx.

1. On the Active Directory Server:

   Administrative Tools > **Group Policy Management**

2. Under your domain, create a new group policy object by right-clicking and selecting **Create a GPO in this domain, and link it here**.

3. Then right-click the newly created GPO in the Group Policy Objects area of the Group Policy Management window and select **Edit**.

4. In the **Group Policy Management Editor** expand **Computer Configuration**, expand **Administrative Templates**, expand **Windows Components**, and then click **Windows Update**.
5. In the details pane, select **Specify intranet Microsoft update service location**.
6. Click **ENABLED** and enter the URL of the WSUS server and statistics server (they are the same for this build): **http://wsus.lab5.nccoe.gov:8530**.

3.10.7 Create WSUS Statistics for Splunk Enterprise

When WSUS is running and downloading updates (you can check this by running a report), you can work with assemblies using Windows PowerShell to connect to the WSUS server. With this connection, PowerShell script can be written to extract information from WSUS. The script creates two .CSV files with WSUS information that are forwarded to Splunk Enterprise. The script to accomplish this task is as follows:

1. Filename: WSUSReport.ps1

```
$wsus

$wsusserver = 'wsus'
```

2. Load required assemblies:

```
[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.A
dministration")| Out-Null

$wsus = [Microsoft.UpdateServices.Administration.AdminProxy]::getUpdateServer(
'wsus',$False,8530)
```

3. Create update scope object:

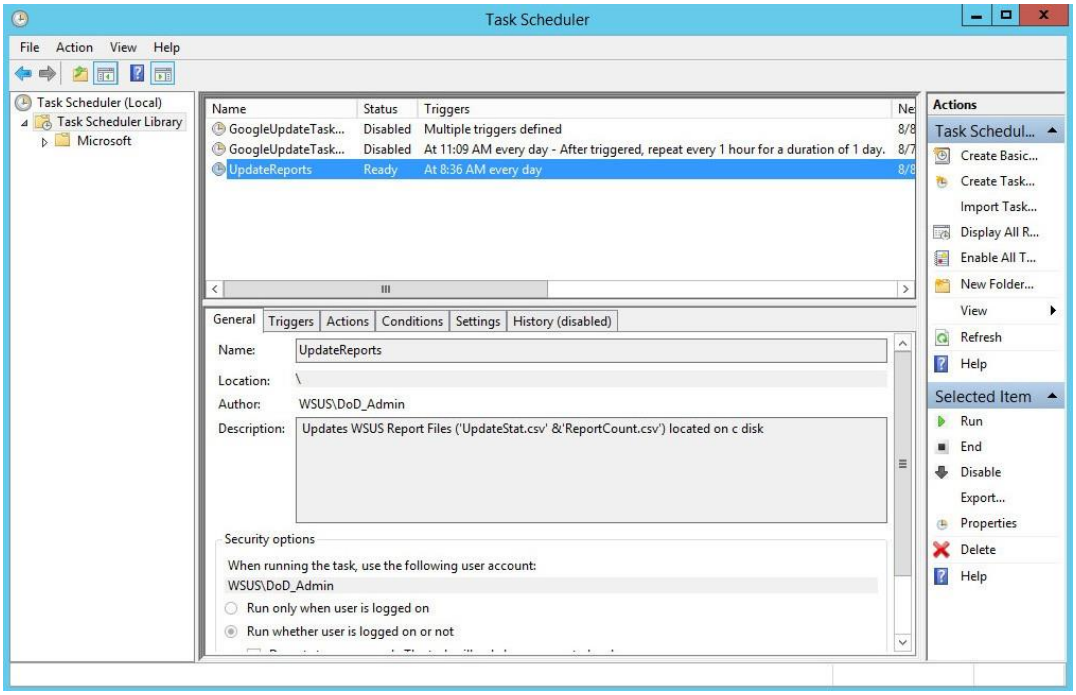
```
$updatescope = New-Object Microsoft.UpdateServices.Administration.UpdateScope

$updatescope.IncludedInstallationStates =
[Microsoft.UpdateServices.Administration.UpdateInstallationStates]::No
tInstalled

$updatescope.FromArrivalDate = [datetime]"12/13/2011"

$computerscope = New-Object
Microsoft.UpdateServices.Administration.ComputerTargetScope

$wsus.GetSummariesPerComputerTarget($updatescope,$computerscope) | Select

@{L='ComputerTarget';E={($wsus.GetComputerTarget([guid]$_.ComputerTarg
etId)).FullDomainName}},

@{L='NeededCount';E={($_.DownloadedCount+$_.NotInstalledCount)}},Downl
oadedCount,NotInstalledCount,InstalledCount,FailedCount | Export-Csv
c:\ReportCount.csv
```

```
$wsus.GetUpdateApprovals($updatescope) | Select
@{L='ComputerTargetGroup';E={$_.GetComputerTargetGroup().Name}},
@{L='UpdateTitle';E={($wsus.GetUpdate([guid]$_.UpdateId.UpdateId.Guid)
).Title}}, GoLiveTime,AdministratorName,Deadline | Export-Csv c:\UpdateStat.csv
```

This script creates two **.CSV** files and places them on the **C** drive: **ReportCount.csv** and **UpdateStat.csv**. These two files contain the fields ComputerTarget, NeededCount, DownloadedCount, NotInstalledCount, InstalledCount, FailedCount; and ComputerTargetGroup, UpdateTitle, GoLiveTime, AdministratorName and Deadline, respectively.

When the script is running error free, a task is scheduled for the script to run daily for updates to the data. To create a scheduled task, complete the following steps:

1. Open Task Scheduler and select **Create Task**.
2. Name the task and give it a description. Select **Run whether user is logged on or not**. Select **Run with highest privileges**. Configure for: **Windows Server 2012 R2**.
3. Select the **Triggers** tab and select **New**. Create a trigger to run every day at the desired time.
4. Select the **Actions** tab and select **New**. Under **Action**, select **Start a Program**. In the Program/script box, enter **c:\Windows\System32\WindowsPowershell\v1.0\powershell.exe** or browse for the PowerShell executable.
5. In the arguments box insert **-ExecutionPolicy Bypass <locationofscript>**. Select **OK** to save the task.
6. Use the defaults for the remaining settings. The scheduled task should look similar to the task highlighted in the following figure.



3.10.8 Installing Splunk Universal Forwarder

Note: You will need a Splunk account to download the Splunk Universal Forwarder. It is free and can be set up at: https://www.splunk.com/page/sign_up.

1. Download the Splunk Universal Forwarder from: http://www.splunk.com/en_us/download/universal-forwarder.html.
2. You want the latest version for `OS version Windows (64-bit)`. Since this is installing on Windows, select the file that ends in `.msi`. An example is:

   ```
   splunkforwader-6.2.5-272645-x64-release.msi
   ```

   Detailed installation instructions can be found at: http://docs.splunk.com/Documentation/Splunk/6.2.3/Forwarding/DeployaWindowsdfmanually#Install_the_universal_forwarder.

3.10.9 Configuring Splunk Universal Forwarder

Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You will also need a copy of your certificate authority's public certificate.

If you entered your certificates during install time, they will be located at:

C:\Program Files\SplunkUniversalForwarder\etc\auth

If not, you will need to manually copy your certificates here.

1. Copy Splunk Universal Forwarder configuration files:

   ```
   copy <server.conf> C:\Program Files\SplunkUniversalForwarder\etc\system\local
   copy <inputs.conf> C:\Program Files\SplunkUniversalForwarder\etc\system\local
   copy <outputs.conf> C:\Program Files\SplunkUniversalForwarder\etc\system\local
   ```

2. Modify **server.conf** so that:

   ServerName=WSUS is your hostname.

   ```
   sslKeysfilePassword = <password for your private key>
   ```

3. Modify **outputs.conf** so that:

   Server = loghost:9997 is your correct Splunk Enterprise server/indexer and port.

   ```
   sslPassword = <password of your certificate private key>
   ```

Note: This will be hashed and not clear text after a restart.

Inputs.conf should work, but you are free to modify it to include the Windows logs that you are interested in.

3.10.10 Configurations and Scripts

C:\Program Files\SplunkUniversalForwarder\etc\system\local server.conf

```
[sslConfig]
sslKeysfilePassword = $1$sznWu23zCGHY


[general]
pass4SymmKey = $1$5HWC5yi1QzPY serverName = WSUS
[lmpool:auto_generated_pool_forwarder] description = auto_generated_pool_forwarder
quota = MAX
slaves = *
stack_id = forwarder
```

```
[lmpool:auto_generated_pool_free] description = auto_generated_pool_free quota = MAX

slaves = * stack_id = free
```

C:\Program Files\SplunkUniversalForwarder\etc\system\local\inputs.conf

```
[default] host = WSUS

sourcetype = wsus

index = wsus


[script://$SPLUNK_HOME\bin\scripts\splunk-wmi.path] disabled = 0

[monitor:///C:\ReportCount.csv] sourcetype=wsus_reportcount

crcSalt is needed because this file doesn't change much and is small crcSalt =
<SOURCE>

ignoreOlderThan = 2d disabled = 0

[monitor:///C:\UpdateStat.csv ] sourcetype=wsus_updatestat ignoreOlderThan = 2d

disabled = 0
```

C:\Program Files\SplunkUniversalForwarder\etc\system\local\outputs.conf

```
[tcpout]

defaultGroup = default-autolb-group


[tcpout:default-autolb-group] server = loghost:9997

[tcpout-server://loghost:9997] sslCertPath = C:\wsus.lab5.nccoe.gov.pem sslPassword =
$1$sznWu23zCGHY

sslRootCAPath = C:\Users\DoD_Admin\Downloads\CAServerCert.pem
```

4 Tier 3

4.1 Active Directory Server

The Active Directory server in the ITAM build uses an NCCoE base 2012 R2 x86_64 DoD STIG image. The installation of the Windows Active Directory server was performed using installation media provided by DISA. This image was chosen because it is standardized, hardened, and fully documented.

4.1.1 Software Configurations

4.1.1.1 Windows 2012 Active Directory Server

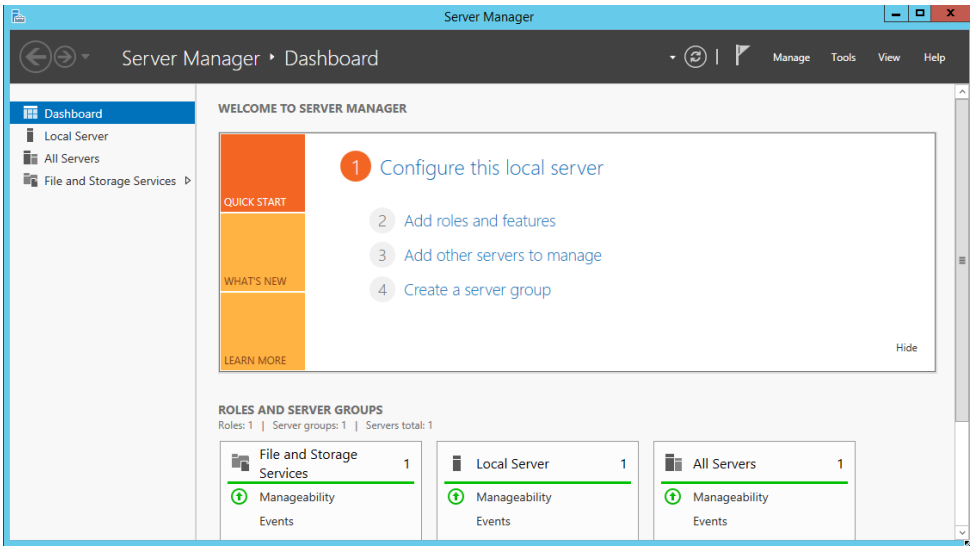
Active Directory provides centralized management, authentication, security, and information storage for end devices and users in a networked environment.

4.1.2 How It's Used

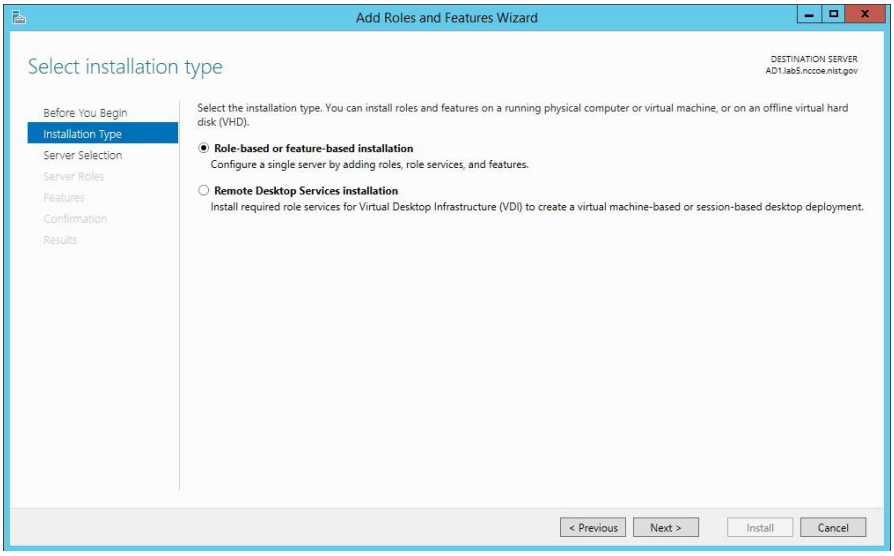
The Active Directory service is used in the ITAM build to provide authentication, user management and security within a mixed environment with Windows and Linux endpoints.

4.1.3 Installation

1. Go to Server Manager and click **Add Roles and Features Wizard**.

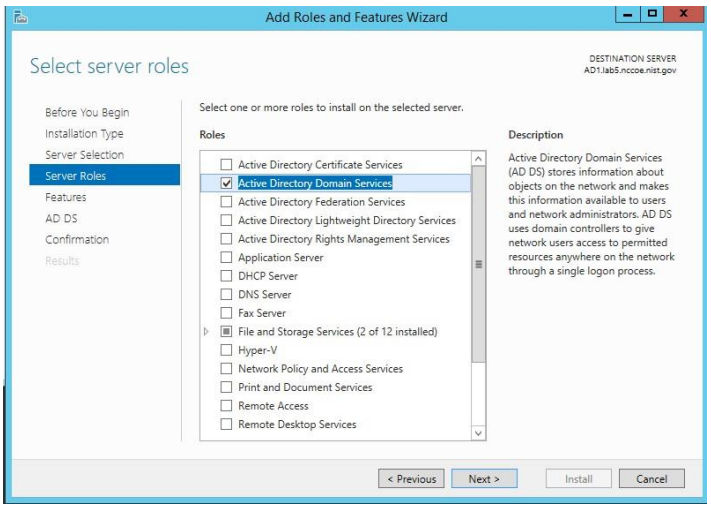


2. Click **Next** and select **Role-based or feature-based installation**. Then, click **Next**.



3. Ensure that the appropriate server name is selected. Then, click **Next**.

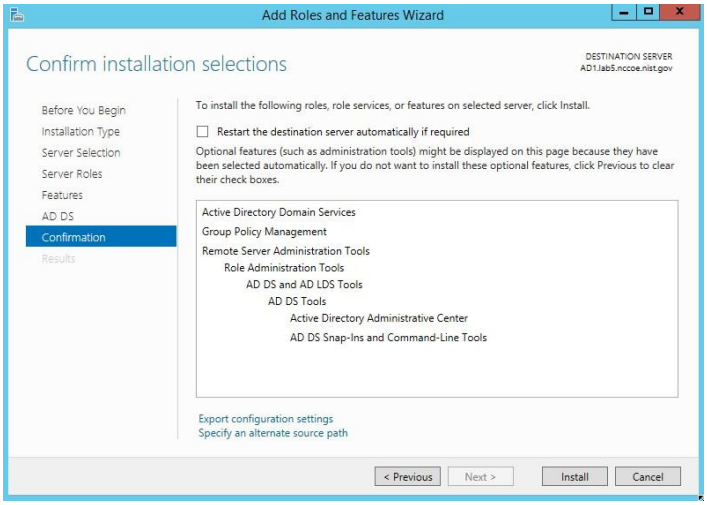
4. Click the checkbox next to **Active Directory Domain Services**. Then click **Next** to advance to the next screen. Then, click **Add Features**.



5. Use the features selected by default. Then, click **Next**.

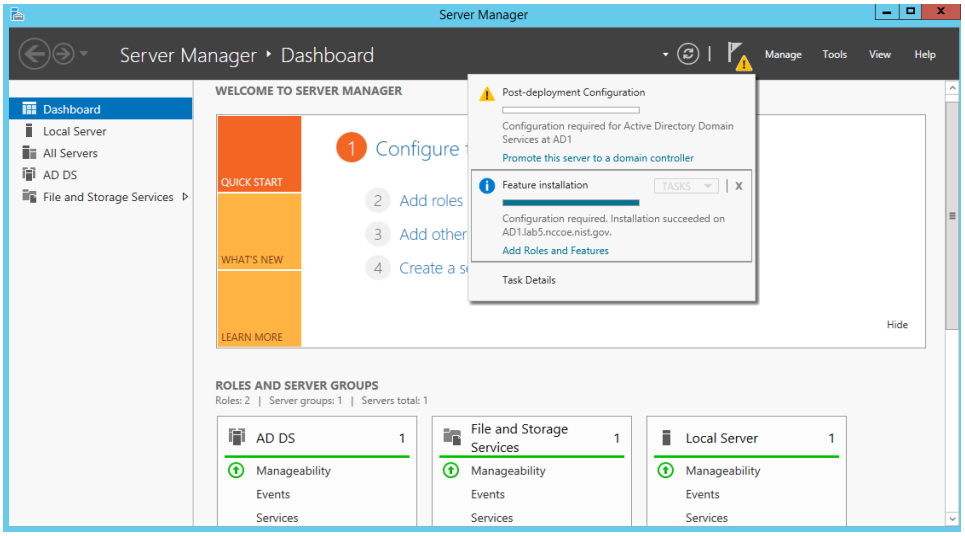
6. In the Active Directory Domain Services screen, click **Next**.

7. On the Confirm installations selections screen, click **Install**.



8. When you see the message that the installation was successful, click **close**.

9. Return to the Server Manager and click on the yellow warning message.



10. On the Post-deployment Configuration box, click **Promote this server to a domain controller**.

11. Choose **Add a new forest**, specify the root domain name and click **Next**.

12. Use the default settings in the Domain Controller Options page. Ensure that **DNS server** is selected. Enter the **Directory Services Restore Mode** password and click **Next**.

13. Choose a **NetBIOS domain Name** and click **Next**.

14. Accept the default locations for **AD DS**, **DS Database**, **log files** and **SYSVOL**.
15. In the Review Options screen, click **Next**.
16. Allow the system to complete the prerequisites check and click **Install**.
17. When the installation completes, reboot the system.

4.2 AssetCentral

AssetCentral is an IT infrastructure management system that stores and displays information related to physical assets including location, make, model, and serial number. AssetCentral can help run an entire data center by monitoring weight, utilization, available space, heat and power distribution. AssetCentral is installed on a CentOS7 system.

4.2.1 How It's Used

In the FS ITAM build AssetCentral is used to provide physical asset location. AssetCentral provides the building, room and rack of an asset.

4.2.2 Virtual Machine Configuration

The virtual machine is configured with 1 network interface cards, 4 GB of RAM and 1 CPU cores.

4.2.3 Network Configuration

The management network interface card is configured as such:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.1.50
- Netmask: 255.255.255.0
- Gateway: 172.16.1.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

4.2.4 Installing AssetCentral

AssetCentral is installed on a hardened CentOS7 Linux system. AssetCentral requires PHP, Web Server (Apache) and MySQL database to be installed.

Table 4-1 Recommended Versions for AssetCentral – Tier 3

Vendor	Product	Version
RedHat	Enterprise Linux Server	Release 6.4 (Santiago) (x86_64)
Apache	Web Server	httpd-2.2.15-26.el6.x86_64
mysql	Server	5.1.6.6
php		5.3.3 or higher

4.2.5 Installing MySQL (MariaDB)

```
# yum -y install mariadb-server mariadb

#systemctl start mariadb.service

#systemctl enable mariadb.service

# mysql_secure_installation
```

1. Answer the questions with the default answers while performing the mysql_secure_installation.
2. Create a database – assetcentral.
3. Create a user – assetcentral.
4. Grant all privileges to assetcentral user.

4.2.6 Installing Apache

```
# yum -y install httpd

#systemctl start httpd.service

#systemctl enable httpd.service

#firewall-cmd --permanent --zone=public --add-service=http

#firewall-cmd --permanent --zone=public --add-service=https

#firewall-cmd -reload
```

4.2.6.1 HTTP Configuration

1. Go to HTTPD root; normally (*/etc/httpd*).
2. Under the modules directory, make sure *libphp5.so* exists.
3. Change documentroot (webroot) as per environment in *httpd.conf*.

4.2.7 Installing PHP5

```
#yum -y install php

#systemctl restart httpd.service

#yum search php

#yum -y install php-mysql

#yum -y install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap curl curl-devel
```

1. Restart Apache:

   ```
   #systemctl restart httpd.service
   ```

4.2.8 Post Installation Tasks

1. Copy AssetCentral files and folders from previous install to the new webroot.
2. Under the location (*../assetcentral/application/config*) make necessary changes as per environment.

4.2.8.1 Sample

```
<?php defined('ASSET_CENTRAL')ordie(''); define('AC_URL_SUBDIR','/acprod');
define('AC_URL_SCRIPT','/index.php'); define('AC_URL_PARAM','go');
define('AC_URL_PREFIX',AC_URL_SUBDIR . AC_URL_SCRIPT.'?'

. AC_URL_PARAM . '='); define('AC_ERROR_REPORTING',E_ERROR);

//no slash at the end of this url define('URL_SITE','http://10.1.xx.xxx');
define('OS','NIX'); // *NIX WIN BSD MAC

//default database (read) define('DB_TYPE_READ','MYSQL');
define('DB_HOST_READ','127.0.0.1');

//usually leave this blank for MYSQL define('DB_PORT_READ','');
define('DB_USER_READ','assetcentral'); define('DB_PASS_READ','xxxxx');
define('DB_DATA_READ','asset_prod'); define('DB_PREFIX_READ','');
```

4.3 Email

Email is the email server for the FS-ITAM build.

4.3.1 How It's Used

In the FS ITAM build, Email provides all users with email.

4.3.2 Virtual Machine Configuration

The Email virtual machine is configured with one network interface card, 4 GB of RAM and one CPU core.

4.3.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.1.50
- Netmask: 255.255.255.0
- Gateway: 172.16.1.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

4.3.4 Installing Email

Email is installed on a hardened Ubuntu 14.04 Linux system. This email system is using the Postfix email program. Complete installation instructions can be found at: https://help.ubuntu.com/community/Postfix#Installation.

For Debian/Ubuntu Linux systems: It is always best to make sure your system is up-to-date by performing:

```
sudo apt-get update sudo apt-get upgrade
```

```
sudo apt-get install postfix
```

4.3.5 Configure Email

From a terminal prompt:

```
sudo dpkg-reconfigure postfix
```

General type of mail configuration: **Internet Site**

NONE doesn't appear to be requested in current config.

System mail name: **mail1.lab5.nccoe.gov**

Root and postmaster mail recipient: `<admin_user_name>`

Other destinations for mail: email1, email1.lab5.nccoe.gov, localhost.lab5.nccoe.gov, localhost.localdomain, localhost, lab5.nccoe.gov

Force synchronous updates on mail queue? No

Local networks: `172.16.0.0/16`

Yes doesn't appear to be requested in current config.

Mailbox size limit (bytes): `0`

Local address extension character: **+**

Internet protocols to use: `all`

Ensure that /etc/postfix/main.cf looks like the version below in the Configuration Files section (Section 4.3.8). Especially take note that the **inet_interfaces** setting. **inet_interfaces = loopback-only** will NOT allow mail from other machines.

4.3.6 User Accounts

1. Create an account for each user that needs email:

   ```
   adduser <username>
   ```

2. Answer the questions.

4.3.7 DNS Settings

For mail to work correctly, an MX record must be set up on the DNS server.

The FS-ITAM build is using a Microsoft Server 2012R2 as its DNS server.

1. First set up a DNS A-Record for the email server, which looks like:

   ```
   Host: email1
   ```

   ```
   FQDN: email1.lab5.nccoe.gov IP address: 172.16.1.50
   ```

2. Check next to Update associates pointer record.

3. Next create an MX record that looks like:

   ```
   Host or child domain: (same as parent folder)
   ```

   ```
   FQDN: lab5.nccoe.gov
   ```

   ```
   FQDN of mail server: email1.lab5.nccoe.gov
   ```

   ```
   Mail server priority: 10
   ```

4.3.8 Configuration Files

/etc/postfix/main.cf

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first

# line of that file to be used as the name. The Debian default

# is /etc/mailname.

#myorigin = /etc/mailname


smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu) biff = no

# appending .domain is the MUA's job. append_dot_mydomain = no


# Uncomment the next line to generate "delayed mail" warnings

#delay_warning_time = 4h readme_directory = no

# TLS parameters

smtpd_tls_cert_file = /etc/ssl/certs/smtpd.crt smtpd_tls_key_file =
/etc/ssl/private/smtpd.key smtpd_use_tls=yes

smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache

smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache


# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for

# information on enabling SSL in the smtp client.


smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
defer_unauth_destination

myhostname = mail1.lab5.nccoe.gov alias_maps = hash:/etc/aliases alias_database =
hash:/etc/aliases

mydestination = email1, email1.lab5.nccoe.gov, localhost.lab5.nccoe.gov,
localhost.localdomain, localhost, lab5.nccoe.gov

relayhost =
```

```
mynetworks = 172.16.0.0/16 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128

mailbox_size_limit = 0 recipient_delimiter = +

#inet_interfaces = loopback-only inet_interfaces = all default_transport = smtp
relay_transport = smtp

myorigin = /etc/mailname inet_protocols = all home_mailbox = Maildir/ mailbox_command
= smtpd_sasl_local_domain = smtpd_sasl_auth_enable = yes

smtpd_sasl_security_options = noanonymous broken_sasl_auth_clients = yes

smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination

smtp_tls_security_level = may smtpd_tls_security_level = may smtpd_tls_auth_only = no
smtp_tls_note_starttls_offer = yes smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem
smtpd_tls_loglevel = 1 smtpd_tls_received_header = yes smtpd_tls_session_cache_timeout
= 3600s tls_random_source = dev:/dev/urandom
```

4.4 Openswan (VPN)

Openswan is an open-source IPsec VPN. Openswan runs on Linux and supports IKEv1, IKEv2, X.509 Digital Certificates and NAT Traversal.

4.4.1 How It's Used

In the FS ITAM build, Openswan is used to form a secure VPN to the mainframe computer owned by Vanguard Integrity Professionals.

4.4.2 Virtual Machine Configuration

The Openswan virtual machine is configured with two network interface cards, 8 GB of RAM and one CPU core.

4.4.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.0.67 (internal interface)
- IP Address: 10.33.5.16 (external interface for the VPN) Netmask: 255.255.255.0
- Gateway: 10.33.5.1
- DNS Servers: 8.8.8.8, 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

4.4.4 Installing Openswan

Openswan is installed on a hardened Ubuntu 14.04 Linux system. Complete installation instructions can be found at https://www.openswan.org/.

4.4.5 Installing Openswan

1. For Debian/Ubuntu Linux systems: It is always best to make sure your system is up-to-date by performing:

   ```
   sudo apt-get update sudo apt-get upgrade

   sudo apt-get install openswan xl2tpd ppp lsof
   ```

2. Copy the provided configuration files into /etc:

   ```
   cp <ipsec.conf> /etc

   cp <ipsec.secrets> /etc
   ```

3. Edit */etc/ipsec.secrets* and replace **MYSECRET** with your pre-shared key.

4. Restart Openswan:

   ```
   service ipsec restart
   ```

5. Verify by running:

   ```
   service ipsec status
   ```

6. Bring up the IPsec tunnel:

   ```
   ipsec auto -up nccoe-vanguard
   ```

7. Verify by running:

   ```
   ipsec auto -verbose -status
   ```

   If you see **(ISAKMP SA established)** then that is good.

   A little script was created to keep the connection up - connect_vanguard.sh.

8. Copy `connect_vanguard.sh` somewhere typical like */usr/local/bin*:

   ```
   cp <connect_vanguard.sh> /usr/local/bin chmod 755
   /usr/local/bin/connect_vanguard.sh
   ```

9. Have it run every hour by linking it into cron.daily:

   ```
   ln - s /usr/local/bin/connect_vanguard.sh

   /etc/cron.daily/connect_vanguard
   ```

4.4.6 Configurations and Scripts

/etc/ipsec.conf

```
# /etc/ipsec.conf - Openswan IPsec configuration file

# This file: /usr/share/doc/openswan/ipsec.conf-sample
#
# Manual:     ipsec.conf.5

# conforms to second version of ipsec.conf specification

# basic configuration config setup
# Do not set debug options to debug configuration issues!
# plutodebug / klipsdebug = "all", "none" or a combation from below:
# "raw crypt parsing emitting control klips pfkey natt x509 dpd private"
# eg:
# plutodebug="control parsing"
# Again: only enable plutodebug or klipsdebug when asked by a developer
#
# enable to get logs per-peer
# plutoopts="--perpeerlog"
#
# Enable core dumps (might require system changes, like ulimit -C)
# This is required for abrtd to work properly
# Note: incorrect SElinux policies might prevent pluto writing the core
dumpdir=/var/run/pluto/
#
# NAT-TRAVERSAL support, see README.NAT-Traversal nat_traversal=yes
# exclude networks used on server side by adding %v4:!a.b.c.0/24
# It seems that T-Mobile in the US and Rogers/Fido in Canada are
```

```
# using 25/8 as "private" address space on their 3G network.

# This range has not been announced via BGP (at least upto 2010-12-21)


virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4:172.16.0.0/12,%v
4:25.0.0.0/8,%v6:fd00::/8,%v6:fe80::/10

# OE is now off by default. Uncomment and change to on, to enable. oe=off

# which IPsec stack to use. auto will try netkey, then klips then mast

#protostack=auto protostack=netkey

# Use this to log to a file, or disable logging on embedded systems (like openwrt)

#plutostderrlog=/dev/null

#plutodebug=all plutostderrlog=/var/log/pluto.log nat_traversal=yes

oe=off

#myid=172.16.0.66

# Add connections here conn nccoe-vanguard

type=tunnel forceencaps=yes authby=secret

ike=3des-sha1;modp1024 #don't actually need to specify this keyexchange=ike

ikelifetime=22800s phase2=esp

phase2alg=aes256-sha1;modp1024 salifetime=3600s

pfs=yes #vanguard has pfs on auto=start

keyingtries=3

#rekey=no


left=%defaultroute leftnexthop=%defaultroute

leftsubnet=172.16.0.0/24 #NCCoE ITAM lab internal subnet


# either one of these seems to work

#leftid=10.33.5.16  #behind firewall ip address leftid=136.160.255.42 #public ip
address



#leftsourceip=136.160.255.42 leftsourceip=10.33.5.16
```

```
right=174.47.13.99  #IOS outside address

rightid=174.47.13.99       #IKE ID send by IOS

#rightsubnet is the internal subnet on the distant end rightsubnet=172.17.212.0/24
#network behind IOS rightnexthop=%defaultroute
```

/etc/ipsec.secrets

```
# This file holds shared secrets or RSA private keys for inter-Pluto

# authentication. See ipsec_pluto(8) manpage, and HTML documentation.


# RSA private key for this host, authenticating it to any other host

# which knows the public part. Suitable public keys, for ipsec.conf, DNS,

# or configuration of other implementations, can be extracted conveniently

# with "ipsec showhostkey".


# this file is managed with debconf and will contain the automatically created RSA
keys

# The %any %any line is just for testing

# Replace MYSECRET with your pre-shared key


include /var/lib/openswan/ipsec.secrets.inc 172.16.0.67 174.47.13.99 : PSK "MYSECRET"

10.33.5.16 174.47.13.99 : PSK "MYSECRET"

#%any %any : PSK "MYSECRET"
```

/usr/local/bin/connect_vanguard.sh

```
#!/bin/sh


#start IPsec tunnel

ipsec auto --up nccoe-vanguard
```

```
#status

#ipsec auto --verbose --status
```

4.5 Ubuntu Apt-Cacher

Ubuntu Apt-Cacher is a central repository for update and patch management used by all Ubuntu systems on the network.

4.5.1 How It's Used

In the FS ITAM build, Ubuntu Apt-Cacher provides all Ubuntu systems with patches and updates.

4.5.2 Virtual Machine Configuration

The Ubuntu Apt-Cacher virtual machine is configured with one network interface cards, 4 GB of RAM and one CPU core.

4.5.3 Network Configuration

The management network interface card is configured as follows:

- IPv4 Manual
- IPv6 Ignore/Disabled
- IP Address: 172.16.0.67
- Netmask: 255.255.255.0
- Gateway: 172.16.0.11
- DNS Servers: 172.16.1.20, 172.16.1.21
- Search Domains: lab5.nccoe.gov

4.5.4 Installing Ubuntu Apt-Cacher

Ubuntu Apt-Cacher is installed on a hardened Ubuntu 14.04 Linux system. Complete installation instructions can be found at: https://help.ubuntu.com/community/Apt-Cacher-Server.

1. For Debian/Ubuntu Linux systems: It is always best to make sure your system is up-to-date by performing:

   ```
   sudo apt-get update sudo apt-get upgrade

   sudo apt-get install apt-cacher apache2
   ```

2. Enable apt-cacher by editing */etc/default/apt-cacher* and change **autostart** to **1**.

3. Restart Apache:

```
sudo /etc/init.d/apache2 restart
```

4. Verify that things are working by pointing your Web browser to http://<apt-cacher>:3142.
5. Edit */etc/apt-cacher/apt-cacher.conf* and uncomment the following line: `allowed_hosts = *`
6. Configure as a proxy to APT:

```
sudo nano /etc/apt/apt.conf.d/01proxy
```

7. Inside your new file, add a line that says:

```
Acquire::http::Proxy "http://<IP address or hostname of the apt-cacher
server>:3142";
```

8. Restart apt-cacher:

```
sudo /etc/init.d/apt-cacher restart
```

4.5.5 Client Configuration

1. Client configuration is the same as setting up the server as a proxy to APT:

```
sudo nano /etc/apt/apt.conf.d/01proxy
```

2. Inside your new file, add a line that says:

```
Acquire::http::Proxy "http://172.16.0.77:3142";
```

4.6 Windows 2012 Certificate Authority

The Windows 2012 Certificate Authority server in the ITAM build uses an NCCoE base 2012 R2 x86_64 DoD STIG image. The installation of the Windows 2012 Certificate Authority server was performed using installation media provided by DISA. This image was chosen because it is standardized, hardened, and fully documented.

4.6.1 Software Configurations

Windows 2012 Certificate Authority (CA) server was designed to issue certificates to endpoints that need to be accessed by users such that communication to such devices are deemed secure. It is used in building a PKI system.

4.6.2 How It's Used

The ITAM solution uses the Windows 2012 CA server to issue certificates to endpoints that have services that need to be accessed securely such as HTTPS enabled devices. The pfSense routers utilized these

certificates allowing for secure communication and configuration. The certificates are also utilized by Splunk Enterprise and the Splunk Universal Forwarder.

4.6.2.1 INSTALL ACTIVE DIRECTORY CERTIFICATE SERVICES (AD CS)

1. Go to **Server Manager** and click **Add Roles and Features Wizard**.
2. Click **Next**. Select **Role-based or feature-based installation**. Click **Next**.
3. Select your server on the next screen and click **Next**.
4. Select the **Active Directory Certificate Services** and **Add Features** when prompted.
5. Click **Next** when you see .NET 4.5 framework and other default selections.
6. Click **Next** on informational screens.
7. On the **Role Services for AD CS**, select all checkboxes and click **Next**.
8. When you are prompted to install the IIS web service, click **Install**.
9. Click **Close** when the installation completes.

4.6.2.2 CONFIGURE AD CS SERVICES PART 1

1. Go back to **Server Manager** and click on the warning icon.
2. Click on **Configure Active Directory Certificate Services**. Click **Next**.
3. On the Role Services to configure screen, select Certification Authority, Certification Authority Web Enrollment.
4. Choose **Enterprise CA**. On the following screen click **Next**.
5. Choose **Root CA** and click **Next**.
6. Choose **Create a new private key** and click **Next**.
7. Leave the defaults on the **Specify the cryptographic options** screen and click **Next**.
8. Specify the CA common name and click **Next**.
9. Use the default selection: **Specify a validity period at the default of 5 years for the certificates generated by this CA**.
10. Leave the database locations at default and click **Next**.
11. Click **Configure** to initiate configuration of the selected roles.

12. Click **Close** when the configurations succeed.
13. Click **No** if a **Configure additional role services** pop up is presented.

4.6.2.3 CONFIGURE AD CS PART 2

1. Go back to **Server Manager** and click on the yellow warning sign.
2. Click on **Configure AD CS on the destination server**.
3. Specify a user with credentials to configure role services. The user must be part of the **Enterprise Admins** group.
4. Select the other checkboxes and click **Next**.
5. Select a domain account with the specified permissions.
6. Accept the default **RA** name and click **Next**.
7. Accept the default Cryptographic options cryptographic service providers and key lengths and click **Next**.
8. Select the default CA name as the name to be used for **Certificate Enrollment Services**.
9. Specify the same service account for to be used for Certificate Enrollment Web Service.
10. Choose the available Server Certificate and click **Next**. Click **Configure**; then, click **Close**.

4.6.2.4 CONFIGURE A CERTIFICATE AND PUBLISH TO ACTIVE DIRECTORY

1. Open the Certification Authority tool from **Server Manager**.
2. Right-click **Certificate Templates**.
3. Click **Manage**.
4. Right-click Any template and click **Duplicate**.
5. Give it a distinct name/Template Display name.
6. Click the **Subject Name** tab and select **Common Name** from the subject name format dropdown list.
7. Click **Apply**, click **OK** and then close the dialog box.
8. Go back to the Certification Authority tool and right-click **Certificate Templates**.
9. Select the certificate you just created and click on **Properties**.

10. On the **General** tab, click on **Publish to Active Directory**.
11. Click on the **Security** tab, select **Domain Computers** and check the **Read**, **Enroll** and **Autoenroll** boxes.
12. Click **Apply** and then **OK** to close the dialog box.

4.6.2.5 CONFIGURE GROUP POLICY TO AUTO-ENROLL DOMAIN COMPUTERS

1. Log on to the domain controller.
2. Go to Group Policy Management Tool via Server Manager.
3. Expand the forest, then expand the domain.
4. Right-click on **Default Domain Policy** and click **Edit**.
5. Click Computer Configuration, Policies, Windows Settings, Security Settings, Public Key Policies and open Certificates Services Client Auto-Enrollment policy.
6. Choose **Enabled** from the Configuration Model box, check Renew Expired certificates, update pending certificates, and remove revoked certificates.
7. Also check Update certificates that use certificate templates.
8. Click **Apply**; then, click **OK**.
9. Click Computer Configuration, Policies, Windows Settings, Security Settings, and Public Key Policies.
10. Right-click Certificate Services Client - Certificate Enrollment Policy, click **Properties**.
11. Choose **Enabled** from the **Configuration Model** drop down list.
12. Ensure that **Active Directory Enrollment Policy** is checked.
13. Check Properties of Active Directory Enrollment Policy and ensure that the **Enable for automatic enrollment and renewal** and the **Require strong validation during enrollment** boxes are checked.
14. Click **Apply** and then **OK** to close the dialog boxes.

4.6.3 Certificate Generation and Issuance

This ITAM solution had a mix of endpoints which included Windows and Linux hosts including some pfSense routers. Some of these devices pfSense routers had HTTPS enabled. The PKI implementation was extended to further secure these HTTPS services. The overall process includes the following steps:

1. Generate a certificate signing request (CSR).
2. Copy the CSR over to the Windows Certificate Authority (CA).
3. Submit the CSR to the CA service.
4. Sign the CSR and copying the issued certificate along with the CA certificate to the device.
5. Generate a Certificate Signing Request.
6. Open the terminal in a Linux computer with OpenSSL and run `openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr`

   where `server.key` and `server.csr` represent arbitrary names you have chosen. The common name field should be the FQDN of the endpoint.

   This will generate two files: the private key file and a CSR file.
7. Copy the CSR file.
   a. Use any of the file transfer utilities such as SCP or FTP to copy the CSR to the CA.
   b. Alternatively, the CSR can be copied via USB or other means.
8. Submit the Certificate Signing Request to the CA Service.
   a. Log on to the CA server, go to the command prompt and type `Certreq.exe -attrib "CertificateTemplate:<Nameofthetemplate>" -submit <pathtoCSR>`
   b. An example of what could be typed is `certreq.exe -attrib "CertificateTemplate:WebServer" -submit D:\requestfile.txt`
9. Sign the CSR and copy the Certificates to the device.
   a. To sign the CSR, go to the Windows CA server and perform the following steps:
      i. Click **Start** > **Control Panel** > **Administrative Tools** > **Certification Authority**.
      ii. Expand the **CA name** and click **Pending Requests**.
      iii. Right-click the CSR on the right pane showing a request **ID number** > **Click All Tasks** > **Click Issue**.

b. Run `certutil -ca.cert ca_name.cer` from the command prompt

   where `ca_name.cer` is the arbitrary file name for the CA certificate.

10. Copy the client certificate and CA certificate to client system.
11. Make the application aware of the location of these certificates. Once logged in, the pfSense routers in the ITAM build provide links to copy and paste the contents of the private key, the certificate file and the CA server certificate.

4.7 Common PKI Activities

This section provides instructions for common PKI activities using a Microsoft Certificate Authority (CA) in a heterogeneous environment.

4.7.1 Generating a Certificate Signing Request from OpenSSL

1. Run:

   `openssl req -new -newkey rsa:2048 -nodes -keyout serverFQDN.key -out serverFQDN.csr`

   where **`serverFQDN.key`** is the private key file and the **`serverFQDN.csr`** is the certificate signing request file. The files can be arbitrarily named.

2. When prompted, ensure that the common name field is set to the server FQDN.

   A Certificate Signing Request (CSR) can be generated for as many servers as you need in your enterprise.

3. Copy the CSR file to the Certificate Authority (CA) server for signing.

4.7.2 Submitting the CSR to the CA Service

1. Log on to the CA server.
2. Go to the command prompt and type:

   `Certreq.exe -attrib "CertificateTemplate:<Nameofthetemplate>" -submit`

   `<pathtoCSR>`

   An example command could be:

   `certreq.exe -attrib "CertificateTemplate:WebServer" -submit D:\serverFQDN.key`

4.7.3 Exporting a Root Certificate from a Microsoft CA

1. From the command prompt run:

   ```
   Certutil -ca.cert new_ca_filename.cer
   ```

   where `new_ca_filename.cer` is the arbitrary file name for the exported CA certificate.

The exported CA certificate would need to be copied over to the other servers that would be included in Public Key Infrastructure.

The Microsoft Windows CA root certificate would be in Distinguished Encoding Rules (DER) encoded format. Some platforms, especially Linux platforms, may prefer PEM encoding and conversion to Privacy Enhanced Mail (PEM) encoding might be necessary.

4.7.4 Converting from DER Encoding to PEM Encoding

1. Run:

   ```
   openssl x509 -in DER_CA_CERT.crt -inform der -outformpem -out PEM_CA_CERT.pem
   ```

   where `DER_CA_CERT.crt` is DER encoded and `PEM_CA_CERT` is the transformed PEM encoded certificate.

Additional information on converting certificates can be found at the following link http://info.ssl.com/article.aspx?id=12149.

4.8 Process Improvement Achievers (PIA) Security Evaluation

Process Improvement Achievers (PIA) conducted a remote security evaluation of the FS ITAM build. The evaluation consisted of running multiple tools against the machines in the lab to find any vulnerabilities due to misconfiguration.

Appendix A List of Acronyms

AD	Active Directory
CA	CA Technologies
CA	Certificate Authority
COTS	Commercial Off-The-Shelf
CRADA	Collaborative Research and Development Agreement
CSR	Certificate Signing Request
.csv	Comma-Separated Value
DER	Distinguished Encoding Rules
DMZ	Demilitarized Zone
FS	Financial Sector
HR	Human Resources
ID	Identity
ITAM	Information Technology Asset Management
IDS	Intrusion Detection System
IP	Internet Protocol
NAS	Network Attached Storage
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
OS	Operating System
PEM	Privacy Enhanced Mail
PKI	Public Key Infrastructure
SME	Subject Matter Expert
SQL	Structured Query Language
SSL	Secure Socket Layer
STIG	Security Technical Implementation Guideline

TLS	Transport Layer Security
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network