

IT ASSET MANAGEMENT

How-To Guides

For Security Engineers

Michael Stone

Chinedum Irrechukwu

Harry Perper

Devin Wynne

Leah Kauffman, Editor-in-Chief

NIST SPECIAL PUBLICATION 1800-5c

DRAFT

IT ASSET MANAGEMENT

Financial Services

DRAFT

Michael Stone

National Cybersecurity Center of Excellence
Information Technology Laboratory

Chinedum Irrechukwu

Harry Perper

Devin Wynne

The MITRE Corporation
McLean, VA

Leah Kauffman, Editor-in-Chief

National Cybersecurity Center of Excellence
Information Technology Laboratory

October 2015

U.S. Department of Commerce

Penny Pritzker, Secretary



National Institute of Standards and Technology

Willie May, Under Secretary of Commerce for Standards and Technology and Director

DISCLAIMER

Certain commercial entities, equipment, 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, materials, or equipment 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, 157 pages (October 2015)
CODEN: NSPUE2

Organizations are encouraged to review all draft publications during public comment periods and provide feedback. All publications from NIST's National Cybersecurity Center of Excellence are available at <http://nccoe.nist.gov>.

Comments on this publication may be submitted to: financial_nccoe@nist.gov

Public comment period: October 26, 2015 through January 8, 2016

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
9600 Gudelsky Drive (Mail Stop 2002) Rockville, MD 20850
Email: financial_nccoe@nist.gov

DRAFT

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE) at the National Institute of Standards and Technology (NIST) addresses businesses' most pressing cybersecurity problems with practical, standards-based solutions using commercially available technologies. The NCCoE collaborates with industry, academic, and government experts to build modular, open, end-to-end reference designs that are broadly applicable and repeatable. The center's work results in publically available NIST Cybersecurity Practice Guides, Special Publication Series 1800, that provide users with the materials lists, configuration files, and other information they need to adopt a similar approach.

To learn more about the NCCoE, visit <http://nccoe.nist.gov>. To learn more about NIST, visit <http://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.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. The documents in this series 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.

This NIST Cybersecurity Practice Guide provides a reference build of an ITAM solution. The build contains descriptions of the architecture, all products used in the build and their individual configurations. Additionally, this guide provides a mapping of each product to multiple relevant security standards. While the reference solution was demonstrated with a certain suite of products, the guide does not endorse these products in particular. Instead, it presents the characteristics and capabilities that an organization's security experts can use to identify similar standards-based products that can be integrated quickly and cost-effectively with a financial service company's existing tools and infrastructure.

KEYWORDS

access control; access management; attribute provider; authentication; authorization; identity federation; identity management; Identity Provider; relying party

ACKNOWLEDGMENTS

We gratefully acknowledge the contributions of the following individuals and organizations for their generous contributions of expertise, time, and products.

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
Susan Symington	MITRE Corporation
Sallie Edwards	MITRE Corporation
Sarah Weeks	MITRE Corporation
Lina Scorza	MITRE Corporation
Karen Scarfone	Scarfone Cybersecurity

Contents

2	1	Introduction.....	1
3	1.1	Practice Guides.....	2
4	1.2	Typographical Conventions	3
5	1.3	Build Overview	3
6	1.4	Build Architecture Components Overview	5
7	1.5	Build Network Components	6
8	1.6	Operating Systems	7
9	1.7	Base Windows Installation and Hardening Details	7
10	1.8	Base Linux Installation and Hardening Details	8
11	2	Tier 1	9
12	2.1	Software Configurations.....	10
13	2.1.1	Splunk Enterprise	10
14	2.1.2	How It's Used.....	10
15	2.1.3	Installing Splunk Enterprise	11
16	2.1.3.1	Disable Transparent Huge Pages	12
17	2.1.4	Configurations.....	12
18	2.1.4.1	Splunk Enterprise Data Inputs.....	12
19	2.1.4.2	Splunk Enterprise Indexes	13
20	2.1.4.3	Splunk Enterprise Apps	14
21	2.1.4.4	Splunk Enterprise Connections	15
22	2.1.5	Lookup Table Files	26
23	2.1.5.1	Splunk Enterprise Configuration Files	27
24	2.1.5.2	Splunk Enterprise Dashboards.....	27
25	3	Tier 2	29
26	3.1	AssetCentral	30
27	3.1.1	How It's Used.....	30
28	3.1.2	Virtual Machine Configuration.....	30
29	3.1.3	Network Configuration	30
30	3.1.4	Installing AssetCentral	30
31	3.1.5	Installing MySQL (MariaDB)	31
32	3.1.6	Installing Apache	31
33	3.1.7	Installing PHP5	31
34	3.1.8	Post Installation Tasks.....	32
35	3.1.9	Database Update – Add a View.....	32
36	3.1.10	Add Assets into AssetCentral	33
37	3.2	BelManage.....	34
38	3.2.1	How It's Used.....	34
39	3.2.2	Virtual Machine Configuration.....	34
40	3.2.3	Network Configuration	34

41	3.2.4	Installing BelManage	34
42	3.2.4.1	Prerequisites	35
43	3.2.4.2	Installation Procedure.....	35
44	3.2.5	Integration and Final Steps.....	36
45	3.3	Bro	37
46	3.3.1	How It's Used.....	37
47	3.3.2	Virtual Machine Configuration.....	37
48	3.3.3	Network Configuration	38
49	3.3.4	Installing Bro	38
50	3.3.4.1	Installation Prerequisites	38
51	3.3.4.2	Installation Procedure.....	39
52	3.3.5	Installing Intelligence Gathering Software	40
53	3.3.6	Configuring Bro.....	40
54	3.3.7	Installing Splunk Universal Forwarder	41
55	3.3.8	Configuring Splunk Universal Forwarder	42
56	3.3.9	Configurations and Scripts.....	43
57	3.4	CA Technologies IT Asset Manager	50
58	3.4.1	How It's Used.....	50
59	3.4.2	Virtual Machine Configuration.....	50
60	3.4.3	Network Configuration	51
61	3.4.4	Installing CA ITAM	51
62	3.4.5	Configurations.....	52
63	3.4.5.1	Data Import.....	52
64	3.5	Fathom Sensor from RedJack	54
65	3.5.1	How It's Used.....	54
66	3.5.2	Virtual Machine Configuration.....	54
67	3.5.3	Network Configuration	55
68	3.5.4	Installing Fathom Sensor	55
69	3.5.5	Installing Splunk Universal Forwarder	60
70	3.5.6	Configuring Splunk Universal Forwarder	60
71	3.5.7	Helpful Commands and Information	61
72	3.5.8	Configurations and Scripts.....	62
73	3.6	OpenVAS	63
74	3.6.1	How It's Used.....	63
75	3.6.2	Virtual Machine Configuration.....	63
76	3.6.3	Network Configuration	63
77	3.6.4	Installation Prerequisites.....	63
78	3.6.5	Installing OpenVAS.....	64
79	3.6.6	Configuring OpenVAS	66
80	3.6.7	Installing Splunk Universal Forwarder	67
81	3.6.8	Configuring Splunk Universal Forwarder	68
82	3.6.9	Configurations and Scripts.....	68
83	3.7	Puppet Enterprise	72
84	3.7.1	How It's Used.....	72
85	3.7.2	Prerequisites.....	73

86	3.7.3	Installing Puppet Enterprise Server	73
87	3.7.4	Puppet Enterprise Linux Agent Installation	73
88	3.7.5	Puppet Enterprise Windows Agent Installation	74
89	3.7.6	Puppet Enterprise Agent Configuration	74
90	3.7.7	Puppet Enterprise Manifest Files and Modules	75
91	3.7.7.1	Module: windowsnodes	76
92	3.7.7.2	Module: ubuntubase	76
93	3.7.7.3	Module: redhatbase	76
94	3.7.7.4	Module: clamav	76
95	3.7.7.5	Module: blacklist	77
96	3.7.7.6	Software Blacklist Removal	77
97	3.7.8	Reporting	77
98	3.7.9	Report Directory Cleanup	77
99	3.7.10	Puppet Code and Scripts	78
100	3.8	Snort	89
101	3.8.1	How It's Used	90
102	3.8.2	Virtual Machine Configuration	90
103	3.8.3	Network Configuration	90
104	3.8.4	Installing Snort	90
105	3.8.5	Installing Snort	90
106	3.8.6	Get Updated Community Rules	91
107	3.8.7	Installing Barnyard2	91
108	3.8.8	Testing	92
109	3.8.9	Installing Splunk Universal Forwarder	93
110	3.8.10	Configuring Splunk Universal Forwarder	94
111	3.8.11	Configurations and Scripts	94
112	3.9	Tyco Security Products	125
113	3.9.1	Installing Tyco Security Products	125
114	3.9.2	Configurations	126
115	3.10	Windows Server Update Services (WSUS)	127
116	3.10.1	How It's Used	127
117	3.10.2	Virtual Machine Configuration	127
118	3.10.3	Network Configuration	128
119	3.10.4	Installing WSUS	128
120	3.10.5	Configurations	128
121	3.10.6	Configure Active Directory Server to Require WSUS	129
122	3.10.7	Create WSUS Statistics for Splunk Enterprise	129
123	3.10.8	Installing Splunk Universal Forwarder	131
124	3.10.9	Configuring Splunk Universal Forwarder	131
125	4	Tier 3	135
126	4.1	Active Directory Server	136
127	4.1.1	Software Configurations	136
128	4.1.1.1	Windows 2012 Active Directory Server	136
129	4.1.2	How It's Used	136

130	4.1.3	Installation.....	136
131	4.2	Asset Central	139
132	4.2.1	How It's Used.....	139
133	4.2.2	Virtual Machine Configuration.....	139
134	4.2.3	Network Configuration	139
135	4.2.4	Installing AssetCentral	139
136	4.2.5	Installing MySQL (MariaDB)	139
137	4.2.6	Installing Apache	140
138	4.2.7	Installing PHP5	140
139	4.2.8	Post Installation Tasks	140
140	4.3	Email	141
141	4.3.1	How It's Used.....	141
142	4.3.2	Virtual Machine Configuration.....	141
143	4.3.3	Network Configuration	141
144	4.3.4	Installing Email.....	142
145	4.3.5	Configure Email	142
146	4.3.6	User Accounts	142
147	4.3.7	DNS Settings	143
148	4.3.8	Configuration Files.....	143
149	4.4	Openswan (VPN)	144
150	4.4.1	How It's Used.....	145
151	4.4.2	Virtual Machine Configuration.....	145
152	4.4.3	Network Configuration	145
153	4.4.4	Installing Openswan	145
154	4.4.5	Installing Openswan	145
155	4.4.6	Configurations and Scripts.....	146
156	4.5	Ubuntu Apt-Cacher	148
157	4.5.1	How It's Used.....	149
158	4.5.2	Virtual Machine Configuration.....	149
159	4.5.3	Network Configuration	149
160	4.5.4	Installing Ubuntu Apt-Cacher	149
161	4.5.5	Client Configuration	150
162	4.6	Windows 2012 Certificate Authority	150
163	4.6.1	Software Configurations	150
164	4.6.2	How It's Used.....	150
165	4.6.3	Certificate Generation and Issuance	152
166	4.7	Common PKI Activities	153
167	4.7.1	Generating a Certificate Signing Request from OpenSSL.....	154
168	4.7.2	Submitting the CSR to the CA Service	154
169	4.7.3	Exporting a Root Certificate from a Microsoft CA	154
170	4.7.4	Converting from DER Encoding to PEM Encoding.....	154
171	4.8	Process Improvement Achievers (PIA) Security Evaluation	155
172	Appendix A	Acronyms	157

1 Introduction

2	1.1	Practice Guides.....	2
3	1.2	Typographical Conventions.....	3
4	1.3	Build Overview	3
5	1.4	Build Architecture Components Overview.....	5
6	1.5	Build Network Components.....	6
7	1.6	Operating Systems.....	7
8	1.7	Base Windows Installation and Hardening Details.....	7
9	1.8	Base Linux Installation and Hardening Details.....	8

10

1.1 Practice Guides

The following guides show IT professionals and security engineers how we implemented this example solution to address the challenges associated with providing a secure, centralized, uniform, and efficient solution for managing information technology (IT) hardware assets, software assets, and analysis across multiple integrated financial sector networks. All products that we employed in this solution are included in this guide. We have not recreated the product manufacturer's documentation, which is presumed to be widely available. Rather, these guides describe how we incorporated the products together in our environment.

These guides assume that you have experience implementing security products in the financial sector. While we have used the commercially-available products described here, we assume that you have the knowledge and expertise to choose other products that might better fit your existing infrastructure and business processes.¹ If you use substitute products, we hope that you will seek products that are congruent with standards and best practices in the financial services, as we have.

This NIST Cybersecurity Practice Guide does not describe “the” solution, but a possible solution. This is a draft version. We are seeking feedback on its contents and welcome your input. Comments and suggestions will improve subsequent versions of this guide. Please contribute your thoughts to financial_nccoe@nist.gov, and join the discussion at <http://nccoe.nist.gov/forums/financial-services>.

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.Certain commercial entities, equipment, 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 the National Cybersecurity Center of Excellence (NCCoE), nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

1.2 Typographical Conventions

The following table presents typographic conventions used in this volume.

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

1.3 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 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

Table 1.1 Build Architecture Component List

Host	Product	Function	Internet Protocol Address	Operating System
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	172.16.0.20	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
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				

Table 1.1 Build Architecture Component List

Host	Product	Function	Internet Protocol Address	Operating System
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 management	172.16.3.92	Windows Server 2012R2
Physical Security				
Router4	pfSense	Router/firewall	172.16.0.15 172.16.4.11	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

1.4 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.

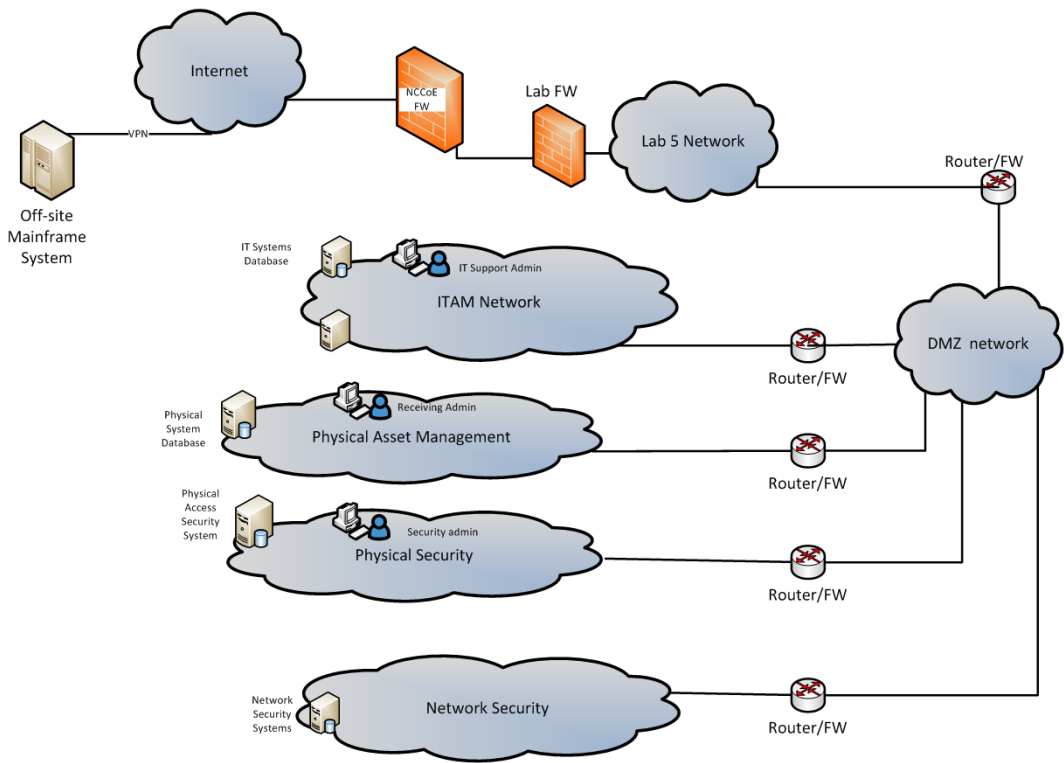


Figure 1.1 ITAM Build

1.5 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 sever 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 particular 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.

1.6 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.

1.7 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.

1.8 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.

2 Tier 1

2	2.1 Software Configurations.....	10
---	----------------------------------	----

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

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

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/latest/Installation/InstallonLinux#RedHat_RPM_install

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
```

Splunk Enterprise also requires several ports to be opened through the firewall(s). To allow these ports through the built-in firewalld 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
```

It is also recommended to increase the amount 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.

Splunk Enterprise can now be accessed by opening up 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/6.3.0/ReleaseNotes/SplunkandTHP>.

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_hugepage/defrag; then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

Ensure that rc.local is executable.

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

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

Syslog TCP

Settings -> Data Inputs -> TCP

TCP port	Host Restriction	Source type	Status	Actions
514		syslog	Enabled / Disable	Clone / Delete

Figure 2.1 Splunk Enterprise Syslog TCP Input

Syslog UDP

Settings -> Data Inputs -> UDP



Figure 2.2 Splunk Enterprise Syslog UDP Input

Receive Data from Splunk Universal Forwarders

Settings -> Forwarding and Receiving -> Configure Receiving

Click the **New** button and enter port **9997**.

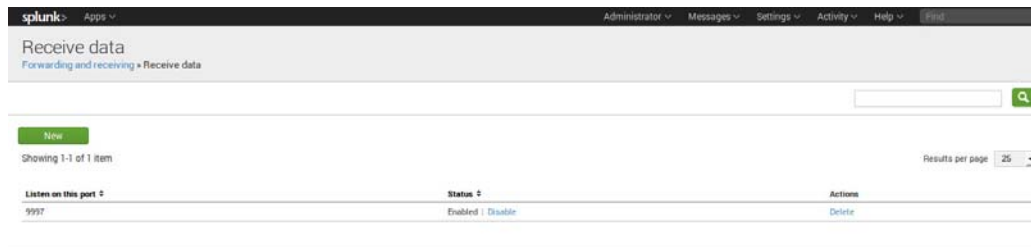


Figure 2.3 Splunk Enterprise Receive from Splunk Universal Forwarder

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 for the list of names.)
4. Ensure that the **Home Path** is set to `/data/splunk`.

Follow these steps for each index that you need to create. For additional information on indexes, go to:

<http://docs.splunk.com/Documentation/Splunk/6.2.0/Indexer/Setupmultipleindexes>.

102

Table 2.2 Splunk Enterprise Indexes

Index Name
alerts
assetcentral
belmanage_computers
belmanage_hotfixesmissing
belmanage_hw_changes
belmanage_sw_changes
belmanage_software
bro
ca_itam
fathom
firewall
mainframe
openvas
puppet
router_configs
snort
syslog
tyco
wsus

103 2.1.4.3 Splunk Enterprise Apps

104 Several Splunk Enterprise Apps were used in this project. The list of Splunk Enterprise Apps
105 needed for the ITAM project can be found in Table 2.3. Splunk Enterprise Apps assist in
106 processing, analyzing and displaying different types of data. To download Splunk Enterprise
107 Apps you must have a valid Splunk account. You can install Splunk Enterprise Apps from
108 <https://splunkbase.splunk.com/>.

109 To installing Splunk Enterprise Apps follow these steps:

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

114

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 v1	Allows database queries to be run as Splunk Enterprise queries.
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.

115 The Splunk DB Connect v1 and Splunk DB Connect v2 apps require the downloading and
 116 installation of specific database drivers. Database-specific drivers should be placed in the
 117 directory `$SPLUNK_HOME/etc/apps/splunk_app_db_connect/bin/lib`. This project required the
 118 installation of database drivers for Microsoft SQL and MySQL. The drivers must be obtained
 119 from the database manufacturers; in this case Microsoft and MySQL/Oracle. For more detailed
 120 information, please refer to **Install database drivers** at
 121 <http://docs.splunk.com/Documentation/DBX/latest/DeployDBX/Installdatabasedrivers>. The
 122 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

124 2.1.4.4 Splunk Enterprise Connections

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

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

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

133 The following tables provide the information needed to perform these steps.

134

Table 2.5 DB Connect v2 Identities

Identity	Used with
asset_query	AssetCentral
mike	BelManage
splunk	CA ITAM

135 2.1.4.4.1 Splunk Enterprise DB Connect v2 Connections

136 There should only be one database connection to each individual database. The database
137 connections use the identities listed in [Table 2.5](#). Please remember to select the **Enable** button
138 when you configure each connection.

139 DB Connect V2 AssetCentral Connection

- 140 ■ AssetCentral
- 141 ■ Status: Enabled
- 142 ■ Connection Name: assetcentral
- 143 ■ App: Splunk DB Connect v2
- 144 ■ Host: assetcentral
- 145 ■ Database Types: MySQL
- 146 ■ Default Database: assetcentral
- 147 ■ Identity: asset_query
- 148 ■ Port: 3306
- 149 ■ Enable SSL: NOT CHECKED
- 150 ■ Readonly: NOT CHECKED

151 DB Connect V2 BelManage Connection

- 152 ■ BelManage
- 153 ■ Status: Enabled
- 154 ■ Connection Name: BelManage
- 155 ■ App: Splunk DB Connect v2
- 156 ■ Host: belmanage
- 157 ■ Database Types: MS-SQL Server Using MS Generic Driver
- 158 ■ Default Database: BelMonitor82_1
- 159 ■ Identity: mike
- 160 ■ Port: 1433
- 161 ■ Enable SSL: NOT CHECKED
- 162 ■ 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 sections show the configurations for each operation.

AssetCentral

DB Input: assetcentral

Name Input 1 of 4

Status: Enabled

Name: assetcentral

Description: Assets from AssetCentral

App: Splunk DB Connect v2

Connection: assetcentral

Click the **Continue** button.

Choose and Preview Table 2 of 4

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

197 Catalog: assetcentral
198 Schema: NULL
199 Table: assetview
200 Max rows: 100
201 Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.
202 Click the **Continue** button.
203
204 Set Parameters 3 of 4
205 Type: Batch Input
206 Max Rows to Retrieve: 100000
207 Timestamp: Current Index Time
208 Output Timestamp Format: YYYY-MM-dd HH:mm:ss
209 Execution Frequency: 0 0 * * *
210 Click the **Continue** button.
211
212 Metadata 4 of 4
213 Source: assetcentral
214 Sourcetype: assetcentral
215 Index: assetcentral
216 Select Resource Pool: local
217 Click the **Save** button.
218
219 **BelManage_Computers**
220 DB Input: BelManage_Computers
221 Name Input 1 of 4
222 Status: Enabled
223 Name: BelManage_Computers
224 Description: Computer info from BelManage
225 App: Splunk DB Connect v2
226 Connection: BelManage
227 Click the **Continue** button.
228
229 Choose and Preview Table 2 of 4
230 Make sure that **Simple Query Mode** is selected.

231 Catalog: BelMonitor82_1
232 Schema: dbo
233 Table: Computers
234 Max rows: 100
235 Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.
236 Click the **Continue** button.
237
238 Set Parameters 3 of 4
239 Type: Rising Column
240 Max Rows to Retrieve: 100000
241 Specify Rising Column: ProfileDate
242 Timestamp: Current Index Time
243 Output Timestamp Format: YYYY-MM-dd HH:mm:ss
244 Execution Frequency: * * * * *
245 Click the **Continue** button.
246
247 Metadata 4 of 4
248 Source: belmanage
249 Souretype: belmanage_computers
250 Index: belmanage_computers
251 Select Resource Pool: local
252 Click the **Save** button.
253
254 **Belmanage_hotfixesmissing**
255 DB Input: belmanage_hotfixesmissing
256 Name Input 1 of 4
257 Status: Enabled
258 Name: belmanage_hotfixesmissing
259 Description: List of hotfixes/patches missing from each computer.
260 App: Splunk DB Connect v2
261 Connection: BelManage
262 Click the **Continue** button.
263
264 Choose and Preview Table 2 of 4

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

266 In the entry box type in the following SQL statement:

267 `SELECT HotfixesMissing.*, Computers.ProfileName, Computers.NetworkIPAddress`
268 `FROM HotfixesMissing INNER JOIN Computers on HotfixesMissing.Id = Computers.Id`

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

270 Click the **Continue** button.

271

272 Set Parameters 3 of 4

273 Type: Batch Input

274 Max Rows to Retrieve: 100000

275 Timestamp: Current Index Time

276 Output Timestamp Format: YYYY-MM-dd HH:mm:ss

277 Execution Frequency: 30 4 * * *

278 Click the **Continue** button.

279

280 Metadata 4 of 4

281 Source: belmanage

282 Sourcetype: belmanage_hotfixesmissing

283 Index: belmanage_hotfixesmissing

284 Select Resource Pool: local

285 Click the **Save** button.

286 **Belmanage_hw_changes**

287 DB Input: belmanage_hw_changes 1 of 4

288 Status: Enabled

289 Name: belmanage_hw_changes

290 Description: BelManage hardware changes

291 App: Splunk DB Connect v2

292 Connection: BelManage

293 Click the **Continue** button.

294

295 Choose and Preview Table 2 of 4

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

297 Catalog: BelMonitor82_1

298 Schema: dbo

299 Table: HistoryReportAllHardware
300 Max rows: 100
301 Click the **Magnifying Glass** button and up to 100 rows should be returned and displayed.
302 Click the **Continue** button.
303
304 Set Parameters 3 of 4
305 Type: Rising Column
306 Max Rows to Retrieve: 10000
307 Specify Rising Column: ActionDate
308 Timestamp: Current Index Time
309 Output Timestamp Format: YYYY-MM-dd HH:mm:ss
310 Execution Frequency: */15 * * * *
311 Click the **Continue** button.
312
313 Metadata 4 of 4
314 Source: belmanage
315 Sourcetype: belmanage_hw_changes
316 Index: belmanage_hw_changes
317 Select Resource Pool: local
318 Click the **Save** button.
319
320 **Belmanage_software**
321 DB Input: belmanage_software
322 Name Input 1 of 4
323 Status: Enabled
324 Name: belmanage_software
325 Description: Software from BelManage
326 App: Splunk DB Connect v2
327 Connection: BelManage
328 Click the **Continue** button.
329

Choose and Preview Table 2 of 4

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

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,
```

```

380         dbo.VersionFormat(dbo.VersionCompose (ProductVersionNoMS,
381 ProductVersionNoLS)) AS ProductVersionNo,
382         dbo.VersionFormat(dbo.VersionCompose (FileVersionNoMS, FileVer-sionNoLS)) AS
383 FileVersionNo,
384         CASE StartUp WHEN 1 THEN 'auto' ELSE 'user' END AS StartUp,
385         CASE InUse WHEN 1 THEN 'yes' WHEN 0 THEN 'no' ELSE NULL END AS InUse,
386         CASE ServiceStatus WHEN 1 THEN 'running' WHEN 0 THEN 'stopped' ELSE NULL END
387 AS ServiceStatus,
388         CASE ServiceStartType WHEN 2 THEN 'auto' WHEN 3 THEN 'manual' WHEN 4 THEN
389 'disabled' ELSE NULL END AS ServiceStartType,
390         LastUserDomain, LastUser, LastUserFullName,
391         CASE WHEN Is64Bit = 1 THEN 'yes' ELSE 'no' END AS Is64Bit,
392         CASE WHEN IsNativeToOs = 1 THEN 'yes' ELSE 'no' END AS IsNativeToOs,
393         MachineType,
394         ExeHeaderTypeLong AS ExeHeaderType,
395         LoginUser,
396         S.Language AS Language_soft, S.LanguageName AS LanguageName_soft
397 FROM
398     Software S INNER JOIN Computers C ON S.Id = C.Id;
399

```

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

Click the **Continue** button.

Set Parameters 3 of 4

Type: Rising Column

Max Rows to Retrieve: 10000

Specify Rising Column: ProfileDate_soft

Timestamp: Current Index Time

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

Execution Frequency: * * * *

Click the **Continue** button.

Metadata 4 of 4

Source: belmanage

Sourcetype: belmanage_software

Index: belmanage_software

Select Resource Pool: local

Click the **Save** button.

418 **Belmanage_sw_changes**

419 DB Input: belmanage_sw_changes

420 Name Input 1 of 4

421 Status: Enabled

422 Name: belmanage_sw_changes

423 Description: Software changes from BelManage

424 App: Splunk DB Connect v2

425 Connection: BelManage

426 Click the **Continue** button.

427

428 Choose and Preview Table 2 of 4

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

430 Catalog: BelMonitor82_1

431 Schema: dbo

432 Table: SoftwareHistoryReport

433 Max rows: 100

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

435 Click the **Continue** button.

436

437 Set Parameters 3 of 4

438 Type: Rising Column

439 Max Rows to Retrieve: 100000

440 Specify Rising Column: ActionDate

441 Timestamp: Current Index Time

442 Output Timestamp Format: YYYY-MM-dd HH:mm:ss

443 Execution Frequency: */30 * * * *

444 Click the **Continue** button.

445

446 Metadata 4 of 4

447 Source: belmanage

448 Sourcetype: belmanage_sw_changes

449 Index: belmanage_sw_changes

450 Select Resource Pool: local

451 Click the **Save** button.

CA ITAM

DB Input: ca-itam

Name Input 1 of 4

Status: Enabled

Name: ca-itam

Description: Asset from CA ITAM software

App: Splunk DB Connect v2

Connection: ca-itam

Click the **Continue** button.

Choose and Preview Table 2 of 4

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

In the entry box type in the following SQL statement:

```

SELECT DISTINCT
aud_ca_owned_resource.resource_name,audit_model_uuid,audit_resource_class,
audit_resource_subclass,
ca_owned_resource.own_resource_id,ca_owned_resource.mac_address,ca_owned_resource.ip_address,ca_owned_resource.host_name,ca_owned_resource.serial_number,ca_owned_resource.asset_source_uuid,ca_owned_resource.creation_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

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

Set Parameters 3 of 4

Type: Rising Column

Max Rows to Retrieve: 1000

Specify Rising Column: last_update_date

490 Timestamp: Current Index Time
491 Output Timestamp Format: YYYY-MM-dd HH:mm:ss
492 Execution Frequency: */5 * * * *
493 Click the **Continue** button.
494
495 Metadata 4 of 4
496 Source: ca-itam
497 Sourcetype: ca-itam
498 Index: ca_itam
499 *NOTE: the index name is **ca_itam** with an underscore. Splunk Enterprise does not accept
500 dashes in index names.
501 Select Resource Pool: local
502 Click the **Save** button.

503 2.1.5 Lookup Table Files

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

507 To create a lookup table file:

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

513 Create the following lookup table files:

514 /opt/splunk/etc/apps/search/lookups/AssetRisk_Alltime.csv
515 /opt/splunk/etc/apps/search/lookups/AssetRisk_Last7days.csv
516 /opt/splunk/etc/apps/search/lookups/AssetRisk_Last24hours.csv
517 /opt/splunk/etc/apps/search/lookups/asset_value_table.csv
518 /opt/splunk/etc/apps/search/lookups/license_table.csv
519 /opt/splunk/etc/apps/search/lookups/updown
520 /opt/splunk/etc/apps/search/lookups/vun_rating_table.csv

521 2.1.5.1 Splunk Enterprise Configuration Files

522 Splunk Enterprise configuration files can be found in the external file titled
523 [Splunk_Configuration_Files.tar.gz](#).

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

526 2.1.5.2 Splunk Enterprise Dashboards

527 Splunk Enterprise stores dashboards in XML format. All of the dashboards can be found in the
528 external file titled [Splunk_Dashboards.tar.gz](#).

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

531

3 Tier 2

2	3.1	AssetCentral.....	30
3	3.2	BelManage	34
4	3.3	Bro.....	37
5	3.4	CA Technologies IT Asset Manager	50
6	3.5	Fathom Sensor from RedJack	54
7	3.6	OpenVAS	63
8	3.7	Puppet Enterprise	72
9	3.8	Snort.....	89
10	3.9	Tyco Security Products	125
11	3.10	Windows Server Update Services (WSUS)	127
12			

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 Email 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

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

Recommended versions:

RedHat	Enterprise Linux Server	6.4 (Santiago) (x86_64)
Apache	Web Server	httpd-2.2.15-26.el6.x86_64
mysql	Server version:	5.1.66
php	version	5.3.3 or higher

38 3.1.5 Installing MySQL (MariaDB)

```
39 # yum -y install mariadb-server mariadb
40 #systemctl start mariadb.service
41 #systemctl enable mariadb.service
42 # mysql_secure_installation
43 Answer the questions with the default answers while performing the
44 mysql_secure_installation.
45 Create a database - assetcentral
46 Create a user - assetcentral
47 Grant all privileges to assetcentral user
```

48 3.1.6 Installing Apache

```
49 # yum -y install httpd
50 #systemctl start httpd.service
51 #systemctl enable httpd.service
52 #firewall-cmd --permanent --zone=public --add-service=http
53 #firewall-cmd --permanent --zone=public --add-service=https
54 #firewall-cmd -reload
```

55 HTTP Configuration

```
56 Go to HTTPD root; normally (/etc/httpd).
57 Under the modules directory make sure libphp5.so exists.
58 Change document root (webroot) as per environment in httpd.conf.
```

59 3.1.7 Installing PHP5

```
60 #yum -y install php
61 #systemctl restart httpd.service
62 #yum search php
63 #yum -y install php-mysql
64 #yum -y install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc
65 php-mbstring php-snmp php-soap curl curl-devel
66 Restart Apache
67 #systemctl restart httpd.service
```

68 3.1.8 Post Installation Tasks

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

72 Sample

```
73 <?php defined('ASSET_CENTRAL') or die('');  
74 define('AC_URL_SUBDIR', '/acprod');  
75 define('AC_URL_SCRIPT', '/index.php');  
76 define('AC_URL_PARAM', 'go');  
77 define('AC_URL_PREFIX', AC_URL_SUBDIR . AC_URL_SCRIPT . '?'  
78     . AC_URL_PARAM . '=');  
79 define('AC_ERROR_REPORTING', E_ERROR);  
80 // no slash at the end of this url  
81 define('URL_SITE', 'http://10.1.xx.xxx');  
82 define('OS', 'NIX'); // *NIX WIN BSD MAC  
83 // default database (read)  
84 define('DB_TYPE_READ', 'MYSQL');  
85 define('DB_HOST_READ', '127.0.0.1');  
86 // usually leave this blank for MYSQL  
87 define('DB_PORT_READ', '');  
88 define('DB_USER_READ', 'assetcentral');  
89 define('DB_PASS_READ', 'xxxxxx');  
90 define('DB_DATA_READ', 'asset_prod');  
91 define('DB_PREFIX_READ', '');
```

92 3.1.9 Database Update – Add a View

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

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

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

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

```
101 create view assetview as
```

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

```

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

```

```

108     from assets a
109     left join racks rack on a.rack_id = rack.rack_id
110     left join systems s on a.system_id = s.system_id
111     left join contacts c on a.contact_id = c.contact_id
112     left join rooms room on rack.room_id = room.room_id
113     left join floors floor on room.floor_id = floor.floor_id
114     where a.asset_deleted != 1;

```

115 Create a new database user and assign that user privileges on the assetview view (from inside of
 116 the MySQL client application):

```

117     create new users and privileges inside mysql/mariadb
118     create user 'asset_query'@'localhost';
119     set password for 'asset_query'@'localhost' = password('password');
120     grant select on assetcentral.assetview to 'asset_query'@'localhost';
121     grant file on *.* to 'asset_query'@'localhost';

```

122 Lastly, ensure that the MySQL network port is listening and is allowed through the firewall. You
 123 must be root to run these commands.

124 To verify that MySQL is listening:

```

125     netstat -l |grep mysql

```

126 To allow MySQL through the firewalld firewall:

```

127     firewall-cmd --permanent --add-service=mysql
128     firewall-cmd --reload

```

129 To make sure the firewall rule was added correctly:

```

130     firewall-cmd --list-services

```

131 3.1.10 Add Assets into AssetCentral

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

134 AssetCentral provides a manual method of adding one or two assets as well as an automated
 135 method of adding numerous assets that have been saved in a spreadsheet. There are detailed
 136 instructions for setting things up and adding assets on the AssetCentral page:

137 http://help.alphapoint-us.net/w/index.php/Starting_From_Scratch.

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 <http://www.belarc.com/belmanage.html>.

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 Bel Manage 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.
 - 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.

- For Linux machines:

The BelMonitor client must be installed as the root user.

- 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
```

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

```
apt-get install rpm
```

- Make the BelMonitorLinux executable.

```
chmod a+x BelMonitorLinux
```

- 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

- 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.
- Right-click MS SQL Server Studio Manager and select **Run as Administrator**.
- Click **Connect** as the default settings should be correct:
Server type: **Database Engine**
Server name: **BELARC\BELMANAGE**
Authentication: **Windows Authentication**
- Once MS SQL Server Management Studio has logged in and started, create a new database user.
 - Select **Security > Logins**.
 - Right-click **Logins** and select **New User**.
 - Enter a **Login name**.
 - Select SQL Server authentication.
 - Enter a password.
 - Enter the password again in the **Confirm password** box.
 - The Enforce password policy, **Enforce password expiration** and **User must change password at next login** should all reflect your organization's security rules.

- 239 Default database = **BelMonitor82_1**
- 240 Default language = **English**
- 241 5. Add the new user that you created in the preceding steps to the **BelMonitor82_1** database.
- 242 a. Select **Databases > BelMonitor82_1 > Security > Users**.
- 243 b. Right-click **Users** and select **New User**.
- 244 c. Enter a user name for the new user in the **User Name** and **Login Name** fields. They
- 245 should be identical.
- 246 Default schema = db_datareader
- 247 Schemas owned by this user = none selected
- 248 d. Database role membership: **BelMonitorReader** and **db_datareader** should be checked.
- 249 6. Turn on or re-enable any security settings that you might have changed, such as AppLocker.

250 3.3 Bro

251 Bro is an open-source network security monitor. Bro efficiently analyzes all network traffic and

252 provides insight into clear text password use, cryptographic certificate errors, traffic to known

253 bad sites, network flow, and file transfers.

254 3.3.1 How It's Used

255 In the FS ITAM build, Bro monitors all traffic traversing the DMZ. Bro has a dedicated network

256 interface in promiscuous mode for sniffing/capturing traffic. This interface does not have an IP

257 address assigned. Bro has a second network interface for management that is assigned IP

258 address 172.16.0.20. When configuring Bro, make sure that Bro is sniffing/capturing on the

259 correct network interface.

260 On the high-level architecture diagram, Bro is in Tier 2. Bro uses the Splunk Universal Forwarder

261 to send logs to Splunk Enterprise. Some of the logs include files, Hypertext Transfer Protocol

262 (HTTP) traffic, Kerberos authentications, Secure Socket Layer (SSL) traffic, x509 certificates

263 seen, known hosts, DNS traffic, all connections, notices, and intelligence alerts.

264 3.3.2 Virtual Machine Configuration

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

266 CPU cores.

267 3.3.3 Network Configuration

268 The management network interface card is configured as follows:

269 IPv4 Manual

270 IPv6 Ignore/Disabled

271 IP Address: 172.16.0.20

272 Netmask: 255.255.255.0

273 Gateway: 172.16.0.11

274 DNS Servers: 172.16.1.20, 172.16.1.21

275 Search Domains: lab5.nccoe.gov

276 3.3.4 Installing Bro

277 Bro is installed on a hardened Ubuntu 14.04 Linux system. Please download the latest source
278 package from Bro and follow the instructions for installing from source. Installation was
279 performed following the instruction from Bro at:

280 <https://www.bro.org/sphinx/install/index.html>

281 3.3.4.1 Installation Prerequisites

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

- 283 ■ Libpcap (<http://www.tcpdump.org>)
- 284 ■ OpenSSL libraries (<http://www.openssl.org>)
- 285 ■ BIND8 library
- 286 ■ Libz
- 287 ■ Bash (for BroControl)
- 288 ■ Python (for BroControl)

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

- 290 ■ CMake 2.8 or greater (<http://www.cmake.org>)
- 291 ■ Make
- 292 ■ C/C++ compiler
- 293 ■ SWIG (<http://www.swig.org>)
- 294 ■ Bison (GNU Parser Generator)
- 295 ■ Flex (Fast Lexical Analyzer)
- 296 ■ Libpcap headers (<http://www.tcpdump.org>)
- 297 ■ OpenSSL headers (<http://www.openssl.org>)
- 298 ■ zlib headers
- 299 ■ Perl

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
```

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
```

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
```

Add Bro bin directory to your runtime path:

Edit `.bashrc`

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

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

Then:

```
source .bashrc
```

To start Bro the first time:

```
broctl deploy
```

335 To check the status of Bro:

336 `broctl status`

337 3.3.5 Installing Intelligence Gathering Software

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

340 `cd /opt`

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

342 `cd /opt/mal-dnssearch`

343 `sudo make`

344 `sudo make install`

345 `mkdir /usr/local/bro_intel`

346 `cd /usr/local/bro_intel`

347 Copy the `update_intel.sh` script into `/usr/local/bro_intel`

348 `cp update_intel.sh /usr/local/bro_intel`

349 `chmod 700 /usr/local/bro_intel/update_intel.sh`

350 `cd /usr/local/bro_intel`

351 `./update_intel.sh`

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

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

355 `ln -s /usr/local/bro_intel/update_intel.sh`

356 `/etc/cron.daily/update_intel`

357 3.3.6 Configuring Bro

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

361 Step 1: Stop Bro.

362 `broctl stop`

363 Step 2: Copy and edit `node.cfg`.

364 `cp /usr/local/bro/etc/node.cfg /usr/local/bro/etc/node.cfg.orig`

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

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

Step 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

Step 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/
```

Step 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

Download the Splunk Universal Forwarder from:

http://www.splunk.com/en_us/download/universal-forwarder.html

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:

splunkforwarder-6.2.5-272645-linux-2.6-amd64.deb

Detailed installation instructions can be found at:

http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_install

An abridged version follows:

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

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

This will install in /opt/splunkforwarder:

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

405 Add forwarder:

406 More information about adding a forwarder can be found at:

407 <http://docs.splunk.com/Documentation/Splunk/6.2.4/Forwarding/Deployanixdfmanually>

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

410 3.3.8 Configuring Splunk Universal Forwarder

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

414 Create a directory to hold your certificates:

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

416 Copy your certificates in PEM format to /opt/splunkforwarder/etc/certs:

```
417 cp CASServerCert.pem /opt/splunkforwarder/etc/certs
418 cp bro_worker1.pem /opt/splunkforwarder/etc/certs
```

419 Copy the Splunk Universal Forwarder configuration files:

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

423 Modify server.conf so that:

- 424 • **ServerName=Bro** is your hostname.
- 425 • **sslKeysfilePassword** = <password for your private key>

426 Modify outputs.conf so that:

- 427 • **Server = loghost:9997** is your correct Splunk Enterprise server/indexer and port.
- 428 • **sslPassword** = <password of your certificate private key>

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

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

432 **Note:** dns.log, conn.log and http.log generate a significant volume of messages for Splunk
433 Enterprise to index. Depending on the size of your Splunk Enterprise license, this data volume
434 might cause license warnings or violations. See
435 <http://docs.splunk.com/Documentation/Splunk/6.2.6/Admin/Aboutlicenseviolations> for more
436 information.

3.3.9 Configurations and Scripts

```
437 Update_intel.sh should be placed in /usr/local/bro_intel.
438
439 #!/bin/sh
440 # This script downloads and formats reputation data from the Internet
441 and formats it so that Bro can use it as intel data.
442 # Good idea to restart bro every now and then:    broctl restart
443 # /usr/local/bro/share/bro/site/local.bro looks for the files in this
444 directory.
445 #
446 # Uses the mal-dnssearch package from Jon Schipp
447 # git clone https://github.com/jonschipp/mal-dnssearch
448 # cd mal-dnssearch
449 # sudo make install
450 #
451
452 cd /usr/local/bro_intel
453
454 # download and format the Mandiant APT info
455 mal-dnssearch -M mandiant -p | mal-dns2bro -T dns -s mandiant -n true >
456 /usr/local/bro_intel/mandiant.intel
457
458 # download and format TOR info
459 mal-dnssearch -M tor -p | mal-dns2bro -T ip -s tor -n true -u
460 http://rules.emergingthreats.net/open/suricata/rules/tor.rules >
461 /usr/local/bro_intel/tor.intel
462
463 # download and format Alienvault reputation info
464 mal-dnssearch -M alienvault -p | mal-dns2bro -T ip -s alienvault -n
465 true > /usr/local/bro_intel/alienvault.intel
```

```
462 /usr/local/bro/etc/node.cfg
463 # Example BroControl node configuration.
464 #
465 # This example has a standalone node ready to go except for possibly
466 changing
467 # the sniffing interface.
468
469 # This is a complete standalone configuration. Most likely you will
470 # only need to change the interface.
471 [bro]
472 type=standalone
473 host=localhost
474 interface=eth1
475
476 ## Below is an example clustered configuration. If you use this,
477 ## remove the [bro] node above.
478
479 #[manager]
480 #type=manager
481 #host=host1
482 #
483 #[proxy-1]
484 #type=proxy
485 #host=host1
486 #
487 #[worker-1]
488 #type=worker
489 #host=host2
490 #interface=eth0
491 #
492 #[worker-2]
493 #type=worker
494 #host=host3
495 #interface=eth0
496 #
497 #[worker-3]
498 #type=worker
499 #host=host4
500 #interface=eth0
```

```
501 /usr/local/bro/share/bro/site/local.bro
502
503 ##! Local site policy. Customize as appropriate.
504 ##!
505 ##! This file will not be overwritten when upgrading or reinstalling!
506
507 # Capture plaintext passwords
508 redef HTTP::default_capture_password=T;
509 redef FTP::default_capture_password=T;
510
511 #Hash all HTTP - for APT script
512 #redef HTTP::generate_md5=/.*/;
513
514 # This script logs which scripts were loaded during each run.
515 @load misc/loaded-scripts
516
517 # Apply the default tuning scripts for common tuning settings.
518 @load tuning/defaults
519
520 # Load the scan detection script.
521 @load misc/scan
522
523 # Log some information about web applications being used by users
524 # on your network.
525 @load misc/app-stats
526
527 # Detect traceroute being run on the network.
528 @load misc/detect-traceroute
529
530 # Generate notices when vulnerable versions of software are discovered.
531 # The default is to only monitor software found in the address space
532 # defined
533 # as "local". Refer to the software framework's documentation for more
534 # information.
535 @load frameworks/software/vulnerable
536
537 # Detect software changing (e.g. attacker installing hacked SSHD).
538 @load frameworks/software/version-changes
539
540 # This adds signatures to detect cleartext forward and reverse windows
541 # shells.
542 @load-sigs frameworks/signatures/detect-windows-shells
```

```
543      # Uncomment the following line to begin receiving (by default hourly)
544      emails
545      # containing all of your notices.
546      # redef Notice::policy += { [$action = Notice::ACTION_ALARM, $priority
547      = 0] };
548
549      # Load all of the scripts that detect software in various protocols.
550      @load protocols/ftp/software
551      @load protocols/smtp/software
552      @load protocols/ssh/software
553      @load protocols/http/software
554      # The detect-webapps script could possibly cause performance trouble
555      when
556      # running on live traffic. Enable it cautiously.
557      #@load protocols/http/detect-webapps
558
559      # This script detects DNS results pointing toward your Site::local_nets
560      # where the name is not part of your local DNS zone and is being hosted
561      # externally. Requires that the Site::local_zones variable is defined.
562      @load protocols/dns/detect-external-names
563
564      # Load dhcp script to log known devices
565      @load protocols/dhcp/known-devices-and-hostnames
566
567      # Script to detect various activity in FTP sessions.
568      @load protocols/ftp/detect
569
570      # Scripts that do asset tracking.
571      @load protocols/conn/known-hosts
572      @load protocols/conn/known-services
573      @load protocols/ssl/known-certs
574
575      # This script enables SSL/TLS certificate validation.
576      @load protocols/ssl/validate-certs
577
578      # Check for SSL Heartbleed attack
579      @load protocols/ssl/heartbleed
580
581      # Check for weak keys
582      @load protocols/ssl/weak-keys
583
584      # Check for expiring certs
585      @load protocols/ssl/expiring-certs
586
```

```

587     # Uncomment the following line to check each SSL certificate hash
588     against the ICSI
589     # certificate notary service; see http://notary.icsi.berkeley.edu .
590     @load protocols/ssl/notary
591
592     # If you have libGeoIP support built in, do some geographic detections
593     and
594     # logging for SSH traffic.
595     @load protocols/ssh/geo-data
596     # Detect hosts doing SSH bruteforce attacks.
597     @load protocols/ssh/detect-bruteforcing
598     # Detect logins using "interesting" hostnames.
599     @load protocols/ssh/interesting-hostnames
600
601     # Detect SQL injection attacks.
602     @load protocols/http/detect-sqli
603
604     const feed_directory = "/usr/local/bro_intel";
605
606     # Intelligence framework
607     #@load policy/frameworks/intel/seen
608     #@load policy/frameworks/intel/do_notice
609     @load frameworks/intel/seen
610     @load frameworks/intel/do_notice
611
612     #@load policy/integration/collective-intel
613     #redef Intel::read_files += {
614     # feed_directory + "/mandiant.intel",
615     # feed_directory + "/tor.intel",
616     # feed_directory + "/alienvault.intel",
617     ##"/usr/local/bro/share/bro/site/bad_domains.txt",
618     ##"/somewhere/yourdata1.txt",
619     #};
620     redef Intel::read_files += {
621         "/usr/local/bro_intel/mandiant.intel",
622         "/usr/local/bro_intel/tor.intel",
623         "/usr/local/bro_intel/alienvault.intel",
624     };
625
626     ##### Network File Handling #####
627
628     # Enable MD5 and SHA1 hashing for all files.
629     @load frameworks/files/hash-all-files

```

```
630      # Detect SHA1 sums in Team Cymru's Malware Hash Registry.
631      @load frameworks/files/detect-MHR
632
633      # Extract collected files
634      #@load extract_files
635
636      # this is the original malware_detect using perl and clamavd
637      #@load malware_detect
638
639      # can define this stuff here or in the site specific .bro scripts
640      #redef Communication::listen_port = 47777/tcp;
641      #redef Communication::nodes += {
642      # ["broping"] = [$host = 127.0.0.1, $class="broping", $events = /ping/,
643      $connect = F, $ssl = F],
644      # ["malware_detect"] = [$host = 127.0.0.1, $class="malware_detect",
645      $events = /malware_message/, $connect= F, $ssl = F]
646      #};
647
648      #@load malware1
649      #@load broccoli
650      #@load whitelisting
651      #@load broping
652
653      event bro_init() {
654          Analyzer::disable_analyzer(Analyzer::ANALYZER_SYSLOG);
655      }
656
657      #event bro_init()
658      # {
659      # local f = Log::get_filter(Notice::ALARM_LOG, "alarm-mail");
660      # f$interv = 1day;
661      # Log::add_filter(Notice::ALARM_LOG, f);
662      # }

```

```
663      /opt/splunkforwarder/etc/system/local/server.conf
664
665      [sslConfig]
666
667      sslKeysfilePassword = $1$20Js1XSip3Un
668
669      [lmpool:auto_generated_pool_forwarder]
670      description = auto_generated_pool_forwarder
671      quota = MAX
672      slaves = *
673      stack_id = forwarder
```

```
672     [lmpool:auto_generated_pool_free]
673     description = auto_generated_pool_free
674     quota = MAX
675     slaves = *
676     stack_id = free
677
678     [general]
679     pass4SymmKey = $1$j644iTHO7Ccn
680     serverName = bro
```

```
681     /opt/splunkforwarder/etc/system/local/inputs.conf
682     [default]
683     host = bro
684     sourcetype=BroLogs
685     index=bro
686
687     [monitor:///usr/local/bro/logs/current/notice.log]
688     sourcetype=bro_notice
689     [monitor:///usr/local/bro/logs/current/weird.log]
690     sourcetype=bro_weird
691     [monitor:///usr/local/bro/logs/current/ssl.log]
692     sourcetype=bro_ssl
693     [monitor:///usr/local/bro/logs/current/ssh.log]
694     sourcetype=bro_ssh
695     [monitor:///usr/local/bro/logs/current/software.log]
696     sourcetype=bro_software
697     [monitor:///usr/local/bro/logs/current/intel.log]
698     sourcetype=bro_intel
699     [monitor:///usr/local/bro/logs/current/http.log]
700     sourcetype=bro_http
701     [monitor:///usr/local/bro/logs/current/conn.log]
702     sourcetype=bro_conn
703     [monitor:///usr/local/bro/logs/current/x509.log]
704     sourcetype=bro_x509
705     [monitor:///usr/local/bro/logs/current/dns.log]
706     sourcetype=bro_dns
707
708     #[monitor:///usr/local/bro/logs/current/*.log]
709     #host=bro-worker1
710     #sourcetype=BroLogs
711     #index=bro
712
713     #[monitor:///opt/splunkforwarder/var/log/splunk/splunkd.log]
```

```
714 /opt/splunkforwarder/etc/system/local/outputs.conf
715 [tcpout]
716 defaultGroup = splunkssl
717
718 [tcpout:splunkssl]
719 server = loghost:9997
720 compressed = true
721 sslVerifyServerCert = false
722 sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem
723 sslCertPath = $SPLUNK_HOME/etc/certs/bro-worker1.pem
724 sslPassword = $1$23DtXas9IZD8
```

725 3.4 CA Technologies IT Asset Manager

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

731 3.4.1 How It's Used

732 In the FS ITAM build, CA ITAM is used to track hardware assets from requisition to disposal. Data
733 collected during this task will be analyzed and used to notify an administrator of a change in the
734 network architecture. When a new hardware asset is received, an administrator will enter into
735 the database information that includes, but is not limited to, the asset name, host name,
736 operating system, serial number, owner, location, mac address and IP address. The data is then
737 stored for retrieval by Splunk Enterprise. For this particular build, the CA ITAM database is
738 pre-loaded with data from machines being used throughout the ITAM architecture. The Tier 1
739 ITAM server is connected to the CA ITAM database to query data stored in the CA ITAM resource
740 tables.

741 3.4.2 Virtual Machine Configuration

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

745 3.4.3 Network Configuration

746 The management network interface card is configured as follows:

747 IPv4 Manual

748 IPv6 Disabled

749 IP Address: 172.16.3.92

750 Netmask: 255.255.255.0

751 Gateway: 172.16.3.11

752 DNS Servers: 172.16.1.20, 172.16.1.21

753 Search Domains: lab5.nccoe.gov

754 3.4.4 Installing CA ITAM

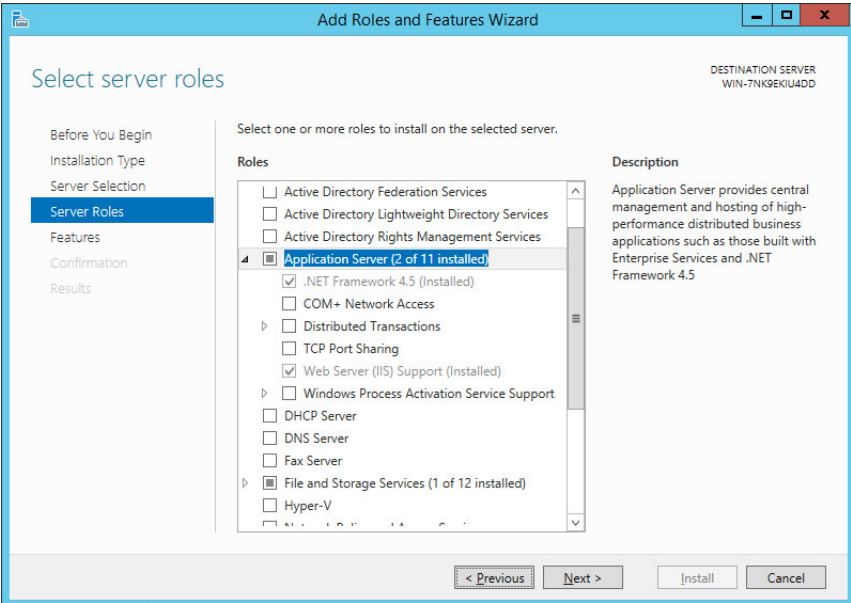
755 CA ITAM is installed on a clean 64-bit Windows Server 2012 R2 image with default Windows
756 firewall configurations. Installation configurations are default for this build and are documented
757 online by CA Technologies. CA Technologies installation guidelines can be found online at the
758 following URL:

759 [https://support.ca.com/cadocs/0/CA%20IT%20Asset%20Manager%2012%208-ENU/Bookshelf](https://support.ca.com/cadocs/0/CA%20IT%20Asset%20Manager%2012%208-ENU/Bookshelf_Files/PDF/APM_Impl_ENU.pdf)
760 [_Files/PDF/APM_Impl_ENU.pdf](https://support.ca.com/cadocs/0/CA%20IT%20Asset%20Manager%2012%208-ENU/Bookshelf_Files/PDF/APM_Impl_ENU.pdf)

761 Prerequisites for this build are as follows:

- 762 ■ Java 7 JRE (32-bit)
 - 763 • Set the JAVA_HOME variable
- 764 ■ SQL Server 2012 with
 - 765 • Database Engine
 - 766 • Backwards Compatibility
 - 767 • Client Connectivity
 - 768 • Management tools
 - 769 • Used mixed authentication as the authentication method
- 770 ■ NET Framework 3.5
- 771 ■ NET Framework 4.5
 - 772 • Select ASP.NET
- 773 ■ IIS

774 **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

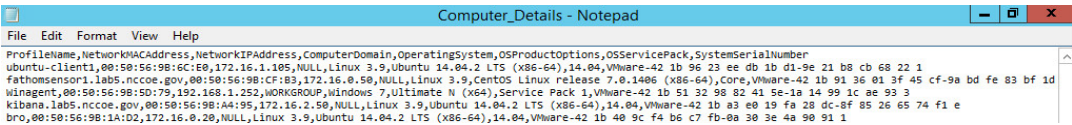
3.4.5.1 Data Import

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**.

The screenshot shows the 'Data Importer' interface in the 'Administration' tab. The 'Basic Information' section includes fields for Name, Description, Legacy Map File, Data File, Upload File, Main Destination Object, First Row Has Column Names, Data File Locale, and Data Delimiter. The 'Advanced Settings' section includes Maximum Error Threshold, Primary Lookup Object Processing Type, and checkboxes for Create Secondary Lookup Object, Update Secondary Lookup Object, and Error on Secondary Lookup Object Errors. The 'Mapping' tab is also visible at the bottom.

3. In the **Administration** tab, specify these settings:
 - **Name:** <Name>
 - **Data File:** <filename>
 - **Main Destination Object:** Asset(Computer)
 - Select **First Row Has Column Names**
 - **Data File Locale:** English (United States)
 - **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.
 - **Computer domain** = Asset.Host Name
 - **NetworkIPAddress** = Asset.IP Address
 - **NetworkMACAddress** = Asset.MAC Address
 - **OperatingSystem** = Asset.Model.Model Name
 - **OSProductOptions** = Asset.Asset Type Hierarchy.Class.Value
 - **OSServicePack** = Asset.Asset Type Hierarchy.Subclass.Value
 - **ProfileName** = Asset.Asset Name
 - **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.

813 9. Check the status of the job under **Import Jobs**.

814 10. Use the data stored in the MDB to run a query through the Splunk DB Connection (See
815 [section 2.1.1, Splunk Enterprise](#) to configure.).

816 11. Query is as follows:

```
817 SELECT DISTINCT
818 aud_ca_owned_resource.resource_name,audit_mode_uuid,audit_resource_
819 class,audit_resource_subclass,ca_owned_resource.own_resource_id,ca_
820 owned_resource.mac_address,ca_owned_resource.ip_address,ca_owned_re
821 source.host_name,ca_owned_resource.serial_number,ca_owned_resource.
822 asset_source_uuid,ca_owned_resource.creation_user,ca_owned_resource
823 .creation_date
824 FROM aud_ca_owned_resource
825 INNER JOIN ca_owned_resource
826 ON aud_ca_owned_resource.resource_name =
827 ca_owned_resource.resource_name
```

828 3.5 Fathom Sensor from RedJack

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

832 3.5.1 How It's Used

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

837 3.5.2 Virtual Machine Configuration

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

840 3.5.3 Network Configuration

841 The management network interface card is configured as such:

842 IPv4 Manual

843 IPv6 Disabled

844 IP Address: 172.16.0.50

845 No IP address for the second network interface card

846 Netmask: 255.255.255.0

847 Gateway: 172.16.0.11

848 DNS Servers: 172.16.1.20, 172.16.1.21

849 Search Domains: lab5.nccoe.gov

850 3.5.4 Installing Fathom Sensor

851 VM Deployments

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

854 Requirements

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

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

861 Configure the VM

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

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

865 * ESX - [KB:
866 1004099](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1004099)
867

868 * Linux - [KB:
869 287](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=287)
870

871 * Fusion - Password prompt can be disabled under Preferences > Network.

872 Install CentOS 7 Minimal

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

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`

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
```

Configure the monitoring interface without an IP:

```
# /etc/sysconfig/network-scripts/ifcfg-eno2
```

```
BOOTPROTO=static
```

```
ONBOOT=yes
```

Disable IPv6 autoconfiguration on the monitoring interface:

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

Configure DNS

```
# vi /etc/resolv.conf
```

```
search lab5.nccoe.gov
```

```
nameserver 172.16.1.20
```

```
nameserver 172.16.1.21
```

Set the hostname

```
# hostnamectl set-hostname fathomsensor1
```

```
# vi /etc/hosts
```

```
127.0.0.1 localhost
```

```
172.16.0.50 fathomsensor1
```

Adjust the Packages

```
# Not required, but if you are planning to install VMWare Tools, you need
```

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

```
# Install basic tools
```

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

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
```

Disable SELinux

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

```
SELINUX=permissive
```

Limit SSH

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

```
ListenAddress 172.16.0.50
```

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
```

942 **disable monitor**

943 Limit NTP to only listening on the management interface:

```
944       #vi /etc/sysconfig/ntpd
945       OPTIONS="-g -I eno1 -I 172.16.0.50"
```

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

```
947       $ ntpdate 172.16.0.11
948       $ hwclock -w
```

949 Set NTP to start:

```
950       $ systemctl enable ntpd
951       $ systemctl start ntpd
```

952 CollectD

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

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

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

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

962 Then enable collectd:

```
963       $ systemctl enable collectd
964       $ systemctl start collectd
```

965 Install Fathom-Sensor

966 First install all the sensor RPMs:

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

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

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

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

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

975 Update dynamic run-time bindings because sometimes it needs it:

```
976       $ ldconfig
```

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
```

Install and Configure Amazon S3 Command Line Tools using PIP

<http://docs.aws.amazon.com/cli/latest/userguide/installing.html>

Verify that you have at least Python 2.7:

```
$ python -version
```

Download the pip installation script:

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

Run the pip installation script

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

Install the AWS CLI

```
$ sudo pip install awscli
```

Configure AWS CLI

```
#aws configure
```

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
```

Create a directory to save the files gathered from Amazon AWS

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

Create a script to sync data with the Amazon AWS

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

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
```

Make the script executable

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

```
1010 Make the script run every hour by placing a link in /etc/cron.hourly
1011 #cd /etc/cron.hourly
1012 #ln -s /usr/local/bin/fathom-sync.sh /etc/cron.hourly/fathom-sync
```

1013 3.5.5 Installing Splunk Universal Forwarder

1014 **Note:** You will need a Splunk account to download the Splunk Universal Forwarder. It is free and
1015 can be setup at:

1016 https://www.splunk.com/page/sign_up

1017 Download the Splunk Universal Forwarder from:

1018 http://www.splunk.com/en_us/download/universal-forwarder.html

1019 Use the latest version for **OS version 2.6+ kernel Linux distributions (64-bit)**. Since this is
1020 installing on Ubuntu select the file that ends in **.deb**. An example is:

1021 **splunkforwarder-6.2.5-272645-linux-2.6-amd64.deb**

1022 Detailed installation instructions can be found at:

1023 <http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinux>

1024 An abridged version follows:

```
1025 rpm -i <splunk_package_name.deb>
```

1026 Example: `rpm -i splunkforwarder-6.2.4-271043-linux-2.6-x86_64.rpm`

1027 This will install in */opt/splunkforwarder*

```
1028 cd /opt/splunkforwarder/bin
```

```
1029 ./splunk start --accept-license
```

```
1030 ./splunk enable boot-start
```

1031 Add forwarder:

1032 More info about adding a forwarder can be found at:

1033 <http://docs.splunk.com/Documentation/Splunk/6.2.4/Forwarding/Deployanixdfmanually>

```
1034 cd /opt/splunkforwarder/bin
```

```
1035 ./splunk add forward-server loghost:9997 -auth admin:changme
```

1036 3.5.6 Configuring Splunk Universal Forwarder

1037 Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509
1038 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You
1039 will also need a copy of your certificate authority's public certificate.

1040 Create a directory to hold your certificates:

```
1041 mkdir /opt/splunkforwarder/etc/certs
```

Copy your certificates in PEM format to `/opt/splunkforwarder/etc/certs`:

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

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

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
```

Modify `server.conf` so that:

ServerName=Bro is your hostname.

sslKeysfilePassword = <password for your private key>

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.

- List your sensor(s)

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

- List data types for a sensor

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

- List dates for the client-banner data type

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

- List individual JSON files on that date

```
aws s3 ls
```

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

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

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

1070 3.5.8 Configurations and Scripts

```
1071 /opt/splunkforwarder/etc/system/local/server.conf
1072 [sslConfig]
1073 sslKeysfilePassword = $1$20Js1XSip3Un
1074 [lmpool:auto_generated_pool_forwarder]
1075 description = auto_generated_pool_forwarder
1076 quota = MAX
1077 slaves = *
1078 stack_id = forwarder
1079 [lmpool:auto_generated_pool_free]
1080 description = auto_generated_pool_free
1081 quota = MAX
1082 slaves = *
1083 stack_id = free
1084 [general]
1085 pass4SymmKey = $1$j644iTHO7Ccn
1086 serverName = fathomsensor1.lab5.nccoe.gov
1087 /opt/splunkforwarder/etc/system/local/inputs.conf
1088 [default]
1089 host = fathomsensor1.lab5.nccoe.gov
1090 sourcetype=fathomsensor
1091 index=fathom
1092 [monitor:///opt/fathom-sync/*/client-banner*]
1093 /opt/splunkforwarder/etc/system/local/outputs.conf
1094 [tcpout]
1095 defaultGroup = splunkssl
1096 [tcpout:splunkssl]
1097 server = loghost:9997
1098 compressed = true
1099 sslVerifyServerCert = false
1100 sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem
1101 sslCertPath = $SPLUNK_HOME/etc/certs/fathomsensor1.lab5.nccoe.gov.pem
1102 sslPassword = $1$23DtXas9IZD8
```

1103 3.6 OpenVAS

1104 OpenVAS is an open-source network vulnerability scanner and manager. OpenVAS run
1105 customizable scans and generates reports in multiple formats. OpenVAS is also a framework,
1106 and additional tools can be added to it.

1107 3.6.1 How It's Used

1108 In the FS ITAM build, OpenVAS automatically runs vulnerability scans on all systems connected
1109 to the network. Every machine is scanned at least once a week. OpenVAS collects the
1110 information, stores it in a database, and creates reports. OpenVAS can also download the latest
1111 vulnerabilities along with their CVE and NVT information.

1112 On the high-level architecture diagram, OpenVAS is in Tier 2. OpenVAS utilizes the Splunk
1113 Universal Forwarder to send reports to Splunk Enterprise. Information is extracted from the
1114 OpenVAS database every hour, and any new records are forwarded to Splunk Enterprise. Splunk
1115 Enterprise uses the information from OpenVAS to provide context to analysts regarding the
1116 security of individual systems as well as aggregating statistics to show the overall organizational
1117 security posture.

1118 3.6.2 Virtual Machine Configuration

1119 The OpenVAS virtual machine is configured with one network interface card, 16 GB of RAM and
1120 four CPU cores.

1121 3.6.3 Network Configuration

1122 The management network interface card is configured as follows:

1123 IPv4 Manual

1124 IPv6 Ignore/Disabled

1125 IP Address: 172.16.2.33

1126 Netmask: 255.255.255.0

1127 Gateway: 172.16.2.11

1128 DNS Servers: 172.16.1.20, 172.16.1.21

1129 Search Domains: lab5.nccoe.gov

1130 <https://www.digitalocean.com/community/tutorials/how-to-use-openvas-to-audit-the-security-of-remote-systems-on-ubuntu-12-04>
1131

1132 3.6.4 Installation Prerequisites

1133 `sudo apt-get update`

1134 `sudo apt-get install python-software-properties`

```
1135 sudo apt-get install sqlite3 xsltproc texlive-latex-base
1136 texlive-latex-extra texlive-latex-recommended htmldoc alien rpm nsis
1137 fakeroot
```

1138 3.6.5 Installing OpenVAS

1139 OpenVAS is installed on a hardened Ubuntu 14.04 Linux system. Please download the latest
1140 source package from OpenVAS and follow the instructions for installing from source.
1141 Installation was performed following the instructions gathered from the following web sites:

1142 <http://www.openvas.org/>

1143 <https://www.digitalocean.com/community/tutorials/how-to-use-openvas-to-audit-the-security-of-remote-systems-on-ubuntu-12-04>

1144 <https://launchpad.net/~openvas/+archive/ubuntu/openvas6>

1145 Add new file in /etc/apt/sources.list.d/openvas-openvas6-trusty.list

```
1146 deb http://ppa.launchpad.net/openvas/openvas6/ubuntu precise main
```

```
1147 deb-src http://ppa.launchpad.net/openvas/openvas6/ubuntu precise main
```

```
1149 sudo apt-get install openvas-manager openvas-scanner
```

```
1150 openvas-administrator openvas-cli greenbone-security-assistant
```

```
1151 sudo openvas-mkcert
```

1152 Answer the questions for the new certificate.

```
1153 sudo openvas-mkcert-client -n om -i
```

1154 Download and build the vulnerability database.

```
1155 sudo openvas-nvt-sync
```

1156 Stop the services.

```
1157 sudo service openvas-manager stop
```

```
1158 sudo service openvas-scanner stop
```

1159 Start the scanner application (this will download and sync a lot of data):

```
1160 sudo openvassd
```

1161 Rebuild the database.

```
1162 sudo openvasmd --rebuild
```

1163 Download and sync SCAP data.

```
1164 sudo openvas-scapdata-sync
```

1165 Download and sync cert data.

```
1166 sudo openvas-certdata-sync
```

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

Now sync the certs and everything should work.

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

Add user and permissions.

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

Edit the following file and insert your OpenVAS IP address.

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

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
```

Enable start up a 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
```

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.

Step 1: Set up schedules

Configuration > Schedules

Click the **Star** icon to create a new schedule.

Create a schedule for every day of the week. Example:

Monday scans - every day at 21:00

Do the same for the other 6 days of the week.

Step 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.

Configuration > Targets

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

Click **Create Target** button to save.

Step 3: Set up Tasks

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

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)

Click the **Create Task** button to save

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

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.

1233 The `openvas_results.py` is run by cron every hour to check for new results from OpenVAS scans.

1234 The Splunk Universal Forwarder checks the CSV file written by `openvas_results.py` for any

1235 changes and sends those to the Splunk Enterprise server/indexer.

1236 Place *openvas_results.py* in `/root` and make sure that it is executable:

1237 `cp <openvas_results.py> /root`

1238 `chmod +x /root/openvas_results.py`

1239 Create a symbolic link in `/etc/cron.hourly` so that `openvas_results.py` runs every hour.

1240 `ln -s /root/openvas_results.py /etc/cron.daily/openvas_results`

1241 3.6.7 Installing Splunk Universal Forwarder

1242 **Note:** You will need a Splunk account to download the Splunk Universal Forwarder. It is free and

1243 can be set up at:

1244 https://www.splunk.com/page/sign_up

1245 Download the Splunk Universal Forwarder from:

1246 http://www.splunk.com/en_us/download/universal-forwarder.html

1247 You want the latest version for OS version 2.6+ kernel Linux distributions

1248 (64-bit). Since this is installing on Ubuntu, select the file that ends in `.deb`. An example is:

1249 `splunkforwarder-6.2.5-272645-linux-2.6-amd64.deb`

1250 Detailed installation instructions can be found at:

1251 [http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_i](http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_install)

1252 [nstall](http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_install)

1253 An abridged version follows:

1254 `dpkg -i <splunk_package_name.deb>`

1255 Example: `dpkg -i splunkforwarder-6.2.5-272645-linux-2.6-amd64.deb`

1256 This will install in `/opt/splunkforwarder`:

1257 `cd /opt/splunkforwarder/bin`

1258 `./splunk start --accept-license`

1259 `./splunk enable boot-start`

1260 Add forwarder:

1261 More information about adding a forwarder can be found at:

1262 <http://docs.splunk.com/Documentation/Splunk/6.2.4/Forwarding/Deployanixdfmanually>

1263 `cd /opt/splunkforwarder/bin`

1264 `./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.

Create a directory to hold your certificates:

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

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

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

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

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
```

Modify server.conf so that:

- **ServerName=openvascd** is your hostname.
- **sslKeysfilePassword** = <password for your private key>

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
```

```

1301     import sqlite3
1302     import csv
1303     import json

1304     # Global variables and configs
1305     # SQLITE3 database file
1306     file_db = "/var/lib/openvas/mgr/tasks.db"

1307     # JSON file to write results to
1308     json_file = "/home/mike/openvas_results.json"

1309     # CSV file to write results to - actually tab delimited
1310     csv_file = "/home/mike/openvas_results.csv"

1311     # last_id is how we keep track of the last item added. This keeps us
1312     # from re-processing old items. This value is kept in the
1313     # openvas_state.txt file
1314     last_id = 0

1315     #openvas_state.txt - change this to 0 if you want to start over
1316     openvas_state_file = "/home/mike/openvas_state.txt"

1317     # this is just a status of how many records have be processed.
1318     new_record_count = 0

1319     print "Getting OpenVAS reports"

1320     if os.path.isfile(openvas_state_file) and
1321     os.access(openvas_state_file, os.W_OK):
1322         openvas_state = open(openvas_state_file, 'r+')
1323         last_id = openvas_state.read()
1324     else:
1325         print "File %s does not exist, creating" % openvas_state_file
1326         #sys.exit()
1327         openvas_state = open(openvas_state_file, 'w')
1328         openvas_state.write('0')

1329     print "Last ID = ", last_id

1330     # stripped removes non-printable characters
1331     def stripped(x):
1332         return "".join([i for i in x if 31 < ord(i) < 127])

1333     try:
1334         db_conn = sqlite3.connect(file_db, check_same_thread=False)
1335     except:
1336         print "Cannot connect to %s" % file_db
1337         sys.exit()

```

```
1338         db_cursor = db_conn.cursor()

1339         #query = """SELECT id, task, subnet, host, port, nvt, type,
1340         description, report from results"""

1341         query = """SELECT results.id, results.task, results.subnet,
1342         results.host, results.port, results.nvt, results.type,
1343         results.description, results.report, nvts.name, nvts.description,
1344         nvts.cve, nvts.cvss_base, nvts.risk_factor from results LEFT JOIN nvts
1345         ON results.nvt = nvts.uuid ORDER BY results.id"""

1346         #field_names = ['id', 'task', 'subnet', 'host', 'port', 'nvt', 'type',
1347         'results_description', 'report', 'nvts_name', 'nvts_description',
1348         'cve', 'cvss_base', 'risk_factor']

1349         csvfile = open(csv_file, 'a')
1350         csv_writer = csv.writer(csvfile, delimiter='\\t', quotechar='|',
1351         quoting=csv.QUOTE_MINIMAL)

1352         jsonfile = open(json_file, 'a')

1353         for row in db_cursor.execute(query):
1354             #print row
1355             id = row[0] #this needs to be a number
1356             task = stripped(str(row[1]))
1357             subnet = stripped(str(row[2]))
1358             host = stripped(str(row[3]))
1359             port = stripped(str(row[4]))
1360             nvt = stripped(str(row[5]))
1361             type = stripped(str(row[6]))
1362             results_description = stripped(str(row[7]))
1363             report = stripped(str(row[8]))
1364             nvts_name = stripped(str(row[9]))
1365             nvts_description = stripped(str(row[10]))
1366             cve = stripped(str(row[11]))
1367             cvss_base = stripped(str(row[12]))
1368             risk_factor = stripped(str(row[13]))

1369

1370             if int(id) > int(last_id):
1371                 #print "Greater!"
1372                 last_id = id
1373                 openvas_state.seek(0,0)
1374                 openvas_state.write(str(last_id))
1375                 new_record_count = new_record_count + 1

1376
```

```

1377         csv_writer.writerow([id, task, subnet, host, port, nvt, type,
1378 results_description, report, nvts_name, nvts_description, cve,
1379 cvss_base, risk_factor])
1380
1381         json_dict = {'id': id, 'task': task, 'subnet': subnet, 'host':
1382 host, 'port': port, 'nvt': nvt, 'type': type, 'results_description':
1383 results_description, 'report': report, 'nvts_name': nvts_name,
1384 'nvts_description': nvts_description, 'cve': cve, 'cvss_base':
1385 cvss_base, 'risk_factor': risk_factor}
1386         json.dump(json_dict, jsonfile, sort_keys = True, indent = 4,
1387 ensure_ascii = False)
1388
1389         #print "ID: %s  LAST: %s" % (id, last_id),
1390
1391     print "\n"
1392
1393     db_conn.close()
1394     csvfile.close()
1395     jsonfile.close()
1396
1397     print "Wrote %s new records." % new_record_count

```

```

1396 /opt/splunkforwarder/etc/system/local/server.conf
1397 [sslConfig]
1398 sslKeysfilePassword = $1$JnofjmZL66ZH
1399
1400 [lmpool:auto_generated_pool_forwarder]
1401 description = auto_generated_pool_forwarder
1402 quota = MAX
1403 slaves = *
1404 stack_id = forwarder
1405
1406 [lmpool:auto_generated_pool_free]
1407 description = auto_generated_pool_free
1408 quota = MAX
1409 slaves = *
1410 stack_id = free
1411
1412 [general]
1413 pass4SymmKey = $1$cTZL0iMNoPRH
1414 serverName = openvas

```

```
1412 /opt/splunkforwarder/etc/system/local/outputs.conf
1413 [tcpout]
1414 defaultGroup = splunkssl
1415 [tcpout:splunkssl]
1416 compressed = true
1417 server = loghost:9997
1418 sslCertPath = $SPLUNK_HOME/etc/certs/openvas.lab5.nccoe.gov.pem
1419 sslPassword = $!$JnofjmZL66ZH
1420 sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem
1421 sslVerifyServerCert = true
1422 /opt/splunkforwarder/etc/system/local/inputs.conf
1423 [default]
1424 host = openvas
1425 index = openvas
1426 sourcetype = openvas
1427 [monitor:///home/mike/openvas_results.csv]
```

1428 3.7 Puppet Enterprise

1429 Puppet Enterprise enforces a configuration baseline on servers and workstations. Puppet
1430 agents installed on the hosts will run periodically. Download a list of instructions referred to as a
1431 configuration catalog from the Master, and then execute it on the hosts. A successful Puppet
1432 Enterprise agent run can make configuration changes, install new software, remove unwanted
1433 software and send reports to the Master.

1434 3.7.1 How It's Used

1435 In the Financial Services ITAM solution, Puppet Enterprise is used to enforce a base
1436 configuration for all endpoints and to enforce basic security configurations. On the endpoints, it
1437 ensures that anti-virus software is installed, firewalls are enabled, IP forwarding is disabled and
1438 the software asset management agent is installed.

1439 Reporting is also a feature that was extended to in this solution. With the inclusion of
1440 customized scripts, Puppet Enterprise sends very valuable reports to the ITAM analysis engine.
1441 The reports include which endpoint has successfully uploaded reports to the Puppet Enterprise
1442 master. Failure to upload a report within a certain interval would indicate an anomaly with the
1443 endpoint or an off line endpoint. Puppet Enterprise's functionality was extended to remove
1444 blacklisted software listed in a file made available from an analyst. A script was written to parse
1445 the file on a daily basis, and inject the appropriate Puppet Enterprise code to remove such
1446 listed software. After successful removal, Puppet Enterprise writes a report identifying the
1447 offending endpoint, the uninstalled software and the time of removal.

1448 3.7.2 Prerequisites

1449 Puppet Enterprise Server requires the following:

- 1450 ■ at least a four core CPU, 6 GB of RAM and 100 GB of hard drive space
- 1451 ■ network-wide name resolution via DNS
- 1452 ■ network-wide time synchronization using NTP

1453 3.7.3 Installing Puppet Enterprise Server

1454 Instructions for installing Puppet Enterprise can be found at
 1455 http://docs.puppetlabs.com/pe/latest/install_pe_mono.html.

- 1456 1. Download the Puppet Enterprise tarball from the Puppet Labs web site. Use the instructions
 1457 referenced in the preceding link to locate and download the file.
- 1458 2. Run `tar -xf <PuppetEnterpriseTarball>` to unpack its contents.
- 1459 3. List directory with `ls` to view current directory contents.
- 1460 4. Change into the directory with name `puppet-enterprise-<version>-<OSversion>`.
- 1461 5. Execute `sudo ./puppet-enterprise-installer`.
- 1462 6. Connect to Puppet Enterprise Server console by going to:
 1463 **`https://YourPuppetServerFQDN:3000`**
- 1464 7. Accept the untrusted connection and make an exception to this site by storing it in your
 1465 trusted list.
- 1466 8. Confirm the security exception.
- 1467 9. From Installation Web page, select **Let's get started**.
- 1468 10. Select **Monolithic Installation**.
- 1469 11. Choose **Install on this Server**.
- 1470 12. Do not enable the Puppet 4 language parser if your existing Puppet code was developed in
 1471 Puppet 3.xx.
- 1472 13. Choose to install PostGreSQL on the same server.
- 1473 14. Supply a console password when prompted.

1474 3.7.4 Puppet Enterprise Linux Agent Installation

1475 To install Puppet Enterprise agent on the same platform as the server:

- 1476 1. Enter `curl -k`
 1477 `https://<YourPuppetServerFQDN>:8140/packages/current/install.bash`
 1478 `| sudo bash` at the agent terminal.
- 1479 2. Request a certificate by typing `puppet agent -t` from the client node.
- 1480 3. Go to the Puppet Enterprise server Web console and log in.

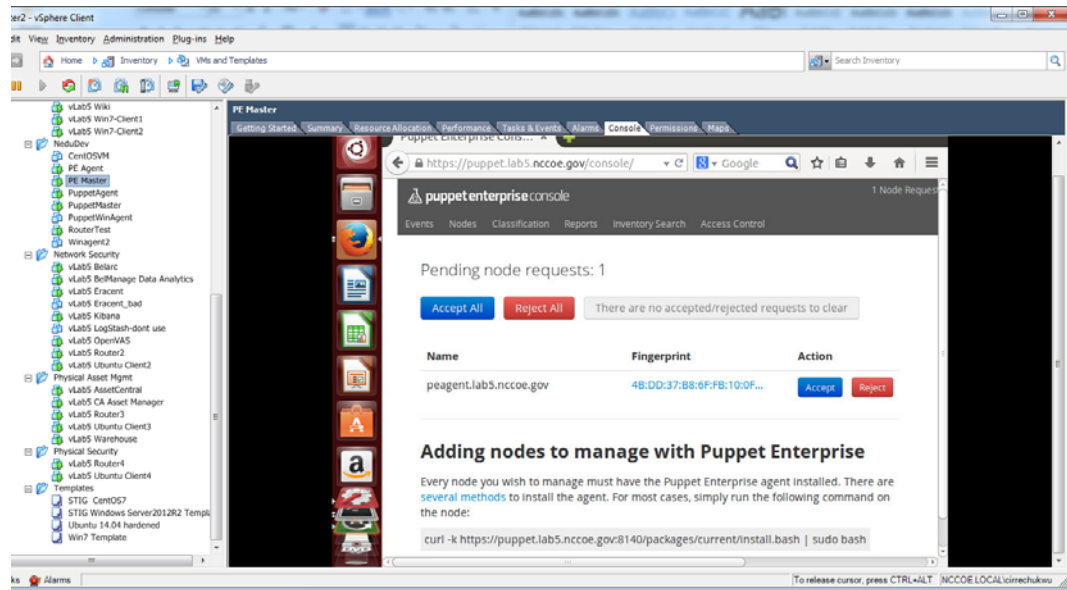
- 1481 4. Accept node requests by clicking on the **Node** link.
- 1482 5. Click **Accept** to sign the Certificate.
- 1483 To install Puppet Enterprise agent on a different platform from the server:
- 1484 1. Go to the Puppet Enterprise Web console.
- 1485 2. Click on **Classification**.
- 1486 3. Select the **PE Master Group**.
- 1487 4. Click the **Classes** tab.
- 1488 5. Select your platform from the new class textbox dropdown.
- 1489 6. Click **Add Class**.
- 1490 7. Click **Commit 1 Change**.
- 1491 8. Run puppet agent `-t` to configure the newly assigned class.
- 1492 9. To install the agent, enter `curl -k`
- 1493 `https://<YourPuppetServerFQDN>:8140/packages/current/install.bash |`
- 1494 `sudo bash`

1495 3.7.5 Puppet Enterprise Windows Agent Installation

- 1496 To install Puppet Enterprise agent on a Windows computer:
- 1497 1. Make sure to start the installation file or log in to the system with an administrator account.
- 1498 2. Double-click the Puppet Enterprise executable file.
- 1499 3. Accept the default options.

1500 3.7.6 Puppet Enterprise Agent Configuration

- 1501 1. Agents need to obtain certificates from the Puppet Enterprise Server/Master. Connect to
- 1502 the Puppet Enterprise Server console at `https://PuppetEnterpriseServerFQDN`.
- 1503 2. Log in to the console with your configured username and password.
- 1504 3. Click on **Nodes**.
- 1505 4. Accept Node requests from each agent you have configured. The agent's fully qualified
- 1506 domain name (FQDN) will be displayed.
- 1507 5. A certificate request can be generated if you do not see one by typing puppet agent `-t`
- 1508 from the agent terminal.
- 1509 6. Certificate requests can be viewed from the Web console of Puppet Enterprise Server.
- 1510 7. Windows agents offer the option of using the graphical user interface by clicking on
- 1511 **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`

1537 Each has a files directory `/etc/puppetlabs/puppet/modules/<modulename>/files` and a
1538 manifests directory with the
1539 `/etc/puppetlabs/puppet/modules/<modulename>/manifests/init.pp` file.

1540 3.7.7.1 Module: windowsnodes

1541 This module configures a baseline for Windows endpoints. Execution of this module copies a
1542 number of executable files and the baseline.bat script over to the endpoints from the Puppet
1543 Enterprise Server. Once baseline.bat is executed on the endpoint, it will look for and install the
1544 copied over executable programs, which consist of the belmonitor.exe asset management
1545 software agent and an anti-virus software. The text of the
1546 `/etc/puppetlabs/puppet/modules/windowsnodes/init.pp` manifest file is shown in the code and
1547 scripts section.

1548 3.7.7.2 Module: ubuntubase

1549 This module configures a baseline for Ubuntu endpoints. It installs software, disables IP
1550 forwarding, installs clamav anti-virus, and copies over files including a script *dailyscript* that
1551 runs daily and is placed in the `/etc/cron.daily` directory. You can use the same technique to
1552 ensure that your scripts remain where you want them.

1553 3.7.7.3 Module: redhatbase

1554 This module configures a baseline for RedHat or CentOS based endpoints. It disables IP
1555 forwarding on endpoints, copies over files including scripts that run periodically, ensures that
1556 the belmonitor asset management software is installed, and configures the logging to the
1557 appropriate logging server.

1558 3.7.7.4 Module: clamav

1559 This module installs clamav anti-virus on Ubuntu endpoints and ensures that the
1560 clamav-daemon service is running.

```
1561 class clamav{  
1562  
1563   package{ 'clamav-daemon':  
1564     ensure=>installed,  
1565   }  
1566  
1567   service{ 'clamav-daemon':  
1568     ensure=>running,  
1569     require=>Package[ 'clamav-daemon' ],  
1570   }  
1571 }
```

1572 3.7.7.5 Module: blacklist

1573 This module removes blacklisted software from endpoints and reports success if the software
 1574 package is removed. Its *init.pp* file is constantly being updated with new software slated for
 1575 removal. A python script called *blacklistenforcer.py* is used to populate the module's
 1576 */etc/puppetlabs/puppet/modules/blacklist/manifests/init.pp* file. Another python script is used
 1577 to read reports from the */var/opt/lib/pe-puppet/reports/<HostFQDN>* subdirectories in order
 1578 to identify successfully removed blacklisted software.

1579 3.7.7.6 Software Blacklist Removal

1580 Puppet Enterprise Server is configured to remove blacklisted software from agent nodes. A
 1581 python script placed in */etc/cron.daily* directory runs daily, checking a blacklisted software. The
 1582 python script will extract the software list from the file */etc/splunkreport/fakeblacklist.csv*,
 1583 write new Puppet code such that Puppet Enterprise catalog includes the blacklisted software,
 1584 and identifies it to Puppet for removal.

1585 3.7.8 Reporting

1586 Puppet agents forward reports of their runs to the Puppet Enterprise server. To ensure
 1587 reporting is enabled, go to */etc/puppetlabs/puppet/puppet.conf* and verify that an entry such
 1588 as `reports = console, puppetdb, store` exists under master section of the file.

1589 Agents upload reports in the form of YAML files to
 1590 */var/opt/lib/pe-puppet/reports/<agent_hostname>*

1591 In this solution, the Puppet Enterprise Server machine was set up to forward two basic reports
 1592 to the ITAM server. Both were done with scripts. The first reporting function forwarded checked
 1593 the fully qualified hostnames of endpoints that failed to upload reports to the server within two
 1594 reporting cycles. If a reporting interval or cycle is 30 minutes, then failure to upload a report for
 1595 more than an hour would result in an endpoint being seen as offline and would trigger the
 1596 forwarding of a syslog message to the ITAM server declaring the endpoint absent. Other
 1597 endpoints that successfully upload reports without missing two cycles are declared present and
 1598 also sending an appropriate message to the ITAM server. The script written that accomplishes
 1599 this is written in BASH and is in the code and scripts section.

1600 The second reporting function reports on the successful removal of blacklisted software. It
 1601 scans through the report files from all the nodes in Puppet Enterprise Server, identifies
 1602 successfully removed software and updates the CSV file */etc/splunkreport/reporttosplunk.csv*
 1603 with information that identifies the endpoint, the successfully removed software and the time
 1604 of removal. The Splunk Universal Forwarder agent monitors this file and forwards changes to
 1605 the ITAM server, which uses Splunk Enterprise as its analysis engine.

1606 3.7.9 Report Directory Cleanup

1607 Thousands of files could be uploaded to the reports directory in a short time. Therefore, it is
 1608 important to delete files that are no longer needed. We used a python script that ran hourly to
 1609 delete files modification times more than 12 hours old. In this solution, that is equivalent to
 1610 files that are more than 12 hours old. This script was placed in the */etc/cron.hourly*.

1611 3.7.10 Puppet Code and Scripts

1612 Main Manifest Configuration File

```
1613 /etc/puppetlabs/puppet/environments/production/manifests/site.pp
1614 ## site.pp ##
1615
1616 # This file (/etc/puppetlabs/puppet/manifests/site.pp) is the main
1617 # entry point used when an agent connects to a master and asks for an #
1618 # updated configuration.
1619 #
1620 # Global objects like filebuckets and resource defaults should go in
1621 # this file, as should the default node definition. (The default node
1622 # can be omitted
1623 # if you use the console and don't define any other nodes in site.pp. #
1624 # See http://docs.puppetlabs.com/guides/language\_guide.html#nodes for #
1625 # more on node definitions.)
1626
1627 ## Active Configurations ##
1628
1629 # PRIMARY FILEBUCKET
1630 # This configures puppet agent and puppet inspect to back up file
1631 # contents when they run. The Puppet Enterprise console needs this to #
1632 # display file contents and differences.
1633
1634 # Define filebucket 'main':
1635 filebucket { 'main':
1636     server => 'puppet.lab5.nccoe.gov',
1637     path   => false,
1638 }
1639
1640 # Make filebucket 'main' the default backup location for all File
1641 # resources:
1642 File { backup => 'main' }
1643
1644 # DEFAULT NODE
1645 # Node definitions in this file are merged with node data from the
1646 # console. See
1647 # http://docs.puppetlabs.com/guides/language\_guide.html#nodes for more
1648 # on node definitions.
1649
1650 # The default node definition matches any node lacking a more specific
1651 # node definition. If there are no other nodes in this file, classes
1652 # declared here will be included in every node's catalog, *in
1653 # addition* to any classes specified in the console for that node.
```

```
1654
1655     node default {
1656         # This is where you can declare classes for all nodes.
1657         # Example:
1658         #   class { 'my_class': }
1659
1660     }
1661     #Changes to the site.pp file were made below this line.
1662     #Nodes were specified with the modules that would execute
1663     #on them
1664     node 'centos1', 'fathomsensor1'{
1665         include redhatbase
1666         include blacklist
1667     }
1668
1669     node 'ubuntu-client1', 'kibana', 'openvas', 'sensu', 'ubuntu-client2',
1670     'wiki'{
1671         include blacklist
1672         include ubuntubase
1673         package{'curl':
1674             ensure => installed,
1675         }
1676     }
1677
1678     node 'ubuntu-template', 'jumpbox', 'bro', 'snort', 'apt-cache',
1679     'warehouse'{
1680         include blacklist
1681         include ubuntubase
1682         package{'curl':
1683             ensure => installed,
1684         }
1685     }
1686
1687     node 'win7-client1', 'win7-client2', 'ad2', 'ad1', 'belarc', 'eracent'{
1688         include blacklist
1689         include windowsnodes
1690     }
1691
1692     node 'asset-manager'{
1693         include blacklist
1694         include windowsnodes
1695     }
```

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,
  }
```

```
}
```

```

1733 /etc/puppetlabs/puppet/modules/windowsnodes/files/baseline.bat
1734
1735 REM Install new user called newuser
1736 net user newuser /add
1737
1738 REM Disable newuser
1739 net user newuser /active:no
1740
1741 REM Disable the guest account
1742 net user guest /active:no
1743
1744 REM Turn on firewall
1745 netsh advfirewall set allprofiles state on
1746
1747 REM Use puppet to check if Malwarebytes is installed
1748 puppet resource package |find "Malwarebytes"
1749
1750 REM Install Malwarebytes silently if not installed
1751 if %errorlevel% neq 0 C:\software\mbamsetup.exe /verysilent /norestart
1752
1753 sc query |find "BelMonitorService"
1754
1755 REM Install Belmonitor if the service is not running
1756 if %errorlevel% neq 0 C:\software\belmonitor.exe

```

ubuntubase Configuration File and Script

```

1758 /etc/puppetlabs/puppet/modules/ubuntubase/manifests/init.pp
1759 #This module configures a baseline for Ubuntu endpoints
1760 class ubuntubase{
1761
1762     #Copy over the CA certificate
1763     file{'/usr/local/share/ca-certificates/CAServerCert.crt':
1764         source => "puppet:///modules/ubuntubase/CAServerCert.crt",
1765     }
1766
1767     # Add CA certificate to Ubuntu endpoint's repository of certificates
1768     exec{'update-ca-certificates':
1769         command=> '/usr/sbin/update-ca-certificates',
1770     }
1771
1772     #Ensure the /etc/ufw directory is present or create it
1773     file{'/etc/ufw':

```

```
1774         ensure=>"directory",
1775     }
1776
1777     #Copy over the sysctl.conf file to each endpoint. IP forwarding will be
1778     #disabled
1779     file{'/etc/ufw/sysctl.conf':
1780         source => "puppet:///modules/ubuntubase/sysctl.conf",
1781         require => File['/etc/ufw'],
1782     }
1783
1784     #Run the clamav module
1785     include clamav
1786
1787     file{'/etc/cron.daily':
1788         ensure=>"directory",
1789     }
1790
1791     file{'/etc/rsyslog.d':
1792         ensure=>"directory",
1793     }
1794
1795     #Copy over this script to endpoint with associated permissions
1796     file{'/etc/cron.daily/dailyscript':
1797         source => "puppet:///modules/ubuntubase/dailyscript",
1798         mode => 754,
1799         require => File['/etc/cron.daily'],
1800     }
1801
1802     #Copy over the 50-default.conf file with specified content
1803     file{'/etc/rsyslog.d/50-default.conf':
1804         content => ".* @loghost\n *.* /var/log/syslog",
1805         require => File['/etc/rsyslog.d'],
1806     }
1807
1808     #Copy over Belmonitor Linux installation file
1809     file{'/opt/BelMonitorLinux':
1810         source => "puppet:///modules/ubuntubase/BelMonitorLinux",
1811     }
1812
1813     #Make the BelMonitorLinux file executable
1814     exec{'belmonitor_executable':
1815         command=>'/bin/chmod a+x /opt/BelMonitorLinux',
1816         require=>File['/opt/BelMonitorLinux'],
```

```

1817     }
1818
1819     exec{'install_rpm':
1820         command=> '/usr/bin/apt-get install -y rpm',
1821         require=> File['/opt/BelMonitorLinux']
1822     }
1823
1824
1825     ##Install 32 bit library
1826     exec{'install_32bitlibrary':
1827         command=> '/usr/bin/apt-get install -y gcc-multilib',
1828         require=> Exec['install_rpm'],
1829     }
1830
1831     ##install 32 bit library
1832     exec{'install_second_32bit_library':
1833         command=> '/usr/bin/apt-get install -y lib32stdc++6',
1834     }
1835
1836     exec{'install_belmonitor':
1837         command=> '/opt/BelMonitorLinux',
1838         require=> Exec['install_32bitlibrary'],
1839     }
1840
1841     service{'BelMonitor':
1842         ensure=> 'running',
1843     }
1844 }

```

```

1845 /etc/puppetlabs/puppet/modules/ubuntubase/files/dailyscript
1846 #!/bin/bash
1847 df -kh
1848 mount
1849 w
1850 netstat -nult
1851 ifconfig -a
1852 iptables -L
1853 /usr/bin/freshclam
1854 cat /var/lib/apt/extended_states
1855 apt-get update

```

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
```

```
1898
1899     file{'/opt':
1900         ensure=>"directory",
1901         }
1902     file{'/opt/BelMonitorLinux':
1903         source => "puppet:///modules/redhatbase/BelMonitorLinux",
1904         }
1905
1906     ##Make BelMonitorLinux executable
1907     exec{'make_executable':
1908         command=>'/bin/chmod a+x /opt/BelMonitorLinux',
1909         require => File['/opt/BelMonitorLinux'],
1910         }
1911
1912     ##Install dependencies
1913     exec{'upgrade_dep1':
1914         command=>'/usr/bin/yum -y upgrade libstdc++',
1915         }
1916     }
1917
1918     exec{'install_dep2':
1919         command=>'/usr/bin/yum -y install libstdc++.i686',
1920         }
1921
1922     exec{'upgrade_dep3':
1923         command=>'/usr/bin/yum -y upgrade zlib',
1924         }
1925
1926     exec{'install_dep4':
1927         command=>'/usr/bin/yum -y install zlib.i686',
1928         }
1929
1930     exec{'install_belmonitor':
1931         command=>'/opt/BelMonitorLinux',
1932         }
1933
1934     file{'/opt/belmonitor_install':
1935         source => "puppet:///modules/redhatbase/belmonitor_install",
1936         }
1937
1938 }
```

```
/etc/puppetlabs/puppet/modules/redhatbase/files/clamininstall
```

```
#!/bin/bash
# /etc/puppetlabs/puppet/modules/redhatbase/files/clamininstall#
# 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
```

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'],
    }
}
```

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
```

```

1979     #List directories in /var/opt/lib/pe-puppet/reports
1980     report_list = os.listdir('/var/opt/lib/pe-puppet/reports')
1981     #Make the path to reports a string
1982     origdir_path = '/var/opt/lib/pe-puppet/reports'
1983
1984     action_term = "file:
1985     /etc/puppetlabs/puppet/modules/blacklist/manifests/init.pp"
1986     outfile = open('/etc/splunkreport/reporttosplunk.csv', 'a')
1987     #For loop iterates through report_list (or the reports directory)
1988     for sub_dirs in report_list:
1989         hostname = sub_dirs
1990         print(hostname)
1991         #Concatenation creates the full path to subdirectories (it remains
1992         a string)
1993         subdir_path = origdir_path+'/'+sub_dirs
1994         #print(subdir_path)
1995         #Creates the list of files in the variable (the variable in this
1996         case would be a sub directory)
1997         #At the end of this block, infile contains a list of line elements
1998         in each file
1999         sub_dirs_list = os.listdir(subdir_path)
2000         for files in sub_dirs_list:
2001             files_path = subdir_path+'/'+files
2002             reportfile = open(files_path, "r")
2003             infile = reportfile.readlines()
2004             reportfile.close()
2005             #line_counter used in keeping track of the index for the line
2006             elements in each file
2007             line_counter = 0
2008
2009             for line in infile:
2010                 if action_term in line:
2011                     if "source" in infile[line_counter + 3]:
2012                         bad_package = infile[line_counter + 3]
2013                         #print(bad_package)
2014                         bad_package = bad_package.replace('\n',',')
2015                         #print(infile[line_counter + 2])
2016                         if "removed" in infile[line_counter + 2]:
2017                             message_var = infile[line_counter + 2]
2018                             message_var = message_var.replace('\n',',')
2019                             if "time" in infile[line_counter + 1]:
2020                                 time_var = infile[line_counter + 1]
2021                                 time_var = time_var.replace('\n',',')
2022                                 refined_bad_pkg = bad_package.split('/')

```

```
2023             bad_pkg = refined_bad_pkg[3]
2024             bad_pkg = bad_pkg + ","
2025
2026         print(hostname+", "+bad_pkg+message_var+time_var+'\n')
2027
2028         outfile.write(hostname+', '+bad_pkg+message_var+time_var+'\n')
2029             line_counter = line_counter + 1
```

2030 **Reports Directory Cleanup Script**

```
2031 /etc/cron.hourly/cleanreportdir.py
2032 #!/usr/bin/python3
2033
2034 #-----cleanreportdir.py-----#
2035 #Script removes files with mtimes older than 12 hours to keep the
2036 number of files to a manageable size
2037 #Files removed are from the reports subdirectory within Puppet
2038 import os
2039 import time
2040 #List directories in /var/opt/lib/pe-puppet/reports
2041 report_list = os.listdir('/var/opt/lib/pe-puppet/reports')
2042 #Make the path to reports a string
2043 origdir_path = '/var/opt/lib/pe-puppet/reports'
2044 #For loop iterates through report_list
2045 for sub_dirs in report_list:
2046     #Concatenation creates the full path to subdirectories (it remains
2047     a string)
2048     subdir_path = origdir_path+'/'+sub_dirs
2049     print('Old files are being removed from ',subdir_path)
2050     #Creates the list of files in the variable sub_dirs_list
2051     sub_dirs_list = os.listdir(subdir_path)
2052     for files in sub_dirs_list:
2053         files_path = subdir_path+'/'+files
2054         mtime = os.path.getmtime(files_path)
2055         current_time = time.time()
2056         time_diff = current_time - mtime
2057         #Removes files with mtimes older than 12 hours
2058         if time_diff > 43200:
2059             print(files_path, " will be deleted")
2060             os.remove(files_path)
```

2061 **Reporting Section Script**

```
2062 #!/bin/bash
2063 #/etc/cron.hourly/nodereport
```

```

2064     #Time in seconds before declaring an agent that has not checked in
2065     absent
2066     # Change the time to suit your needs
2067     let "desired_interval=3600"
2068
2069     for node in $(ls /var/opt/lib/pe-puppet/yaml/node)
2070     do
2071         #Strip out the yaml extension from the node name
2072         node=${node%.*}
2073         #Get time of most recent agent run or check in
2074         #This time will be reported without formatting
2075         node_report_time=$(date -r
2076 /var/opt/lib/pe-puppet/yaml/facts/$node.yaml)
2077
2078         #Get epoch time of agent facter yaml file, assign time to variable
2079         node_time=$(date +%s -r
2080 /var/opt/lib/pe-puppet/yaml/facts/$node.yaml)
2081
2082         #Assign current epoch_time to variable
2083         current_time=$(date +%s)
2084
2085         #Subtract node most recent report time from current time and
2086         #assign to variable
2087         node_interval=$((current_time-node_time))
2088
2089         #Nodes with that have not reported in the given interval are
2090         #declared absent, otherwise they are declared present
2091         if ((" $node_interval" > " $desired_interval"))
2092         then
2093             echo $node "is absent with a last run time of " $node_report_time
2094             logger $node "is absent. Last run is " $node_report_time
2095
2096         else
2097             echo $node "is present with a last run time of "
2098 $node_report_time
2099             logger $node "is present. Last run is " $node_report_time
2100         fi
2101     done

```

2102 3.8 Snort

2103 Snort is an open-source intrusion detection system. Snort efficiently analyzes all network traffic
2104 and matches it with signatures of know bad traffic. An alert is generated if a signature is
2105 matched.

2106 3.8.1 How It's Used

- 2107 In the FS ITAM build, Snort monitors all traffic traversing the DMZ.
- 2108 On the high-level architecture diagram, Snort is in Tier 2. Snort utilizes the Splunk Universal
- 2109 Forwarder to send alerts to Splunk Enterprise.

2110 3.8.2 Virtual Machine Configuration

- 2111 The Snort virtual machine is configured with one network interface card, 2 GB of RAM and one
- 2112 CPU core.

2113 3.8.3 Network Configuration

- 2114 The management network interface card is configured as follows:
- 2115 IPv4 Manual
- 2116 IPv6 Ignore/Disabled
- 2117 IP Address: 172.16.0.40
- 2118 Netmask: 255.255.255.0
- 2119 Gateway: 172.16.0.11
- 2120 DNS Servers: 172.16.1.20, 172.16.1.21
- 2121 Search Domains: lab5.nccoe.gov

2122 3.8.4 Installing Snort

- 2123 Snort is installed on a hardened Ubuntu 14.04 Linux system. Complete installation instructions
- 2124 can be found at: <https://www.snort.org/>.
- 2125 This installation utilized the Snort IDS and Barnyard2 to interpret binary Snort alerts into
- 2126 readable text.

2127 3.8.5 Installing Snort

- 2128 For Debian/Ubuntu Linux systems, it is always best to make sure you system is up-to-date by
- 2129 performing:
- 2130 **`sudo apt-get update`**
- 2131 **`sudo apt-get upgrade`**
- 2132 **`sudo apt-get install snort`**
- 2133 You will be asked to input your local networks. For the FS-ITAM lab this is **172.16.0.0/16**.
- 2134 Configure */etc/snort/snort.debian.conf*.

2135 Make sure that the correct HOME_NET and INTERFACE are specified in
 2136 */etc/snort/snort.debian.conf*.

2137 **DEBIAN_SNORT_HOME_NET="172.16.0.0/16"**

2138 **DEBIAN_SNORT_INTERFACE="eth0"**

2139 Configure */etc/snort/snort.conf*.

2140 Comment out all output configuration lines and add the following:

2141 **output unified2: filename /var/log/snort/snort.log, limit 128, mpls_event_types,**
 2142 **vlan_event_types**

2143 The preceding line is important for Barnyard2 to work correctly.

2144 3.8.6 Get Updated Community Rules

2145 `cd /opt`
 2146 `wget https://snort.org/downloads/community/community-rules.tar.gz`
 2147 `tar xzvf community.rules.tar.gz -C /etc/snort/rules`

2148 These community rules contain the **sid-msg.map** file that Barnyard2 needs.

2149 `mkdir /etc/snort/etc`
 2150 `cp /etc/snort/rules/community-rules/sid-msg.map /etc/snort/etc`

2151 **Note:** In a production environment, it is advisable to install an automatic rule updater such as
 2152 PulledPork. PulledPork requires obtaining an account at Snort.org which results in an Oinkcode.

2153 3.8.7 Installing Barnyard2

2154 Install the prerequisites:

2155 `sudo apt-get install build-essential libtool autoconf git nmap`
 2156 `sudo apt-get install libpcap-dev libmysqld-dev libpcrc3-dev`
 2157 `libdumbnet-dev`
 2158 `sudo apt-get install flex bison`
 2159 `ldconfig`

2160 Barnyard2 requires the `<dnet.h>` header. Unfortunately, Ubuntu names this header
 2161 `<dumbnet.h>` so we must create a symbolic link for Barnyard2 to compile.

2162 `cd /usr/include`
 2163 `ln -s /usr/include/dumbnet.h dnet.h`

2164 **Note:** You need to be root to install Barnyard2

2165 `cd /opt`
 2166 Need the Daq libraries from Snort
 2167 `wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz`
 2168 `tar xzvf daq-2.0.6.tar.gz`

```
2169 cd /opt/daq-2.0.6
2170 ./configure
2171 make
2172 make install
2173 git clone https://github.com/firnsy/barnyard2.git
2174 cd /opt/barnyard2
2175 ./autogen.sh
2176 ./configure
2177 make
2178 make install

2179 Copy the provided barnyard2.conf file to /usr/local/etc.
2180 cp /usr/local/etc/barnyard2.conf /usr/local/etc/barnyard2.conf.orig
2181 cp <barnyard2.conf> /usr/local/etc

2182 Create a link inside /etc/snort to this file
2183 ln -s /usr/local/etc/barnyard2 /etc/snort/barnyard.conf

2184 Copy the provided barnyard2 init script to /etc/init.d and make it executable
2185 cp <barnyard2> /etc/init.d
2186 chmod 755 /etc/init.d/barnyard2
2187 sudo update-rc.d barnyard2 defaults
2188 sudo update-rc.d barnyard2 enable

2189 Start up Barnyard2
2190 /etc/init.d/barnyard2 start

2191 Error messages can be found in /var/log/syslog.
```

2192 3.8.8 Testing

```
2193 Performing these steps will let you know that Snort and Barnyard2 are working.
2194
2195 1. Add a local rule.
2196
2197 2. Edit /etc/snort/rules/local.rules by adding the following line at the bottom that will
2198 generate alerts for any ICMP/Ping traffic.
2199
2200 alert icmp any any -> any any (msg: "ICMP Detected"; classtype:unknown; sid:1000001;
2201 rev:1;)
2202
2203 Note: the sid must be greater than 1 million.

2204 3. Restart Snort.
2205
2206 service snort restart
2207
2208 4. Verify that Snort is running.
2209
2210 ps -ef |grep snort
```

- 2204 5. Verify that Barnyard2 is running.
- 2205 `ps -ef |grep barnyard2`
- 2206 6. Check the logs in `/var/log/snort`. The `snort.log` and `alert` files should both be growing fast.
- 2207 7. You can view the alert file.
- 2208 `tail -f /var/log/snort/alert`
- 2209 **Note:** Do not leave this test running. If you do, it will fill your hard drive.
- 2210 8. If everything is good just comment out the line that you created in `local.rules` and restart
- 2211 `Snort`.

2212 3.8.9 Installing Splunk Universal Forwarder

2213 **Note:** You will need a Splunk account to download the Splunk Universal Forwarder. It is free and

2214 can be set up at:

2215 https://www.splunk.com/page/sign_up

2216 Download the Splunk Universal Forwarder from:

2217 http://www.splunk.com/en_us/download/universal-forwarder.html

2218 You want the latest version for OS version 2.6+ kernel Linux distributions

2219 (64-bit). Since this is installing on Ubuntu, select the file that ends in `.deb`. An example is:

2220 `splunkforwarder-6.2.5-272645-linux-2.6-amd64.deb`

2221 Detailed installation instructions can be found at:

2222 [http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_i](http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_install)

2223 [n](http://docs.splunk.com/Documentation/Splunk/6.2.4/Installation/InstallonLinuxDebian_DEB_install)`stall`

2224 An abridged version follows:

2225 `dpkg -i <splunk_package_name.deb>`

2226 Example: `dpkg -i splunkforwarder-6.2.5-272645-linux-2.6-amd64.deb`

2227 This will install in `/opt/splunkforwarder`:

2228 `cd /opt/splunkforwarder/bin`

2229 `./splunk start --accept-license`

2230 `./splunk enable boot-start`

2231 Add forwarder:

2232 More information about adding a forwarder can be found at:

2233 <http://docs.splunk.com/Documentation/Splunk/6.2.4/Forwarding/Deployanixdfmanually>

2234 `cd /opt/splunkforwarder/bin`

2235 `./splunk add forward-server loghost:9997 -auth admin:changme`

2236 3.8.10 Configuring Splunk Universal Forwarder

2237 Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509
2238 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You
2239 will also need a copy of your certificate authority's public certificate.

2240 Create a directory to hold your certificates:

2241 `mkdir /opt/splunkforwarder/etc/certs`

2242 Copy your certificates in PEM format to `/opt/splunkforwarder/etc/certs`:

2243 `cp CAsServerCert.pem /opt/splunkforwarder/etc/certs`

2244 `cp bro_worker1.pem /opt/splunkforwarder/etc/certs`

2245 Copy Splunk Universal Forwarder configuration files:

2246 `cp <server.conf> /opt/splunkforwarder/etc/system/local`

2247 `cp <inputs.conf> /opt/splunkforwarder/etc/system/local`

2248 `cp <outputs.conf> /opt/splunkforwarder/etc/system/local`

2249 Modify **server.conf** so that:

- 2250 • **ServerName=snort** is your hostname.
- 2251 • **sslKeysfilePassword** = <password for your private key>

2252 Modify **outputs.conf** so that:

- 2253 • **Server = loghost:9997** is your correct Splunk Enterprise server/indexer and port.
- 2254 • **sslPassword** = <password of your certificate private key>

2255 **Note:** This will be hashed and not clear text after a restart.

2256 **Inputs.conf** should work, but you are free to modify it to include the Bro logs that you are
2257 interested in.

2258 3.8.11 Configurations and Scripts

2259 `/etc/default/barnyard2`

2260 `# Config file for /etc/init.d/barnyard2`

2261 `#LOG_FILE="snort_unified.log"`

2262 `LOG_FILE="snort.log"`

2263 `# You probably don't want to change this, but in case you do`

2264 `SNORTDIR="/var/log/snort"`

2265 `INTERFACES="eth0"`

2266 `# Probably not this either`

2267 `CONF=/etc/snort/barnyard2.conf`

2268 `EXTRA_ARGS="`

```

2269 /etc/snort/snort.conf
2270 #-----
2271 #   VRT Rule Packages Snort.conf
2272 #
2273 #   For more information visit us at:
2274 #       http://www.snort.org                Snort Website
2275 #       http://vrt-blog.snort.org/          Sourcefire VRT Blog
2276 #
2277 #       Mailing list Contact:      snort-sigs@lists.sourceforge.net
2278 #       False Positive reports:    fp@sourcefire.com
2279 #       Snort bugs:                bugs@snort.org
2280 #
2281 #       Compatible with Snort Versions:
2282 #       VERSIONS : 2.9.6.0
2283 #
2284 #       Snort build options:
2285 #       OPTIONS : --enable-gre --enable-mpls --enable-targetbased
2286 #       --enable-ppm --enable-perfprofiling --enable-zlib
2287 #       --enable-active-response --enable-normalizer --enable-reload
2288 #       --enable-react --enable-flexresp3
2289 #
2290 #       Additional information:
2291 #       This configuration file enables active response, to run snort in
2292 #       test mode -T you are required to supply an interface -i
2293 #       <interface>
2294 #       or test mode will fail to fully validate the configuration and
2295 #       exit with a FATAL error
2296 #-----
2297 #####
2298 # This file contains a sample snort configuration.
2299 # You should take the following steps to create your own custom
2300 # configuration:
2301 #
2302 #   1) Set the network variables.
2303 #   2) Configure the decoder
2304 #   3) Configure the base detection engine
2305 #   4) Configure dynamic loaded libraries
2306 #   5) Configure preprocessors
2307 #   6) Configure output plugins
2308 #   7) Customize your rule set
2309 #   8) Customize preprocessor and decoder rule set
2310 #   9) Customize shared object rule set
2311 #####

```

```
2312 #####
2313 # Step #1: Set the network variables. For more information, see
2314 README.variables
2315 #####

2316 # Setup the network addresses you are protecting
2317 #
2318 # Note to Debian users: this value is overridden when starting
2319 # up the Snort daemon through the init.d script by the
2320 # value of DEBIAN_SNORT_HOME_NET s defined in the
2321 # /etc/snort/snort.debian.conf configuration file
2322 #
2323 ipvar HOME_NET any

2324 # Set up the external network addresses. Leave as "any" in most
2325 situations
2326 ipvar EXTERNAL_NET any
2327 # If HOME_NET is defined as something other than "any", alternative,
2328 you can
2329 # use this definition if you do not want to detect attacks from your
2330 internal
2331 # IP addresses:
2332 #ipvar EXTERNAL_NET !$HOME_NET

2333 # List of DNS servers on your network
2334 ipvar DNS_SERVERS $HOME_NET

2335 # List of SMTP servers on your network
2336 ipvar SMTP_SERVERS $HOME_NET

2337 # List of web servers on your network
2338 ipvar HTTP_SERVERS $HOME_NET

2339 # List of sql servers on your network
2340 ipvar SQL_SERVERS $HOME_NET

2341 # List of telnet servers on your network
2342 ipvar TELNET_SERVERS $HOME_NET

2343 # List of ssh servers on your network
2344 ipvar SSH_SERVERS $HOME_NET

2345 # List of ftp servers on your network
2346 ipvar FTP_SERVERS $HOME_NET

2347 # List of sip servers on your network
2348 ipvar SIP_SERVERS $HOME_NET
```

```

2349      # List of ports you run web servers on
2350      portvar HTTP_PORTS
2351      [36,80,81,82,83,84,85,86,87,88,89,90,311,383,555,591,593,631,801,808,8
2352      18,901,972,1158,1220,1414,1533,1741,1830,2231,2301,2381,2809,3029,3037
2353      ,3057,3128,3443,3702,4000,4343,4848,5117,5250,6080,6173,6988,7000,7001
2354      ,7144,7145,7510,7770,7777,7779,8000,8008,8014,8028,8080,8081,8082,8085
2355      ,8088,8090,8118,8123,8180,8181,8222,8243,8280,8300,8500,8509,8800,8888
2356      ,8899,9000,9060,9080,9090,9091,9111,9443,9999,10000,11371,12601,15489,
2357      29991,33300,34412,34443,34444,41080,44449,50000,50002,51423,53331,5525
2358      2,55555,56712]

2359      # List of ports you want to look for SHELLCODE on.
2360      portvar SHELLCODE_PORTS !80

2361      # List of ports you might see oracle attacks on
2362      portvar ORACLE_PORTS 1024:

2363      # List of ports you want to look for SSH connections on:
2364      portvar SSH_PORTS 22

2365      # List of ports you run ftp servers on
2366      portvar FTP_PORTS [21,2100,3535]

2367      # List of ports you run SIP servers on
2368      portvar SIP_PORTS [5060,5061,5600]

2369      # List of file data ports for file inspection
2370      portvar FILE_DATA_PORTS [$HTTP_PORTS,110,143]

2371      # List of GTP ports for GTP preprocessor
2372      portvar GTP_PORTS [2123,2152,3386]

2373      # other variables, these should not be modified
2374      ipvar AIM_SERVERS
2375      [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0
2376      /24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.18
2377      8.153.0/24,205.188.179.0/24,205.188.248.0/24]

2378      # Path to your rules files (this can be a relative path)
2379      # Note for Windows users: You are advised to make this an absolute
2380      path,
2381      # such as: c:\snort\rules
2382      #var RULE_PATH /etc/snort/rules
2383      var RULE_PATH rules
2384      var SO_RULE_PATH /etc/snort/so_rules
2385      var PREPROC_RULE_PATH /etc/snort/preproc_rules

2386      # If you are using reputation preprocessor set these
2387      # Currently there is a bug with relative paths, they are relative to
2388      where snort is

```

```
2389      # not relative to snort.conf like the above variables
2390      # This is completely inconsistent with how other vars work, BUG 89986
2391      # Set the absolute path appropriately
2392      var WHITE_LIST_PATH /etc/snort/rules
2393      var BLACK_LIST_PATH /etc/snort/rules

2394      #####
2395      # Step #2: Configure the decoder. For more information, see
2396      README.decode
2397      #####

2398      # Stop generic decode events:
2399      config disable_decode_alerts

2400      # Stop Alerts on experimental TCP options
2401      config disable_tcpopt_experimental_alerts

2402      # Stop Alerts on obsolete TCP options
2403      config disable_tcpopt_obsolete_alerts

2404      # Stop Alerts on T/TCP alerts
2405      config disable_tcpopt_ttcp_alerts

2406      # Stop Alerts on all other TCPOption type events:
2407      config disable_tcpopt_alerts

2408      # Stop Alerts on invalid ip options
2409      config disable_ipopt_alerts

2410      # Alert if value in length field (IP, TCP, UDP) is greater th elength
2411      of the packet
2412      # config enable_decode_oversized_alerts

2413      # Same as above, but drop packet if in Inline mode (requires
2414      enable_decode_oversized_alerts)
2415      # config enable_decode_oversized_drops

2416      # Configure IP / TCP checksum mode
2417      config checksum_mode: all

2418      # Configure maximum number of flowbit references. For more information,
2419      see README.flowbits
2420      # config flowbits_size: 64

2421      # Configure ports to ignore
2422      # config ignore_ports: tcp 21 6667:6671 1356
2423      # config ignore_ports: udp 1:17 53
```

```

2424     # Configure active response for non inline operation. For more
2425     information, see README.active
2426     # config response: eth0 attempts 2
2427     # Configure DAQ related options for inline operation. For more
2428     information, see README.daq
2429     #
2430     # config daq: <type>
2431     # config daq_dir: <dir>
2432     # config daq_mode: <mode>
2433     # config daq_var: <var>
2434     #
2435     # <type> ::= pcap | afpacket | dump | nfq | ipq | ipfw
2436     # <mode> ::= read-file | passive | inline
2437     # <var> ::= arbitrary <name>=<value passed to DAQ
2438     # <dir> ::= path as to where to look for DAQ module so's

2439     # Configure specific UID and GID to run snort as after dropping privs.
2440     For more information see snort -h command line options
2441     #
2442     # config set_gid:
2443     # config set_uid:

2444     # Configure default snaplen. Snort defaults to MTU of in use interface.
2445     For more information see README
2446     #
2447     # config snaplen:
2448     #

2449     # Configure default bpf_file to use for filtering what traffic reaches
2450     snort. For more information see snort -h command line options (-F)
2451     #
2452     # config bpf_file:
2453     #

2454     # Configure default log directory for snort to log to. For more
2455     information see snort -h command line options (-l)
2456     #
2457     # config logdir:

2458     #####
2459     # Step #3: Configure the base detection engine. For more information,
2460     see README.decode
2461     #####
2462
2463     # Configure PCRE match limitations
2464     config pcre_match_limit: 3500

```

```
2465         config pcre_match_limit_recursion: 1500
2466
2467         # Configure the detection engine  See the Snort Manual, Configuring
2468         Snort - Includes - Config
2469         config detection: search-method ac-split search-optimize
2470         max-pattern-len 20
2471
2472         # Configure the event queue.  For more information, see
2473         README.event_queue
2474         config event_queue: max_queue 8 log 5 order_events content_length
2475
2476         #####
2477         ## Configure GTP if it is to be used.
2478         ## For more information, see README.GTP
2479         #####
2480
2481         # config enable_gtp
2482
2483         #####
2484         # Per packet and rule latency enforcement
2485         # For more information see README.ppm
2486         #####
2487
2488         # Per Packet latency configuration
2489         #config ppm: max-pkt-time 250, \
2490         #   fastpath-expensive-packets, \
2491         #   pkt-log
2492
2493         # Per Rule latency configuration
2494         #config ppm: max-rule-time 200, \
2495         #   threshold 3, \
2496         #   suspend-expensive-rules, \
2497         #   suspend-timeout 20, \
2498         #   rule-log alert
2499
2500         #####
2501         # Configure Perf Profiling for debugging
2502         # For more information see README.PerfProfiling
2503         #####
2504
2505         #config profile_rules: print all, sort avg_ticks
2506         #config profile_preprocs: print all, sort avg_ticks
2507
2508         #####
```

```

2509     # Configure protocol aware flushing
2510     # For more information see README.stream5
2511     #####
2512     config paf_max: 16000
2513
2514     #####
2515     # Step #4: Configure dynamic loaded libraries.
2516     # For more information, see Snort Manual, Configuring Snort - Dynamic
2517     Modules
2518     #####
2519
2520     # path to dynamic preprocessor libraries
2521     dynamicpreprocessor directory /usr/lib/snort_dynamicpreprocessor/
2522
2523     # path to base preprocessor engine
2524     dynamicengine /usr/lib/snort_dynamicengine/libsf_engine.so
2525
2526     # path to dynamic rules libraries
2527     dynamicdetection directory /usr/lib/snort_dynamicrules
2528
2529     #####
2530     # Step #5: Configure preprocessors
2531     # For more information, see the Snort Manual, Configuring Snort -
2532     Preprocessors
2533     #####
2534
2535     # GTP Control Channle Preprocessor. For more information, see
2536     README.GTP
2537     # preprocessor gtp: ports { 2123 3386 2152 }
2538
2539     # Inline packet normalization. For more information, see
2540     README.normalize
2541     # Does nothing in IDS mode
2542     preprocessor normalize_ip4
2543     preprocessor normalize_tcp: ips ecn stream
2544     preprocessor normalize_icmp4
2545     preprocessor normalize_ip6
2546     preprocessor normalize_icmp6
2547
2548     # Target-based IP defragmentation. For more information, see
2549     README.frag3
2550     preprocessor frag3_global: max_fragments 65536
2551     preprocessor frag3_engine: policy windows detect_anomalies
2552     overlap_limit 10 min_fragment_length 100 timeout 180

```

```
2553
2554     # Target-Based stateful inspection/stream reassembly.  For more
2555     inforation, see README.stream5
2556     preprocessor stream5_global: track_tcp yes, \
2557         track_udp yes, \
2558         track_icmp no, \
2559         max_tcp 262144, \
2560         max_udp 131072, \
2561         max_active_responses 2, \
2562         min_response_seconds 5
2563     preprocessor stream5_tcp: policy windows, detect_anomalies,
2564     require_3whs 180, \
2565         overlap_limit 10, small_segments 3 bytes 150, timeout 180, \
2566         ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137
2567         139 143 \
2568             161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665
2569         6666 6667 6668 6669 \
2570             7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778
2571         32779, \
2572         ports both 36 80 81 82 83 84 85 86 87 88 89 90 110 311 383 443 465
2573         563 555 591 593 631 636 801 808 818 901 972 989 992 993 994 995 1158
2574         1220 1414 1533 1741 1830 2231 2301 2381 2809 3029 3037 3057 3128 3443
2575         3702 4000 4343 4848 5117 5250 6080 6173 6988 7907 7000 7001 7144 7145
2576         7510 7802 7770 7777 7779 \
2577             7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912
2578         7913 7914 7915 7916 \
2579             7917 7918 7919 7920 8000 8008 8014 8028 8080 8081 8082 8085 8088
2580         8090 8118 8123 8180 8181 8222 8243 8280 8300 8500 8509 8800 8888 8899
2581         9000 9060 9080 9090 9091 9111 9443 9999 10000 11371 12601 15489 29991
2582         33300 34412 34443 34444 41080 44449 50000 50002 51423 53331 55252 55555
2583         56712
2584     preprocessor stream5_udp: timeout 180
2585
2586     # performance statistics.  For more information, see the Snort Manual,
2587     Configuring Snort - Preprocessors - Performance Monitor
2588     # preprocessor perfmonitor: time 300 file /var/snort/snort.stats pktcnt
2589     10000
2590
2591     # HTTP normalization and anomaly detection.  For more information, see
2592     README.http_inspect
2593     preprocessor http_inspect: global iis_unicode_map unicode.map 1252
2594     compress_depth 65535 decompress_depth 65535 max_gzip_mem 104857600
2595     preprocessor http_inspect_server: server default \
2596         http_methods { GET POST PUT SEARCH MKCOL COPY MOVE LOCK UNLOCK
2597         NOTIFY POLL BCOPY BDELETE BMOVE LINK UNLINK OPTIONS HEAD DELETE TRACE
2598         TRACK CONNECT SOURCE SUBSCRIBE UNSUBSCRIBE PROPFIND PROPPATCH BPROPFIND
```

```

2599     BPROPPATCH RPC_CONNECT PROXY_SUCCESS BITS_POST CCM_POST SMS_POST
2600     RPC_IN_DATA RPC_OUT_DATA RPC_ECHO_DATA } \
2601         chunk_length 500000 \
2602         server_flow_depth 0 \
2603         client_flow_depth 0 \
2604         post_depth 65495 \
2605         oversize_dir_length 500 \
2606         max_header_length 750 \
2607         max_headers 100 \
2608         max_spaces 200 \
2609         small_chunk_length { 10 5 } \
2610         ports { 36 80 81 82 83 84 85 86 87 88 89 90 311 383 555 591 593 631
2611         801 808 818 901 972 1158 1220 1414 1741 1830 2231 2301 2381 2809 3029
2612         3037 3057 3128 3443 3702 4000 4343 4848 5117 5250 6080 6173 6988 7000
2613         7001 7144 7145 7510 7770 7777 7779 8000 8008 8014 8028 8080 8081 8082
2614         8085 8088 8090 8118 8123 8180 8181 8222 8243 8280 8300 8500 8509 8800
2615         8888 8899 9000 9060 9080 9090 9091 9111 9443 9999 10000 11371 12601
2616         15489 29991 33300 34412 34443 34444 41080 44449 50000 50002 51423 53331
2617         55252 55555 56712 } \
2618         non_rfc_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \
2619         enable_cookie \
2620         extended_response_inspection \
2621         inspect_gzip \
2622         normalize_utf \
2623         unlimited_decompress \
2624         normalize_javascript \
2625         apache_whitespace no \
2626         ascii no \
2627         bare_byte no \
2628         directory no \
2629         double_decode no \
2630         iis_backslash no \
2631         iis_delimiter no \
2632         iis_unicode no \
2633         multi_slash no \
2634         utf_8 no \
2635         u_encode yes \
2636         webroot no
2637
2638     # ONC-RPC normalization and anomaly detection.  For more information,
2639     see the Snort Manual, Configuring Snort - Preprocessors - RPC Decode
2640     preprocessor rpc_decode: 111 32770 32771 32772 32773 32774 32775 32776
2641     32777 32778 32779 no_alert_multiple_requests no_alert_large_fragments
2642     no_alert_incomplete
2643

```

```
2644     # Back Orifice detection.
2645     preprocessor bo
2646
2647     # FTP / Telnet normalization and anomaly detection.  For more
2648     information, see README.ftptelnet
2649     preprocessor ftp_telnet: global inspection_type stateful
2650     encrypted_traffic no check_encrypted
2651     preprocessor ftp_telnet_protocol: telnet \
2652         ayt_attack_thresh 20 \
2653         normalize ports { 23 } \
2654         detect_anomalies
2655     preprocessor ftp_telnet_protocol: ftp server default \
2656         def_max_param_len 100 \
2657         ports { 21 2100 3535 } \
2658         telnet_cmds yes \
2659         ignore_telnet_erase_cmds yes \
2660         ftp_cmds { ABOR ACCT ADAT ALLO APPE AUTH CCC CDUP } \
2661         ftp_cmds { CEL CLNT CMD CONF CWD DELE ENC EPRT } \
2662         ftp_cmds { EPSV ESTA ESTP FEAT HELP LANG LIST LPRT } \
2663         ftp_cmds { LPSV MACB MAIL MDTM MIC MKD MLSD MLST } \
2664         ftp_cmds { MODE NLST NOOP OPTS PASS PASV PBSZ PORT } \
2665         ftp_cmds { PROT PWD QUIT REIN REST RETR RMD RNFR } \
2666         ftp_cmds { RNTD SDUP SITE SIZE SMNT STAT STOR STOU } \
2667         ftp_cmds { STRU SYST TEST TYPE USER XCUP XCRC XCWD } \
2668         ftp_cmds { XMAS XMD5 XMKD XPWD XRCF XRMD XRSQ XSEM } \
2669         ftp_cmds { XSEN XSHA1 XSHA256 } \
2670         alt_max_param_len 0 { ABOR CCC CDUP ESTA FEAT LPSV NOOP PASV PWD
2671         QUIT REIN STOU SYST XCUP XPWD } \
2672         alt_max_param_len 200 { ALLO APPE CMD HELP NLST RETR RNFR STOR STOU
2673         XMKD } \
2674         alt_max_param_len 256 { CWD RNTD } \
2675         alt_max_param_len 400 { PORT } \
2676         alt_max_param_len 512 { SIZE } \
2677         chk_str_fmt { ACCT ADAT ALLO APPE AUTH CEL CLNT CMD } \
2678         chk_str_fmt { CONF CWD DELE ENC EPRT EPSV ESTP HELP } \
2679         chk_str_fmt { LANG LIST LPRT MACB MAIL MDTM MIC MKD } \
2680         chk_str_fmt { MLSD MLST MODE NLST OPTS PASS PBSZ PORT } \
2681         chk_str_fmt { PROT REST RETR RMD RNFR RNTD SDUP SITE } \
2682         chk_str_fmt { SIZE SMNT STAT STOR STRU TEST TYPE USER } \
2683         chk_str_fmt { XCRC XCWD XMAS XMD5 XMKD XRCF XRMD XRSQ } \
2684         chk_str_fmt { XSEM XSEN XSHA1 XSHA256 } \
2685         cmd_validity ALLO < int [ char R int ] > \
2686         cmd_validity EPSV < [ { char 12 | char A char L char L } ] > \
2687         cmd_validity MACB < string > \
```

```

2688     cmd_validity MDTM < [ date nnnnnnnnnnnnnnn[n[n[n]]] ] string > \
2689     cmd_validity MODE < char ASBCZ > \
2690     cmd_validity PORT < host_port > \
2691     cmd_validity PROT < char CSEP > \
2692     cmd_validity STRU < char FRPO [ string ] > \
2693     cmd_validity TYPE < { char AE [ char NTC ] | char I | char L [
2694 number ] } >
2695     preprocessor ftp_telnet_protocol: ftp client default \
2696         max_resp_len 256 \
2697         bounce yes \
2698         ignore_telnet_erase_cmds yes \
2699         telnet_cmds yes

2700     # SMTP normalization and anomaly detection.  For more information, see
2701     README.SMTP
2702     preprocessor smtp: ports { 25 465 587 691 } \
2703         inspection_type stateful \
2704         b64_decode_depth 0 \
2705         qp_decode_depth 0 \
2706         bitenc_decode_depth 0 \
2707         uu_decode_depth 0 \
2708         log_mailfrom \
2709         log_rcptto \
2710         log_filename \
2711         log_email_hdrs \
2712         normalize_cmds \
2713         normalize_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM
2714 ESND ESOM ETRN EVFY } \
2715         normalize_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT
2716 RSET SAML SEND SOML } \
2717         normalize_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT
2718 X-DRCP X-ERCP X-EXCH50 } \
2719         normalize_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN
2720 XLICENSE XQUE XSTA XTRN XUSR } \
2721         max_command_line_len 512 \
2722         max_header_line_len 1000 \
2723         max_response_line_len 512 \
2724         alt_max_command_line_len 260 { MAIL } \
2725         alt_max_command_line_len 300 { RCPT } \
2726         alt_max_command_line_len 500 { HELP HELO ETRN EHLO } \
2727         alt_max_command_line_len 255 { EXPN VRFY ATRN SIZE BDAT DEBUG EMAL
2728 ESAM ESND ESOM EVFY IDENT NOOP RSET } \
2729         alt_max_command_line_len 246 { SEND SAML SOML AUTH TURN ETRN DATA
2730 RSET QUIT ONEX QUEU STARTTLS TICK TIME TURNME VERB X-EXPS X-LINK2STATE
2731 XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \

```

```
2732         valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND
2733 ESOM ETRN EVFY } \
2734         valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET
2735 SAML SEND SOML } \
2736         valid_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP
2737 X-ERCP X-EXCH50 } \
2738         valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN
2739 XLICENSE XQUE XSTA XTRN XUSR } \
2740         xlink2state { enabled }
2741
2742 # Portscan detection. For more information, see README.sfportscan
2743 # preprocessor sfportscan: proto { all } memcap { 10000000 }
2744 sense_level { low }
2745
2746 # ARP spoof detection. For more information, see the Snort Manual -
2747 Configuring Snort - Preprocessors - ARP Spoof Preprocessor
2748 # preprocessor arpspoof
2749 # preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00
2750
2751 # SSH anomaly detection. For more information, see README.ssh
2752 preprocessor ssh: server_ports { 22 } \
2753         autodetect \
2754         max_client_bytes 19600 \
2755         max_encrypted_packets 20 \
2756         max_server_version_len 100 \
2757         enable_respoverflow enable_sshlrcrc32 \
2758         enable_srvoverflow enable_protomismatch
2759
2760 # SMB / DCE-RPC normalization and anomaly detection. For more
2761 information, see README.dcerpc2
2762 preprocessor dcerpc2: memcap 102400, events [co ]
2763 preprocessor dcerpc2_server: default, policy WinXP, \
2764         detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593],
2765 \
2766         autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \
2767         smb_max_chain 3, smb_invalid_shares ["C$", "D$", "ADMIN$"]
2768
2769 # DNS anomaly detection. For more information, see README.dns
2770 preprocessor dns: ports { 53 } enable_rdata_overflow
2771
2772 # SSL anomaly detection and traffic bypass. For more information, see
2773 README.ssl
2774 preprocessor ssl: ports { 443 465 563 636 989 992 993 994 995 7801 7802
2775 7900 7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913
2776 7914 7915 7916 7917 7918 7919 7920 }, trustservers, noinspect_encrypted
```

```
2777
2778     # SDF sensitive data preprocessor.  For more information see
2779     README.sensitive_data
2780     preprocessor sensitive_data: alert_threshold 25
2781
2782     # SIP Session Initiation Protocol preprocessor.  For more information
2783     see README.sip
2784     preprocessor sip: max_sessions 40000, \
2785         ports { 5060 5061 5600 }, \
2786         methods { invite \
2787             cancel \
2788             ack \
2789             bye \
2790             register \
2791             options \
2792             refer \
2793             subscribe \
2794             update \
2795             join \
2796             info \
2797             message \
2798             notify \
2799             benotify \
2800             do \
2801             qauth \
2802             sprack \
2803             publish \
2804             service \
2805             unsubscribe \
2806             prack }, \
2807         max_uri_len 512, \
2808         max_call_id_len 80, \
2809         max_requestName_len 20, \
2810         max_from_len 256, \
2811         max_to_len 256, \
2812         max_via_len 1024, \
2813         max_contact_len 512, \
2814         max_content_len 2048
2815
2816     # IMAP preprocessor.  For more information see README.imap
2817     preprocessor imap: \
2818         ports { 143 } \
2819         b64_decode_depth 0 \
```

```
2820         qp_decode_depth 0 \
2821         bitenc_decode_depth 0 \
2822         uu_decode_depth 0
2823
2824     # POP preprocessor. For more information see README.pop
2825     preprocessor pop: \
2826         ports { 110 } \
2827         b64_decode_depth 0 \
2828         qp_decode_depth 0 \
2829         bitenc_decode_depth 0 \
2830         uu_decode_depth 0
2831
2832     # Modbus preprocessor. For more information see README.modbus
2833     preprocessor modbus: ports { 502 }
2834
2835     # DNP3 preprocessor. For more information see README.dnp3
2836     preprocessor dnp3: ports { 20000 } \
2837         memcap 262144 \
2838         check_crc
2839
2840     #
2841     # Note to Debian users: this is disabled since it is an experimental
2842     # preprocessor. If you want to use it you have to create the rules
2843     # files
2844     # referenced below in the /etc/snort/rules directory
2845     #
2846     # Reputation preprocessor. For more information see README.reputation
2847     #preprocessor reputation: \
2848     #     memcap 500, \
2849     #     priority whitelist, \
2850     #     nested_ip inner, \
2851     #     whitelist $WHITE_LIST_PATH/white_list.rules, \
2852     #     blacklist $BLACK_LIST_PATH/black_list.rules
2853
2854     #####
2855     # Step #6: Configure output plugins
2856     # For more information, see Snort Manual, Configuring Snort - Output
2857     # Modules
2858     #####
2859
2860     # unified2
2861     # Recommended for most installs
2862     # output unified2: filename merged.log, limit 128, nostamp,
2863     mpls_event_types, vlan_event_types
```

```

2864     #output unified2: filename snort.log, limit 128, nostamp,
2865     mpls_event_types, vlan_event_types
2866     output unified2: filename /var/log/snort/snort.log, limit 128,
2867     mpls_event_types, vlan_event_types
2868
2869     # Additional configuration for specific types of installs
2870     # output alert_unified2: filename snort.alert, limit 128, nostamp
2871     # output log_unified2: filename snort.log, limit 128, nostamp
2872
2873     # syslog
2874     # output alert_syslog: LOG_AUTH LOG_ALERT
2875
2876     # pcap
2877     # output log_tcpdump: tcpdump.log
2878
2879     # metadata reference data. do not modify these lines
2880     include classification.config
2881     include reference.config
2882
2883
2884     #####
2885     # Step #7: Customize your rule set
2886     # For more information, see Snort Manual, Writing Snort Rules
2887     #
2888     # NOTE: All categories are enabled in this conf file
2889     #####
2890
2891     # Note to Debian users: The rules preinstalled in the system
2892     # can be *very* out of date. For more information please read
2893     # the /usr/share/doc/snort-rules-default/README.Debian file
2894
2895     #
2896     # If you install the official VRT Sourcefire rules please review this
2897     # configuration file and re-enable (remove the comment in the first
2898     # line) those
2899     # rules files that are available in your system (in the
2900     # /etc/snort/rules
2901     # directory)
2902
2903     # site specific rules
2904     include $RULE_PATH/local.rules
2905
2906     #include $RULE_PATH/app-detect.rules
2907     include $RULE_PATH/attack-responses.rules

```

```
2908     include $RULE_PATH/backdoor.rules
2909     include $RULE_PATH/bad-traffic.rules
2910     #include $RULE_PATH/blacklist.rules
2911     #include $RULE_PATH/botnet-cnc.rules
2912     #include $RULE_PATH/browser-chrome.rules
2913     #include $RULE_PATH/browser-firefox.rules
2914     #include $RULE_PATH/browser-ie.rules
2915     #include $RULE_PATH/browser-other.rules
2916     #include $RULE_PATH/browser-plugins.rules
2917     #include $RULE_PATH/browser-webkit.rules
2918     include $RULE_PATH/chat.rules
2919     #include $RULE_PATH/content-replace.rules
2920     include $RULE_PATH/ddos.rules
2921     include $RULE_PATH/dns.rules
2922     include $RULE_PATH/dos.rules
2923     include $RULE_PATH/experimental.rules
2924     #include $RULE_PATH/exploit-kit.rules
2925     include $RULE_PATH/exploit.rules
2926     #include $RULE_PATH/file-executable.rules
2927     #include $RULE_PATH/file-flash.rules
2928     #include $RULE_PATH/file-identify.rules
2929     #include $RULE_PATH/file-image.rules
2930     #include $RULE_PATH/file-java.rules
2931     #include $RULE_PATH/file-multimedia.rules
2932     #include $RULE_PATH/file-office.rules
2933     #include $RULE_PATH/file-other.rules
2934     #include $RULE_PATH/file-pdf.rules
2935     include $RULE_PATH/finger.rules
2936     include $RULE_PATH/ftp.rules
2937     include $RULE_PATH/icmp-info.rules
2938     include $RULE_PATH/icmp.rules
2939     include $RULE_PATH/imap.rules
2940     #include $RULE_PATH/indicator-compromise.rules
2941     #include $RULE_PATH/indicator-obfuscation.rules
2942     #include $RULE_PATH/indicator-scan.rules
2943     #include $RULE_PATH/indicator-shellcode.rules
2944     include $RULE_PATH/info.rules
2945     #include $RULE_PATH/malware-backdoor.rules
2946     #include $RULE_PATH/malware-cnc.rules
2947     #include $RULE_PATH/malware-other.rules
2948     #include $RULE_PATH/malware-tools.rules
2949     include $RULE_PATH/misc.rules
2950     include $RULE_PATH/multimedia.rules
```

```
2951     include $RULE_PATH/mysql.rules
2952     include $RULE_PATH/netbios.rules
2953     include $RULE_PATH/nntp.rules
2954     include $RULE_PATH/oracle.rules
2955     #include $RULE_PATH/os-linux.rules
2956     #include $RULE_PATH/os-mobile.rules
2957     #include $RULE_PATH/os-other.rules
2958     #include $RULE_PATH/os-solaris.rules
2959     #include $RULE_PATH/os-windows.rules
2960     include $RULE_PATH/other-ids.rules
2961     include $RULE_PATH/p2p.rules
2962     #include $RULE_PATH/phishing-spam.rules
2963     #include $RULE_PATH/policy-multimedia.rules
2964     #include $RULE_PATH/policy-other.rules
2965     include $RULE_PATH/policy.rules
2966     #include $RULE_PATH/policy-social.rules
2967     #include $RULE_PATH/policy-spam.rules
2968     include $RULE_PATH/pop2.rules
2969     include $RULE_PATH/pop3.rules
2970     #include $RULE_PATH/protocol-dns.rules
2971     #include $RULE_PATH/protocol-finger.rules
2972     #include $RULE_PATH/protocol-ftp.rules
2973     #include $RULE_PATH/protocol-icmp.rules
2974     #include $RULE_PATH/protocol-imap.rules
2975     #include $RULE_PATH/protocol-nntp.rules
2976     #include $RULE_PATH/protocol-pop.rules
2977     #include $RULE_PATH/protocol-rpc.rules
2978     #include $RULE_PATH/protocol-scada.rules
2979     #include $RULE_PATH/protocol-services.rules
2980     #include $RULE_PATH/protocol-snmp.rules
2981     #include $RULE_PATH/protocol-telnet.rules
2982     #include $RULE_PATH/protocol-tftp.rules
2983     #include $RULE_PATH/protocol-voip.rules
2984     #include $RULE_PATH/pua-adware.rules
2985     #include $RULE_PATH/pua-other.rules
2986     #include $RULE_PATH/pua-p2p.rules
2987     #include $RULE_PATH/pua-toolbars.rules
2988     include $RULE_PATH/rpc.rules
2989     include $RULE_PATH/rservices.rules
2990     #include $RULE_PATH/scada.rules
2991     include $RULE_PATH/scan.rules
2992     #include $RULE_PATH/server-apache.rules
2993     #include $RULE_PATH/server-iis.rules
```

```
2994      #include $RULE_PATH/server-mail.rules
2995      #include $RULE_PATH/server-mssql.rules
2996      #include $RULE_PATH/server-mysql.rules
2997      #include $RULE_PATH/server-oracle.rules
2998      #include $RULE_PATH/server-other.rules
2999      #include $RULE_PATH/server-samba.rules
3000      #include $RULE_PATH/server-webapp.rules
3001      #
3002      # Note: These rules are disable by default as they are
3003      # too coarse grained. Enabling them causes a large
3004      # performance impact
3005      #include $RULE_PATH/shellcode.rules
3006      include $RULE_PATH/smtp.rules
3007      include $RULE_PATH/snmp.rules
3008      #include $RULE_PATH/specific-threats.rules
3009      #include $RULE_PATH/spyware-put.rules
3010      include $RULE_PATH/sql.rules
3011      include $RULE_PATH/telnet.rules
3012      include $RULE_PATH/tftp.rules
3013      include $RULE_PATH/virus.rules
3014      #include $RULE_PATH/voip.rules
3015      #include $RULE_PATH/web-activex.rules
3016      include $RULE_PATH/web-attacks.rules
3017      include $RULE_PATH/web-cgi.rules
3018      include $RULE_PATH/web-client.rules
3019      include $RULE_PATH/web-coldfusion.rules
3020      include $RULE_PATH/web-frontpage.rules
3021      include $RULE_PATH/web-iis.rules
3022      include $RULE_PATH/web-misc.rules
3023      include $RULE_PATH/web-php.rules
3024      include $RULE_PATH/xll.rules
3025      include $RULE_PATH/community-sql-injection.rules
3026      include $RULE_PATH/community-web-client.rules
3027      include $RULE_PATH/community-web-dos.rules
3028      include $RULE_PATH/community-web-iis.rules
3029      include $RULE_PATH/community-web-misc.rules
3030      include $RULE_PATH/community-web-php.rules
3031      include $RULE_PATH/community-sql-injection.rules
3032      include $RULE_PATH/community-web-client.rules
3033      include $RULE_PATH/community-web-dos.rules
3034      include $RULE_PATH/community-web-iis.rules
3035      include $RULE_PATH/community-web-misc.rules
3036      include $RULE_PATH/community-web-php.rules
```

```

3037
3038
3039 #####
3040 # Step #8: Customize your preprocessor and decoder alerts
3041 # For more information, see README.decoder_preproc_rules
3042 #####
3043
3044 # decoder and preprocessor event rules
3045 # include $PREPROC_RULE_PATH/preprocessor.rules
3046 # include $PREPROC_RULE_PATH/decoder.rules
3047 # include $PREPROC_RULE_PATH/sensitive-data.rules
3048
3049 #####
3050 # Step #9: Customize your Shared Object Snort Rules
3051 # For more information, see
3052 http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html
3053
3054 #####
3055 # dynamic library rules
3056 # include $SO_RULE_PATH/bad-traffic.rules
3057 # include $SO_RULE_PATH/chat.rules
3058 # include $SO_RULE_PATH/dos.rules
3059 # include $SO_RULE_PATH/exploit.rules
3060 # include $SO_RULE_PATH/icmp.rules
3061 # include $SO_RULE_PATH/imap.rules
3062 # include $SO_RULE_PATH/misc.rules
3063 # include $SO_RULE_PATH/multimedia.rules
3064 # include $SO_RULE_PATH/netbios.rules
3065 # include $SO_RULE_PATH/nntp.rules
3066 # include $SO_RULE_PATH/p2p.rules
3067 # include $SO_RULE_PATH/smtp.rules
3068 # include $SO_RULE_PATH/snmp.rules
3069 # include $SO_RULE_PATH/specific-threats.rules
3070 # include $SO_RULE_PATH/web-activex.rules
3071 # include $SO_RULE_PATH/web-client.rules
3072 # include $SO_RULE_PATH/web-iis.rules
3073 # include $SO_RULE_PATH/web-misc.rules
3074
3075 # Event thresholding or suppression commands. See threshold.conf
3076 include threshold.conf

```

```
3076 /etc/snort/snort.debian.conf
3077 # snort.debian.config (Debian Snort configuration file)
3078 #
3079 # This file was generated by the post-installation script of the snort
3080 # package using values from the debconf database.
3081 #
3082 # It is used for options that are changed by Debian to leave
3083 # the original configuration files untouched.
3084 #
3085 # This file is automatically updated on upgrades of the snort package
3086 # *only* if it has not been modified since the last upgrade of that
3087 # package.
3088 #
3089 # If you have edited this file but would like it to be automatically
3090 # updated
3091 # again, run the following command as root:
3092 #   dpkg-reconfigure snort
3093
3094 DEBIAN_SNORT_STARTUP="boot"
3095 DEBIAN_SNORT_HOME_NET="172.16.0.0/16"
3096 DEBIAN_SNORT_OPTIONS=""
3097 DEBIAN_SNORT_INTERFACE="eth0"
3098 DEBIAN_SNORT_SEND_STATS="true"
3099 DEBIAN_SNORT_STATS_RCPT="root"
3100 DEBIAN_SNORT_STATS_THRESHOLD="1"
```

```
3101 /usr/local/etc/barnyard2.conf
3102 Also linked from /etc/snort/barnyard.conf.
3103 #
3104 # Barnyard2 example configuration file
3105 #
3106
3107 #
3108 # This file contains a sample barnyard2 configuration.
3109 # You can take the following steps to create your own custom
3110 # configuration:
3111 #
3112 #   1) Configure the variable declarations
3113 #   2) Setup the input plugins
3114 #   3) Setup the output plugins
3115 #
3116
```

```
3117      #
3118      # Step 1: configure the variable declarations
3119      #
3120
3121      # in order to keep from having a commandline that uses every letter in
3122      the
3123      # alphabet most configuration options are set here.
3124
3125      # use UTC for timestamps
3126      #
3127      #config utc
3128
3129      # set the appropriate paths to the file(s) your Snort process is using.
3130      #
3131      config reference_file:      /etc/snort/etc/reference.config
3132      config classification_file: /etc/snort/etc/classification.config
3133      config gen_file:           /etc/snort/gen-msg.map
3134      config sid_file:           /etc/snort/etc/sid-msg.map
3135
3136
3137      # Configure signature suppression at the spooler level see
3138      doc/README.sig_suppress
3139      #
3140      #
3141      #config sig_suppress: 1:10
3142
3143
3144      # Set the event cache size to defined max value before recycling of
3145      event occur.
3146      #
3147      #
3148      #config event_cache_size: 4096
3149
3150      # define dedicated references similar to that of snort.
3151      #
3152      #config reference: mybugs http://www.mybugs.com/?s=
3153
3154      # define explicit classifications similar to that of snort.
3155      #
3156      #config classification: shortname, short description, priority
3157
3158      # set the directory for any output logging
3159      #
3160      config logdir: /var/log/barnyard2
```

```
3161
3162     # to ensure that any plugins requiring some level of uniqueness in
3163     their output
3164     # the alert_with_interface_name, interface and hostname directives are
3165     provided.
3166     # An example of usage would be to configure them to the values of the
3167     associated
3168     # snort process whose unified files you are reading.
3169     #
3170     # Example:
3171     #   For a snort process as follows:
3172     #       snort -i eth0 -c /etc/snort.conf
3173     #
3174     #   Typical options would be:
3175     #       config hostname:  thor
3176     #       config interface: eth0
3177     #       config alert_with_interface_name
3178     #
3179     config hostname:    snort
3180     config interface:  eth0
3181
3182     # enable printing of the interface name when alerting.
3183     #
3184     #config alert_with_interface_name
3185
3186     # at times snort will alert on a packet within a stream and dump that
3187     stream to
3188     # the unified output. barnyard2 can generate output on each packet of
3189     that
3190     # stream or the first packet only.
3191     #
3192     #config alert_on_each_packet_in_stream
3193
3194     # enable daemon mode
3195     #
3196     config daemon
3197
3198     # make barnyard2 process chroot to directory after initialisation.
3199     #
3200     #config chroot: /var/spool/barnyard2
3201
3202     # specifiy the group or GID for barnyard2 to run as after
3203     initialisation.
3204     #
```

```
3205     #config set_gid: 999
3206
3207     # specify the user or UID for barnyard2 to run as after
3208     initialisation.
3209     #
3210     #config set_uid: 999
3211
3212     # specify the directory for the barnyard2 PID file.
3213     #
3214     #config pidpath: /var/run/by2.pid
3215
3216     # enable decoding of the data link (or second level headers).
3217     #
3218     #config decode_data_link
3219
3220     # dump the application data
3221     #
3222     #config dump_payload
3223
3224     # dump the application data as chars only
3225     #
3226     #config dump_chars_only
3227
3228     # enable verbose dumping of payload information in log style output
3229     plugins.
3230     #
3231     #config dump_payload_verbose
3232
3233     # enable obfuscation of logged IP addresses.
3234     #
3235     #config obfuscate
3236
3237     # enable the year being shown in timestamps
3238     #
3239     config show_year
3240
3241     # set the umask for all files created by the barnyard2 process (eg. log
3242     files).
3243     #
3244     #config umask: 066
3245
3246     # enable verbose logging
3247     #
3248     #config verbose
```

```
3249
3250     # quiet down some of the output
3251     #
3252     #config quiet
3253
3254     # define the full waldo filepath.
3255     #
3256     config waldo_file: /tmp/waldo
3257
3258     # specify the maximum length of the MPLS label chain
3259     #
3260     #config max_mpls_labelchain_len: 64
3261
3262     # specify the protocol (ie ipv4, ipv6, ethernet) that is encapsulated
3263     by MPLS.
3264     #
3265     #config mpls_payload_type: ipv4
3266
3267     # set the reference network or homenet which is predominantly used by
3268     the
3269     # log_ascii plugin.
3270     #
3271     #config reference_net: 192.168.0.0/24
3272
3273     #
3274     # CONTINUOUS MODE
3275     #
3276
3277     # set the archive directory for use with continuous mode
3278     #
3279     #config archivedir: /tmp
3280
3281     # when in operating in continuous mode, only process new records and
3282     ignore any
3283     # existing unified files
3284     #
3285     #config process_new_records_only
3286
3287
3288     #
3289     # Step 2: setup the input plugins
3290     #
3291
3292     # this is not hard, only unified2 is supported ;)
```

```

3293     input unified2
3294
3295
3296     #
3297     # Step 3: setup the output plugins
3298     #
3299
3300     # alert_cef
3301     #
3302     -----
3303     #
3304     # Purpose:
3305     #   This output module provides the ability to output alert information
3306     #   to a
3307     #   remote network host as well as the local host using the open standard
3308     #   Common Event Format (CEF).
3309     #
3310     # Arguments: host=hostname[:port], severity facility
3311     #               arguments should be comma delimited.
3312     #   host      - specify a remote hostname or IP with optional port
3313     #               number
3314     #               this is only specific to WIN32 (and is not yet fully
3315     #               supported)
3316     #   severity  - as defined in RFC 3164 (eg. LOG_WARN, LOG_INFO)
3317     #   facility  - as defined in RFC 3164 (eg. LOG_AUTH, LOG_LOCAL0)
3318     #
3319     # Examples:
3320     #   output alert_cef
3321     #   output alert_cef: host=192.168.10.1
3322     #   output alert_cef: host=sysserver.com:1001
3323     #   output alert_cef: LOG_AUTH LOG_INFO
3324     #
3325
3326     # alert_bro
3327     #
3328     -----
3329     #
3330     # Purpose: Send alerts to a Bro-IDS instance.
3331     #
3332     # Arguments: hostname:port
3333     #
3334     # Examples:
3335     #   output alert_bro: 127.0.0.1:47757
3336

```

```
3337      # alert_fast
3338      #
3339      -----
3340      # Purpose: Converts data to an approximation of Snort's "fast alert"
3341      mode.
3342      #
3343      # Arguments: file <file>, stdout
3344      #             arguments should be comma delimited.
3345      #   file - specifiy alert file
3346      #   stdout - no alert file, just print to screen
3347      #
3348      # Examples:
3349      #   output alert_fast
3350      #   output alert_fast: stdout
3351      #
3352      #output alert_fast: stdout
3353      output alert_fast: /var/log/snort/alert
3354
3355
3356      # prelude: log to the Prelude Hybrid IDS system
3357      #
3358      -----
3359      #
3360      # Purpose:
3361      #   This output module provides logging to the Prelude Hybrid IDS system
3362      #
3363      # Arguments: profile=snort-profile
3364      #   snort-profile   - name of the Prelude profile to use (default is
3365      snort).
3366      #
3367      # Snort priority to IDMEF severity mappings:
3368      # high < medium < low < info
3369      #
3370      # These are the default mapped from classification.config:
3371      # info    = 4
3372      # low     = 3
3373      # medium  = 2
3374      # high   = anything below medium
3375      #
3376      # Examples:
3377      #   output alert_prelude
3378      #   output alert_prelude: profile=snort-profile-name
3379      #
3380
```

```

3381     # alert_syslog
3382     #
3383     -----
3384     #
3385     # Purpose:
3386     #   This output module provides the ability to output alert information
3387     #   to local syslog
3388     #
3389     #   severity    - as defined in RFC 3164 (eg. LOG_WARN, LOG_INFO)
3390     #   facility    - as defined in RFC 3164 (eg. LOG_AUTH, LOG_LOCAL0)
3391     #
3392     # Examples:
3393     #   output alert_syslog
3394     #   output alert_syslog: LOG_AUTH LOG_INFO
3395     #
3396     output alert_syslog: LOG_AUTH LOG_INFO
3397
3398     # syslog_full
3399     #-----
3400     # Available as both a log and alert output plugin. Used to output data
3401     # via TCP/UDP or LOCAL ie(syslog())
3402     # Arguments:
3403     #   sensor_name $sensor_name          - unique sensor name
3404     #   server $server                    - server the device will report
3405     #   to
3406     #   local                                              - if defined, ignore all remote
3407     #   information and use syslog() to send message.
3408     #   protocol $protocol                      - protocol device will report
3409     #   over (tcp/udp)
3410     #   port $port                                  - destination port device will
3411     #   report to (default: 514)
3412     #   delimiters $delimiters                  - define a character that will
3413     #   delimit message sections ex: "|", will use | as message section
3414     #   delimiters. (default: |)
3415     #   separators $separators                  - define field separator
3416     #   included in each message ex: " ", will use space as field separator.
3417     #   (default: [:space:])
3418     #   operation_mode $operation_mode          - default | complete : default
3419     #   mode is compatible with default snort syslog message, complete prints
3420     #   more information such as the raw packet (hexed)
3421     #   log_priority $log_priority              - used by local option for
3422     #   syslog priority call. (man syslog(3) for supported options) (default:
3423     #   LOG_INFO)
3424     #   log_facility $log_facility              - used by local option for
3425     #   syslog facility call. (man syslog(3) for supported options) (default:
3426     #   LOG_USER)

```

```
3427      #      payload_encoding      - (default: hex) support
3428      hex/ascii/base64 for log_syslog_full using operation_mode complete
3429      only.
3430
3431      # Usage Examples:
3432      # output alert_syslog_full: sensor_name snortIds1-eth2, server
3433      xxx.xxx.xxx.xxx, protocol udp, port 514, operation_mode default
3434      # output alert_syslog_full: sensor_name snortIds1-eth2, server
3435      xxx.xxx.xxx.xxx, protocol udp, port 514, operation_mode complete
3436      # output log_syslog_full: sensor_name snortIds1-eth2, server
3437      xxx.xxx.xxx.xxx, protocol udp, port 514, operation_mode default
3438      # output log_syslog_full: sensor_name snortIds1-eth2, server
3439      xxx.xxx.xxx.xxx, protocol udp, port 514, operation_mode complete
3440      # output alert_syslog_full: sensor_name snortIds1-eth2, server
3441      xxx.xxx.xxx.xxx, protocol udp, port 514
3442      # output log_syslog_full: sensor_name snortIds1-eth2, server
3443      xxx.xxx.xxx.xxx, protocol udp, port 514
3444      # output alert_syslog_full: sensor_name snortIds1-eth2, local
3445      # output log_syslog_full: sensor_name snortIds1-eth2, local,
3446      log_priority LOG_CRIT,log_facility LOG_CRON
3447
3448      # log_ascii
3449      #
3450      -----
3451      #
3452      # Purpose: This output module provides the default packet logging
3453      functionality
3454      #
3455      # Arguments: None.
3456      #
3457      # Examples:
3458      #   output log_ascii
3459      #
3460      output log_ascii
3461
3462      # log_tcpdump
3463      #
3464      -----
3465      #
3466      # Purpose
3467      #   This output module logs packets in binary tcpdump format
3468      #
3469      # Arguments:
3470      #   The only argument is the output file name.
3471      #
```

```

3472     # Examples:
3473     #   output log_tcpdump: tcpdump.log
3474     #
3475     output log_tcpdump: /var/log/snort/tcpdump.log
3476
3477     # sgul
3478     #
3479     -----
3480     #
3481     # Purpose: This output module provides logging ability for the sgul
3482     interface
3483     # See doc/README.sgul
3484     #
3485     # Arguments: agent_port <port>, sensor_name <name>
3486     #             arguments should be comma delimited.
3487     #   agent_port - explicitly set the sgul agent listening port
3488     #                 (default: 7736)
3489     #   sensor_name - explicitly set the sensor name
3490     #                 (default: machine hostname)
3491     #
3492     # Examples:
3493     #   output sgul
3494     #   output sgul: agent_port=7000
3495     #   output sgul: sensor_name=argyle
3496     #   output sgul: agent_port=7000, sensor_name=argyle
3497     #
3498
3499
3500     # database: log to a variety of databases
3501     #
3502     -----
3503     #
3504     # Purpose: This output module provides logging ability to a variety of
3505     databases
3506     # See doc/README.database for additional information.
3507     #
3508     # Examples:
3509     #   output database: log, mysql, user=root password=test dbname=db
3510     host=localhost
3511     #   output database: alert, postgresql, user=snort dbname=snort
3512     #   output database: log, odbc, user=snort dbname=snort
3513     #   output database: log, mssql, dbname=snort user=snort password=test
3514     #   output database: log, oracle, dbname=snort user=snort password=test
3515     #

```

```
3516      #output database: log, mysql, user=root password=1Password!
3517      dbname=snortdb
3518
3519      # alert_fwsam: allow blocking of IP's through remote services
3520      #
3521      -----
3522      # output alert_fwsam: <SnortSam Station>:<port>/<key>
3523      #
3524      # <FW Mgmt Station>: IP address or host name of the host running
3525      SnortSam.
3526      # <port>:          Port the remote SnortSam service listens on (default
3527      898).
3528      # <key>:           Key used for authentication (encryption really)
3529      #                   of the communication to the remote service.
3530      #
3531      # Examples:
3532      #
3533      # output alert_fwsam: snortsambox/idspassword
3534      # output alert_fwsam: fw1.domain.tld:898/mykey
3535      # output alert_fwsam: 192.168.0.1/borderfw 192.168.1.254/wanfw
3536      #
```

```
3537      /opt/splunkforwarder/etc/system/local/server.conf
3538
3539      [sslConfig]
3540      sslKeysfilePassword = $1$A0zU/599eO4g
3541
3542      [lmpool:auto_generated_pool_forwarder]
3543      description = auto_generated_pool_forwarder
3544      quota = MAX
3545      slaves = *
3546      stack_id = forwarder
3547
3548      [lmpool:auto_generated_pool_free]
3549      description = auto_generated_pool_free
3550      quota = MAX
3551      slaves = *
3552      stack_id = free
3553
3554      [general]
3555      pass4SymmKey = $1$VACAO9o7M7wg
3556      serverName = snort
```

```
3553      /opt/splunkforwarder/etc/system/local/inputs.conf
```

```
3554      Note: The sourcetype=snort_alert_full is important if you are using the Splunk TA_Snort app.
```

```

3555     [default]
3556     host=snort
3557     sourcetype=snort_alert_full
3558     index=snort

3559     [monitor:///var/log/snort/alert]
3560     sourcetype=snort_alert_full

```

```

3561 /opt/splunkforwarder/etc/system/local/outputs.conf
3562 [tcpout]
3563 defaultGroup = splunkssl

3564 [tcpout:splunkssl]
3565 server = loghost:9997
3566 compressed = true
3567 sslVerifyServerCert = false
3568 sslRootCAPath = $SPLUNK_HOME/etc/certs/CAServerCert.pem
3569 sslCertPath = $SPLUNK_HOME/etc/certs/snort.lab5.nccoe.gov.pem
3570 sslPassword = $!$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 the following figure.

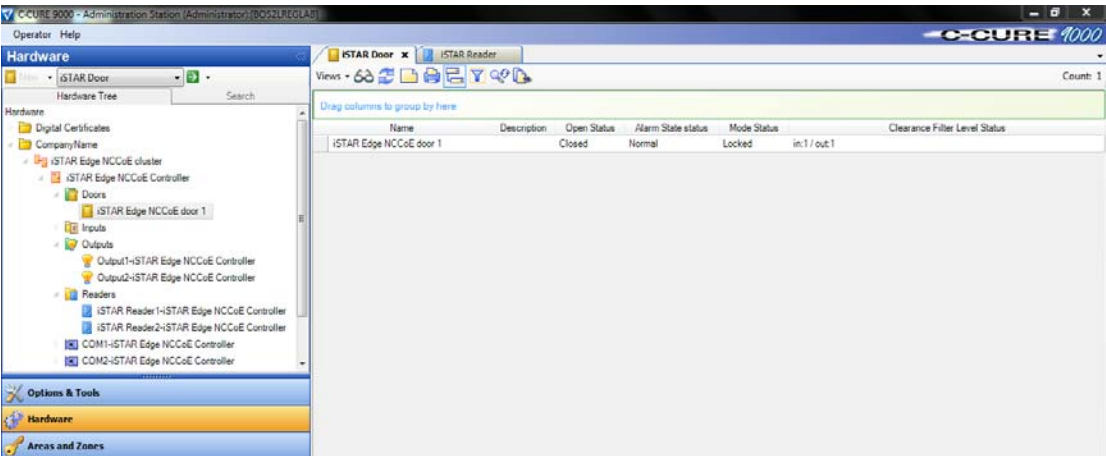


Figure 3.1 CCURE 9000 Overview

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.

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) ([Unused]).	8/20/2015 12:56:15 PM

Figure 3.2 CCURE 9000 Messages

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.

3638 3.10.3 Network Configuration

3639 The management network interface card is configured as follows:

3640 IPv4 Manual

3641 IPv6 Disabled

3642 IP Address: 172.16.0.45

3643 Netmask: 255.255.255.0

3644 Gateway: 172.16.0.11

3645 DNS Servers: 172.16.1.20, 172.16.1.21

3646 Search Domains: lab5.nccoe.gov

3647 3.10.4 Installing WSUS

3648 WSUS is installed through the add roles and features wizard in Server Manager. Documentation
3649 is provided by Microsoft at

3650 <https://technet.microsoft.com/en-us/windowsserver/bb332157.aspx>.

3651 WSUS should NOT be a member of your domain.

3652 3.10.5 Configurations

3653 You configure WSUS using the WSUS Server Configuration Wizard. When the wizard prompts
3654 you, set these options as follows:

- 3655 ■ **Update Source and Proxy Server – Synchronize from Microsoft Update**
- 3656 ■ **Products and Classifications – Microsoft SQL Server 2012, Microsoft SQL Server 2014, SQL**
3657 **Server 2008 R2, SQL Server 2008, SQL Server 2012 Product Updates for Setup, SQL server**
3658 **Feature Pack, Windows 7, Windows Server 2012 R2 and later drivers, Windows Server**
3659 **2012 R2**
- 3660 ■ **Update Files and Languages – Store update files locally on this server < Download update**
3661 **files to this server only when updates are approved, Download updates only in English**
- 3662 ■ **Synchronization Schedule – Automatically > 1 per day**
- 3663 ■ **Automatic Approvals – Default**
- 3664 ■ **Computers – Use the Update Services console**
- 3665 ■ **Reporting Rollup – N/A**
- 3666 ■ **E-mail Notifications – N/A**
- 3667 ■ **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:

Filename: **WSUSReport.ps1**

```
$wsus
```

```
$wsusserver = 'wsus'
```

Load required Assemblies

```
[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration") | Out-Null
```

```
$wsus =
```

```
[Microsoft.UpdateServices.Administration.AdminProxy]::getUpdateServer('wsus', $False, 8530)
```

create update scope object

```
$updatescope = New-Object
```

```
Microsoft.UpdateServices.Administration.UpdateScope
```

```
$updatescope.IncludedInstallationStates =
```

```
[Microsoft.UpdateServices.Administration.UpdateInstallationStates]::NotInstalled
```

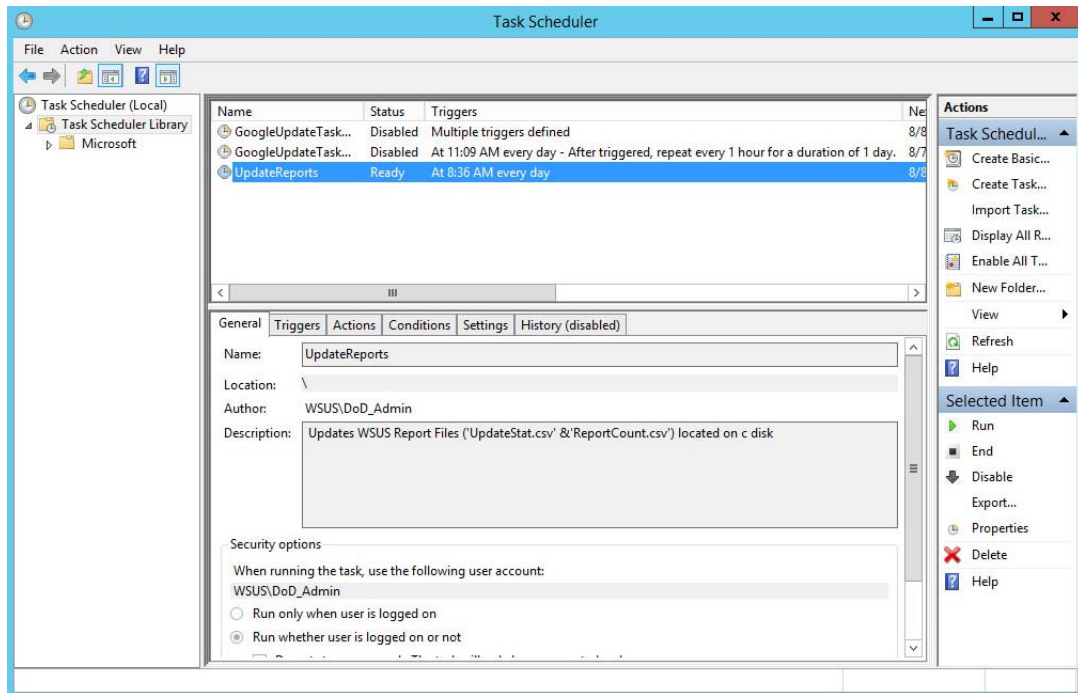
```
$updatescope.FromArrivalDate = [datetime]"12/13/2011"
```

```
3707 $computerscope = New-Object
3708 Microsoft.UpdateServices.Administration.ComputerTargetScope
3709 $wsus.GetSummariesPerComputerTarget($updatescope,$computerscope) |
3710 Select
3711 @{'L='ComputerTarget';E={( $wsus.GetComputerTarget([guid]$_ .ComputerTargetId) ).FullDomainName}},
3712 @{'L='NeededCount';E={( $_.DownloadedCount+$_ .NotInstalledCount)}} ,DownloadedCount,NotInstalledCount,InstalledCount,FailedCount | Export-Csv
3713 c:\ReportCount.csv
3714
3715 $wsus.GetUpdateApprovals($updatescope) | Select
3716 @{'L='ComputerTargetGroup';E={$_.GetComputerTargetGroup().Name}},
3717 @{'L='UpdateTitle';E={( $wsus.GetUpdate([guid]$_ .UpdateId.UpdateId.Guid) ).Title}}, GoLiveTime,AdministratorName,Deadline | Export-Csv
3718 c:\UpdateStat.csv
```

3721 This script creates two **CSV** files and places them on the **C** drive: **ReportCount.csv** and
3722 **UpdateStat.csv**. These two files contain the fields ComputerTarget, NeededCount,
3723 DownloadedCount, NotInstalledCount, InstalledCount, FailedCount; and
3724 ComputerTargetGroup, UpdateTitle, GoLiveTime, AdministratorName and Deadline,
3725 respectively.

3726 When the script is running error free, a task is scheduled for the script to run daily for updates
3727 to the data. To create a scheduled task, complete the following steps:

- 3728 1. Open Task Scheduler and select **Create Task**.
- 3729 2. Name the task and give it a description. Select **Run whether user is logged on or not**. Select
3730 **Run with highest privileges**. Configure for: **Windows Server 2012 R2**.
- 3731 3. Select the **Triggers** tab and select **New**. Create a trigger to run every day at the desired time.
- 3732 4. Select the **Actions** tab and select **New**. Under **Action**, select **Start a Program**. In the
3733 Program/script box enter
3734 **c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe** or browse for the
3735 **PowerShell** executable.
- 3736 5. In the arguments box insert **-ExecutionPolicy Bypass <locationofscript>**. Select **OK** to save
3737 the task.
- 3738 6. Use the defaults for the remaining settings. The scheduled task should look similar to the
3739 task highlighted in the following figure.



3740

3741 3.10.8 Installing Splunk Universal Forwarder

3742 **Note:** You will need a Splunk account to download the Splunk Universal Forwarder. It is free and
 3743 can be set up at:

3744 https://www.splunk.com/page/sign_up

3745 Download the Splunk Universal Forwarder from:

3746 http://www.splunk.com/en_us/download/universal-forwarder.html

3747 You want the latest version for OS version Windows (64-bit). Since this is installing on
 3748 Windows, select the file that ends in .msi. An example is:

3749 `splunkforwarder-6.2.5-272645-x64-release.msi`

3750 Detailed installation instructions can be found at:

3751 http://docs.splunk.com/Documentation/Splunk/6.2.4/Forwarding/DeployaWindowsdfmanually#Install_the_universal_forwarder.
 3752

3753 3.10.9 Configuring Splunk Universal Forwarder

3754 Configuring Splunk Universal Forwarder as shown in the FS-ITAM use case requires X.509
 3755 Certificates for the Splunk Enterprise server/indexer and each Splunk Universal Forwarder. You
 3756 will also need a copy of your certificate authority's public certificate.

3757 If you entered your certificates during install time, they will be located at:

3758 `C:\Program Files\SplunkUniversalForwarder\etc\auth`

3759 If not, you will need to manually copy your certificates here.

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
```

Modify **server.conf** so that:

- **ServerName=WSUS** is your hostname.
- `sslKeysfilePassword = <password for your private key>`

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.

C:\Program Files\SplunkUniversalForwarder\etc\system\local server.conf

```
[sslConfig]
```

```
sslKeysfilePassword = $1$sznWu23zCGHY
```

```
[general]
```

```
pass4SymmKey = $1$5HWC5yilQzPY
```

```
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
```

```
3789 C:\Program Files\SplunkUniversalForwarder\etc\system\local\inputs.conf
3790 [default]
3791 host = WSUS
3792 sourcetype = wsus
3793 index = wsus
3794 [script://$SPLUNK_HOME\bin\scripts\splunk-wmi.path]
3795 disabled = 0
3796 [monitor:///C:\ReportCount.csv]
3797 sourcetype=wsus_reportcount
3798 crcSalt is needed because this file doesn't change much and is small
3799 crcSalt = <SOURCE>
3800 ignoreOlderThan = 2d
3801 disabled = 0
3802 [monitor:///C:\UpdateStat.csv ]
3803 sourcetype=wsus_updatestat
3804 ignoreOlderThan = 2d
3805 disabled = 0
3806 C:\Program Files\SplunkUniversalForwarder\etc\system\local\outputs.conf
3807 [tcpout]
3808 defaultGroup = default-autolb-group
3809 [tcpout:default-autolb-group]
3810 server = loghost:9997
3811 [tcpout-server://loghost:9997]
3812 sslCertPath = C:\wsus.lab5.nccoe.gov.pem
3813 sslPassword = $1$sznWu23zCGHY
3814 sslRootCAPath = C:\Users\DoD_Admin\Downloads\CAServerCert.pem
3815
```

4 Tier 3

2	4.1	Active Directory Server	136
3	4.2	Asset Central.....	139
4	4.3	Email	141
5	4.4	Openswan (VPN)	144
6	4.5	Ubuntu Apt-Cacher.....	148
7	4.6	Windows 2012 Certificate Authority	150
8	4.7	Common PKI Activities.....	153
9	4.8	Process Improvement Achievers (PIA) Security Evaluation.....	155

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

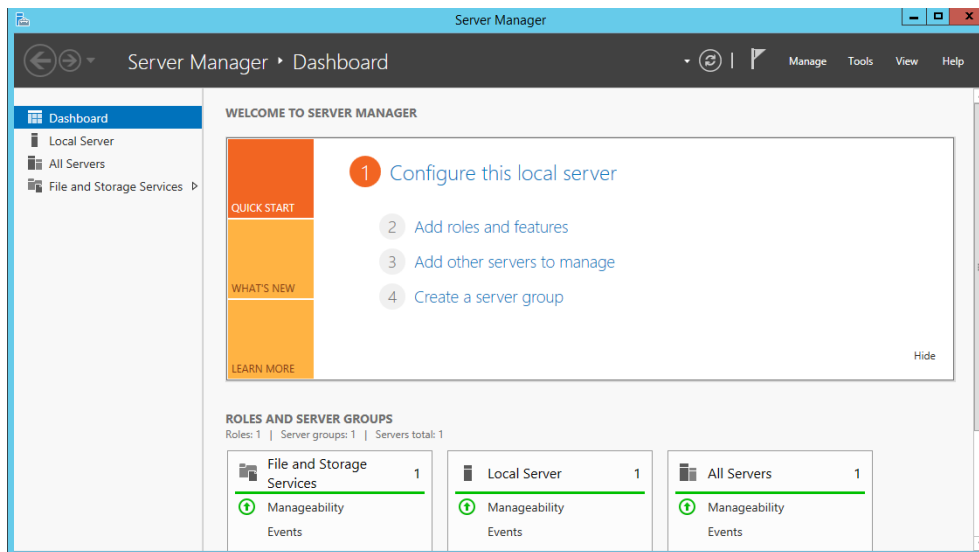
Active Directory provides centralized management, authentication, security, and information storage for end devices and users in a networked environment.

4.1.2 How It's Used

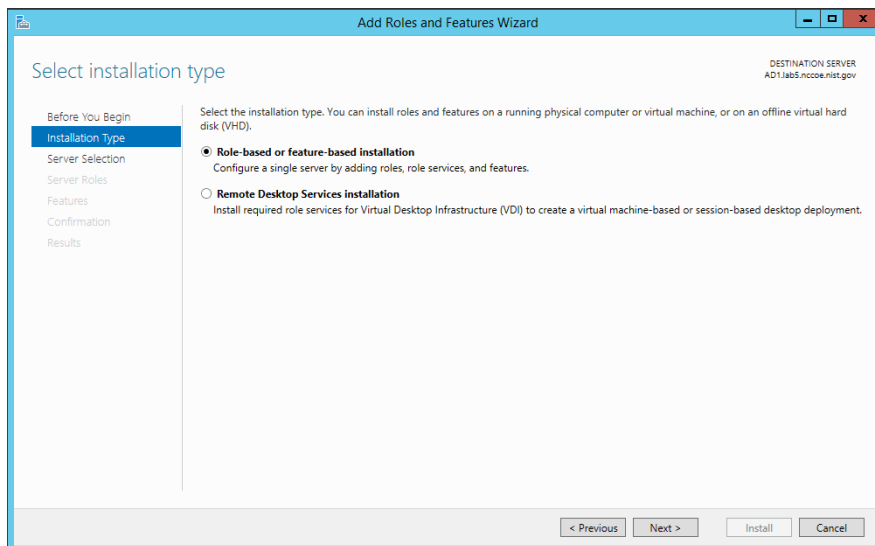
The Active Directory service is used in the ITAM build to provide authentication, user management and security within a mixed environment with Windows and Linux endpoints.

4.1.3 Installation

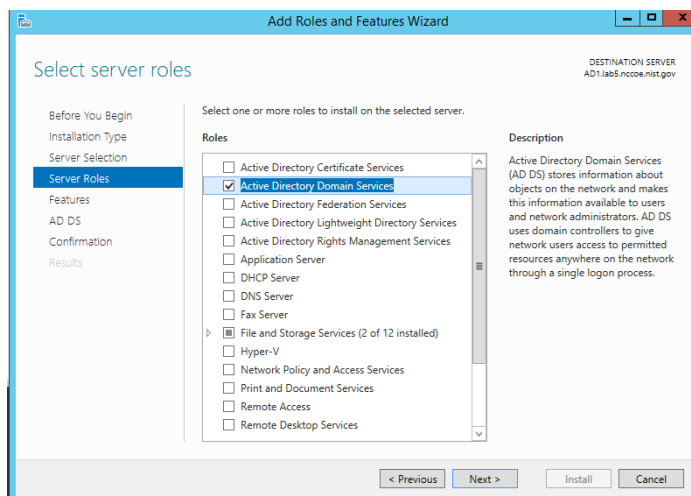
1. Go to Server Manager and click **Add Roles and Features Wizard**.



- 25 2. Click **Next** and select **Role-based or feature-based installation**. Then, click **Next**.

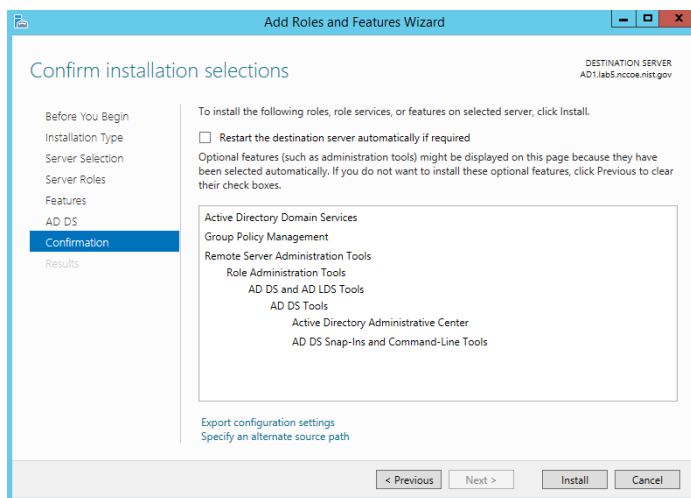


- 26
- 27 3. Ensure that the appropriate server name is selected. Then, click **Next**.
- 28 4. Click the checkbox next to **Active Directory Domain Services**. Then click **Next** to advance to
- 29 the next screen. Then, click **Add Features**.

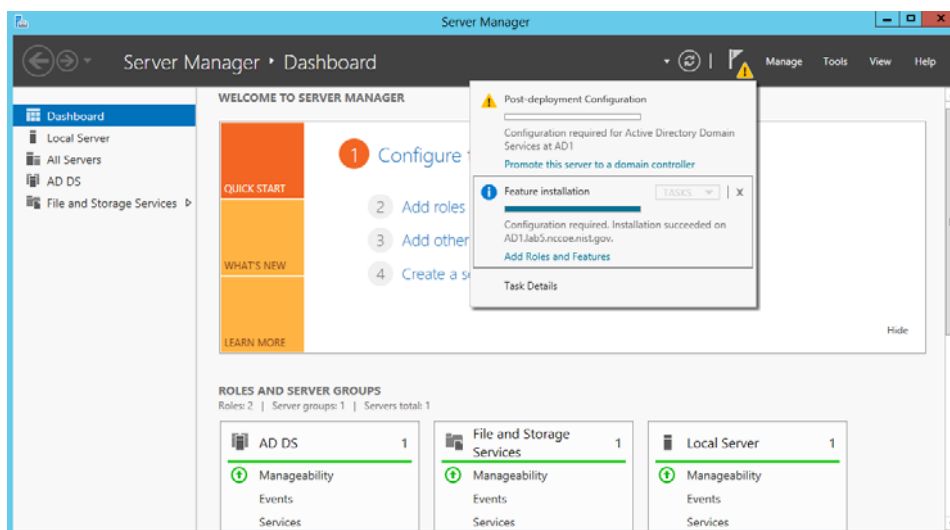


- 30
- 31 5. Use the features selected by default. Then, click **Next**.
- 32 6. In the Active Directory Domain Services screen, click **Next**.

- 33 7. On the Confirm installations selections screen, click **Install**.



- 34
- 35 8. When you see the message that the installation was successful, click **close**.
- 36 9. Return to the Server Manager and click on the yellow warning message.



- 37
- 38 10. On the Post-deployment Configuration box, click **Promote this server to a domain controller**.
- 39
- 40 11. Choose **Add a new forest**, specify the root domain name and click **Next**.
- 41 12. Use the default settings in the Domain Controller Options page. Ensure that **DNS server** is
- 42 selected. Enter the **Directory Services Restore Mode** password and click **Next**.
- 43 13. Choose a **NetBIOS domain Name** and click **Next**.
- 44 14. Accept the default locations for **AD DS**, **DS Database**, **log files** and **SYSVOL**.
- 45 15. In the Review Options screen, click **Next**.
- 46 16. Allow the system to complete the prerequisites check and click **Install**.
- 47 17. When the installation completes, reboot the system.

4.2 Asset Central

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 Email 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

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

Recommended versions:

RedHat	Enterprise Linux Server	Release	6.4 (Santiago) (x86_64)
Apache	httpd-2.2.15-26.el6.x86_64		
mysql	Server version:	5.1.66	
php	version	5.3.3 or	higher

4.2.5 Installing MySQL (MariaDB)

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

```
78      #systemctl start mariadb.service
79      #systemctl enable mariadb.service

80      # mysql_secure_installation

81      Answer the questions with the default answers while performing the
82      mysql_secure_installation.

83      Create a database - assetcentral

84      Create a user - assetcentral

85      Grant all privileges to assetcentral user
```

86 4.2.6 Installing Apache

```
87      # yum -y install httpd

88      #systemctl start httpd.service

89      #systemctl enable httpd.service

90      #firewall-cmd --permanent --zone=public --add-service=http
91      #firewall-cmd --permanent --zone=public --add-service=https
92      #firewall-cmd -reload

93      HTTP Configuration

94      Go to HTTPD root; normally (/etc/httpd).

95      Under the modules directory make sure libphp5.so exists.

96      Change documentroot (webroot) as per environment in httpd.conf.
```

97 4.2.7 Installing PHP5

```
98      #yum -y install php
99      #systemctl restart httpd.service
100     #yum search php
101     #yum -y install php-mysql
102     #yum -y install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc
103     php-mbstring php-snmp php-soap curl curl-devel

104     Restart Apache

105     #systemctl restart httpd.service
```

106 4.2.8 Post Installation Tasks

```
107     Copy AssetCentral files and folders from previous install to the new webroot.

108     Under the location (../assetcentral/application/config) make necessary changes as per
109     environment.
```

Sample

```

110
111 <?php defined('ASSET_CENTRAL')or    die('');
112 define('AC_URL_SUBDIR','/acprod');
113 define('AC_URL_SCRIPT','/index.php');
114 define('AC_URL_PARAM','go');
115 define('AC_URL_PREFIX',AC_URL_SUBDIR . AC_URL_SCRIPT.'?'
116         . AC_URL_PARAM . '=');
117 define('AC_ERROR_REPORTING',E_ERROR);
118 //      no slash at the end of this url
119 define('URL_SITE','http://10.1.xx.xxx');
120 define('OS','NIX'); // *NIX WIN BSD MAC
121 //      default database (read)
122 define('DB_TYPE_READ','MYSQL');
123 define('DB_HOST_READ','127.0.0.1');
124 //      usually leave this blank for MYSQL
125 define('DB_PORT_READ','');
126 define('DB_USER_READ','assetcentral');
127 define('DB_PASS_READ','xxxxx');
128 define('DB_DATA_READ','asset_prod');
129 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

144 DNS Servers: 172.16.1.20, 172.16.1.21

145 Search Domains: lab5.nccoe.gov

146 4.3.4 Installing Email

147 Email is installed on a hardened Ubuntu 14.04 Linux system. This email system is using the
148 Postfix email program. Complete installation instructions can be found at:

149 <https://help.ubuntu.com/community/Postfix#Installation>

150 For Debian/Ubuntu Linux systems: It is always best to make sure you system is up-to-date by
151 performing:

152 `sudo apt-get update`

153 `sudo apt-get upgrade`

154 `sudo apt-get install postfix`

155 4.3.5 Configure Email

156 From a terminal prompt:

157 `sudo dpkg-reconfigure postfix`

158 General type of mail configuration: **Internet Site**

159 NONE doesn't appear to be requested in current config.

160 System mail name: **mail1.lab5.nccoe.gov**

161 Root and postmaster mail recipient: <admin_user_name>

162 Other destinations for mail: email1, email1.lab5.nccoe.gov, localhost.lab5.nccoe.gov,
163 localhost.localdomain, localhost, lab5.nccoe.gov

164 Force synchronous updates on mail queue? No

165 Local networks: 172.16.0.0/16

166 Yes doesn't appear to be requested in current config.

167 Mailbox size limit (bytes): 0

168 Local address extension character: +

169 Internet protocols to use: all

170 Ensure that /etc/postfix/main.cf looks like the version below in the Configuration Files section.
171 Especially take note that the **inet_interfaces** setting. **inet_interfaces = loopback-only** will NOT
172 allow mail from other machines.

173 4.3.6 User Accounts

174 Create an account for each user that needs email:

175 `adduser <username>`

176 Then answer the questions.

177 4.3.7 DNS Settings

178 For mail to work correctly, an MX record must be set up on the DNS server.

179 The FS-ITAM build is using a Microsoft Server 2012R2 as its DNS server. First set up a DNS
180 A-Record for the email server, which looks like:

```
181 Host: email1
182 FQDN: email1.lab5.nccoe.gov
183 IP address: 172.16.1.50
```

184 Check next to Update associates pointer record.

185 Next create an MX record that looks like:

```
186 Host or child domain: (same as parent folder)
187 FQDN: lab5.nccoe.gov
188 FQDN of mail server: email1.lab5.nccoe.gov
189 Mail server priority: 10
```

190 4.3.8 Configuration Files

```
191 /etc/postfix/main.cf
192 # See /usr/share/postfix/main.cf.dist for a commented, more complete version
193
194 # Debian specific: Specifying a file name will cause the first
195 # line of that file to be used as the name. The Debian default
196 # is /etc/mailname.
197 #myorigin = /etc/mailname
198
199 smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
200 biff = no
201
202 # appending .domain is the MUA's job.
203 append_dot_mydomain = no
204
205 # Uncomment the next line to generate "delayed mail" warnings
206 #delay_warning_time = 4h
207
208 readme_directory = no
209
210 # TLS parameters
211 smtpd_tls_cert_file = /etc/ssl/certs/smtpd.crt
212 smtpd_tls_key_file = /etc/ssl/private/smtpd.key
213 smtpd_use_tls=yes
214 smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
```

```
215     smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
216
217     # See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
218     # information on enabling SSL in the smtp client.
219
220     smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
221     defer_unauth_destination
222     myhostname = mail1.lab5.nccoe.gov
223     alias_maps = hash:/etc/aliases
224     alias_database = hash:/etc/aliases
225     mydestination = email1, email1.lab5.nccoe.gov, localhost.lab5.nccoe.gov,
226     localhost.localdomain, localhost, lab5.nccoe.gov
227     relayhost =
228     mynetworks = 172.16.0.0/16 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
229     mailbox_size_limit = 0
230     recipient_delimiter = +
231     #inet_interfaces = loopback-only
232     inet_interfaces = all
233     default_transport = smtp
234     relay_transport = smtp
235     myorigin = /etc/mailname
236     inet_protocols = all
237     home_mailbox = Maildir/
238     mailbox_command =
239     smtpd_sasl_local_domain =
240     smtpd_sasl_auth_enable = yes
241     smtpd_sasl_security_options = noanonymous
242     broken_sasl_auth_clients = yes
243     smtpd_recipient_restrictions =
244     permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
245     smtp_tls_security_level = may
246     smtpd_tls_security_level = may
247     smtpd_tls_auth_only = no
248     smtpd_tls_note_starttls_offer = yes
249     smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem
250     smtpd_tls_loglevel = 1
251     smtpd_tls_received_header = yes
252     smtpd_tls_session_cache_timeout = 3600s
253     tls_random_source = dev:/dev/urandom
```

254 4.4 Openswan (VPN)

255 Openswan is an open-source IPsec VPN. Openswan runs on Linux and supports IKEv1, IKEv2,
256 X.509 Digital Certificates and NAT Traversal.

257 4.4.1 How It's Used

258 In the FS ITAM build, Openswan is used to form a secure VPN to the mainframe computer
259 owned by Vanguard Integrity Professionals.

260 4.4.2 Virtual Machine Configuration

261 The Openswan virtual machine is configured with two network interface cards, 8 GB of RAM
262 and one CPU core.

263 4.4.3 Network Configuration

264 The management network interface card is configured as follows:

265 IPv4 Manual

266 IPv6 Ignore/Disabled

267 IP Address: 172.16.0.67 (internal interface)

268 IP Address: 10.33.5.16 (external interface for the VPN)

269 Netmask: 255.255.255.0

270 Gateway: 10.33.5.1

271 DNS Servers: 8.8.8.8, 172.16.1.20, 172.16.1.21

272 Search Domains: lab5.nccoe.gov

273 4.4.4 Installing Openswan

274 Openswan is installed on a hardened Ubuntu 14.04 Linux system. Complete installation
275 instructions can be found at <https://www.openswan.org/>.

276 4.4.5 Installing Openswan

277 For Debian/Ubuntu Linux systems: It is always best to make sure your system is up-to-date by
278 performing:

279 `sudo apt-get update`

280 `sudo apt-get upgrade`

281 `sudo apt-get install openswan xl2tpd ppp lsof`

282 Copy the provided configuration files into `/etc`.

283 `cp <ipsec.conf> /etc`

284 `cp <ipsec.secrets> /etc`

285 Edit `/etc/ipsec.secrets` and replace **MYSECRET** with your pre-shared key.

286 Restart Openswan:

287 `service ipsec restart`

```
288     Verify by running:
289     service ipsec status
290
291     Bring up the IPsec tunnel:
292     ipsec auto -up nccoe-vanguard
293
294     Verify by running:
295     ipsec auto -verbose -status
296
297     If you see (ISAKMP SA established) then that is good.
298     A little script was created to keep the connection up - connect_vanguard.sh.
299
300     Copy connect_vanguard.sh somewhere typical like /usr/local/bin.
301     cp <connect_vanguard.sh> /usr/local/bin
302     chmod 755 /usr/local/bin/connect_vanguard.sh
303
304     Have it run every hour by linking it into cron.daily.
305     ln -s /usr/local/bin/connect_vanguard.sh /etc/cron.daily/connect_vanguard
```

302 4.4.6 Configurations and Scripts

```
303     /etc/ipsec.conf
304     # /etc/ipsec.conf - Openswan IPsec configuration file
305
306     # This file: /usr/share/doc/openswan/ipsec.conf-sample
307     #
308     # Manual:      ipsec.conf.5
309
310     # conforms to second version of ipsec.conf specification
311
312     # basic configuration
313     config setup
314         # Do not set debug options to debug configuration issues!
315         # plutodebug / klipsdebug = "all", "none" or a combination from below:
316         # "raw crypt parsing emitting control klips pfkey natt x509 dpd
317     private"
318         # eg:
319         # plutodebug="control parsing"
320         # Again: only enable plutodebug or klipsdebug when asked by a developer
321         #
322         # enable to get logs per-peer
323         # plutoopts="--perpeerlog"
324         #
325         # Enable core dumps (might require system changes, like ulimit -C)
326         # This is required for abrt to work properly
```

```

327         # Note: incorrect SELinux policies might prevent pluto writing the core
328         dumpdir=/var/run/pluto/
329         #
330         # NAT-TRAVERSAL support, see README.NAT-Traversal
331         nat_traversal=yes
332         # exclude networks used on server side by adding %v4:!a.b.c.0/24
333         # It seems that T-Mobile in the US and Rogers/Fido in Canada are
334         # using 25/8 as "private" address space on their 3G network.
335         # This range has not been announced via BGP (at least upto 2010-12-21)
336
337         virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4:172.16.0.0/12,%v4:25.0.0
338         .0/8,%v6:fd00::/8,%v6:fe80::/10
339         # OE is now off by default. Uncomment and change to on, to enable.
340         oe=off
341         # which IPsec stack to use. auto will try netkey, then klips then mast
342         #protostack=auto
343         protostack=netkey
344         # Use this to log to a file, or disable logging on embedded systems
345         (like openwrt)
346         #plutostderrlog=/dev/null
347         #plutodebug=all
348         plutostderrlog=/var/log/pluto.log
349         nat_traversal=yes
350         oe=off
351         #myid=172.16.0.66
352
353         # Add connections here
354
355         conn nccoe-vanguard
356             type=tunnel
357             forceencaps=yes
358             authby=secret
359             ike=3des-sha1;modp1024 #don't actually need to specify this
360             keyexchange=ike
361             ikelifetime=22800s
362             phase2=esp
363             phase2alg=aes256-sha1;modp1024
364             salifetime=3600s
365             pfs=yes #vanguard has pfs on
366             auto=start
367             keyingtries=3
368             #rekey=no
369
370             left=%defaulttroute
371             leftnexthop=%defaulttroute
372             leftsubnet=172.16.0.0/24 #NCCoE ITAM lab internal subnet
373

```

```
374         # either one of these seems to work
375         #leftid=10.33.5.16  #behind firewall ip address
376         leftid=136.160.255.42 #public ip address
377
378
379         #leftsourceip=136.160.255.42
380         leftsourceip=10.33.5.16
381
382         right=174.47.13.99  #IOS outside address
383         rightid=174.47.13.99  #IKE ID send by IOS
384         #rightsubnet is the internal subnet on the distant end
385         rightsubnet=172.17.212.0/24 #network behind IOS
386         rightnexthop=%defaultroute
387
388
389 /etc/ipsec.secrets
390
391 # This file holds shared secrets or RSA private keys for inter-Pluto
392 # authentication. See ipsec_pluto(8) manpage, and HTML documentation.
393
394 # RSA private key for this host, authenticating it to any other host
395 # which knows the public part. Suitable public keys, for ipsec.conf, DNS,
396 # or configuration of other implementations, can be extracted conveniently
397 # with "ipsec showhostkey".
398
399 # this file is managed with debconf and will contain the automatically created
400 RSA keys
401 # The %any %any line is just for testing
402 # Replace MYSECRET with your pre-shared key
403
404 include /var/lib/openswan/ipsec.secrets.inc
405 172.16.0.67 174.47.13.99 : PSK "MYSECRET"
406 10.33.5.16 174.47.13.99 : PSK "MYSECRET"
407 #%any %any : PSK "MYSECRET"
408
409
410 /usr/local/bin/connect_vanguard.sh
411
412 #!/bin/sh
413
414 #start IPsec tunnel
415 ipsec auto --up nccoe-vanguard
416
417 #status
418 #ipsec auto --verbose --status
```

4.5 Ubuntu Apt-Cacher

409 Ubuntu Apt-Cacher is a central repository for update and patch management used by all
410 Ubuntu systems on the network.
411

412 4.5.1 How It's Used

413 In the FS ITAM build, Ubuntu Apt-Cacher provides all Ubuntu systems with patches and
414 updates.

415 4.5.2 Virtual Machine Configuration

416 The Ubuntu Apt-Cacher virtual machine is configured with one network interface cards, 4 GB of
417 RAM and one CPU core.

418 4.5.3 Network Configuration

419 The management network interface card is configured as follows:

420 IPv4 Manual

421 IPv6 Ignore/Disabled

422 IP Address: 172.16.0.67

423 Netmask: 255.255.255.0

424 Gateway: 172.16.0.11

425 DNS Servers: 172.16.1.20, 172.16.1.21

426 Search Domains: lab5.nccoe.gov

427 4.5.4 Installing Ubuntu Apt-Cacher

428 Ubuntu Apt-Cacher is installed on a hardened Ubuntu 14.04 Linux system. Complete installation
429 instructions can be found at <https://help.ubuntu.com/community/Apt-Cacher-Server>.

430 For Debian/Ubuntu Linux systems: It is always best to make sure your system is up-to-date by
431 performing:

432 `sudo apt-get update`

433 `sudo apt-get upgrade`

434 `sudo apt-get install apt-cacher apache2`

435 Enable apt-cacher by editing `/etc/default/apt-cacher` and change **autostart** to **1**.

436 Restart Apache

437 `sudo /etc/init.d/apache2 restart`

438 Verify that things are working by pointing your Web browser to `http://<apt-cacher>:3142`

439 Edit `/etc/apt-cacher/apt-cacher.conf` and uncomment the following line:

440 `allowed_hosts = *`

441 Configure as a proxy to APT

442 `sudo nano /etc/apt/apt.conf.d/01proxy`

443 Inside your new file, add a line that says:

444 Acquire::http::Proxy "http://<IP address or hostname of the apt-cacher

445 server>:3142";

446 Restart apt-cacher:

447 sudo /etc/init.d/apt-cacher restart

448 4.5.5 Client Configuration

449 Client configuration is the same as setting up the server as a proxy to APT.

450 sudo nano /etc/apt/apt.conf.d/01proxy

451 Inside your new file, add a line that says:

452 Acquire::http::Proxy "http://172.16.0.77:3142";

453 4.6 Windows 2012 Certificate Authority

454 The Windows 2012 Certificate Authority server in the ITAM build uses an NCCoE base 2012 R2

455 x86_64 DoD STIG image. The installation of the Windows 2012 Certificate Authority server was

456 performed using installation media provided by DISA. This image was chosen because it is

457 standardized, hardened, and fully documented.

458 4.6.1 Software Configurations

459 Windows 2012 Certificate Authority (CA) server was designed to issue certificates to endpoints

460 that need to be accessed by users such that communication to such devices are deemed secure.

461 It is used in building a PKI system.

462 4.6.2 How It's Used

463 The ITAM solution uses the Windows 2012 CA server to issue certificates to endpoints that have

464 services that need to be accessed securely such as HTTPS enabled devices. The pfSense routers

465 utilized these certificates allowing for secure communication and configuration. The certificates

466 are also utilized by Splunk Enterprise and the Splunk Universal Forwarder.

467 **INSTALL ACTIVE DIRECTORY CERTIFICATE SERVICES (AD CS)**

- 468 1. Go to **Server Manager** and click **Add Roles and Features Wizard**.
- 469 2. Click **Next**. Select **Role-based or feature-based installation**. Click **Next**.
- 470 3. Select your server on the next screen and click **Next**.
- 471 4. Select the **Active Directory Certificate Services** and **Add Features** when prompted.
- 472 5. Click **Next** when you see .NET 4.5 framework and other default selections.
- 473 6. Click **Next** on informational screens.
- 474 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.

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.

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**.

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.

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.
 - Use any of the file transfer utilities such as SCP or FTP to copy the CSR to the CA.
 - Alternatively, the CSR can be copied via USB or other means.
8. Submit the Certificate Signing Request to the CA Service.
 - Log on to the CA server, go to the command prompt and type `Certreq.exe -attrib "CertificateTemplate:<Nameofthetemplate>" -submit <pathtoCSR>`
 - 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 >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 -outform pem -out  
PEM_CA_CERT.pem
```

where `DER_CA_CERT.crt` is DER encoded and `PEM_CA_CERT` is the transformed PEM encoded certificate

613 Additional information on converting certificates can be found at the following link
614 <http://info.ssl.com/article.aspx?id=12149>.

615 4.8 Process Improvement Achievers (PIA) Security 616 Evaluation

617 Process Improvement Achievers (PIA) conducted a remote security evaluation of the FS ITAM
618 build. The evaluation consisted of running multiple tools against the machines in the lab to find
619 any vulnerabilities due to misconfiguration.

Appendix A Acronyms

2	AD	Active Directory
3	CA	CA Technologies
4	CA	Certificate Authority
5	COTS	Commercial Off-The-Shelf
6	CRADA	Collaborative Research and Development Agreement
7	CSF	NIST Framework for Improving Critical Infrastructure Cybersecurity
8	CSR	Certificate Signing Request
9	.csv	Comma-Separated Value
10	DER	Distinguished Encoding Rules
11	DMZ	Demilitarized Zone
12	FS	Financial Sector
13	HR	Human Resources
14	ID	Identity
15	ITAM	Information Technology Asset Management
16	IDS	Intrusion Detection System
17	IP	Internet Protocol
18	NAS	Network Attached Storage
19	NCCoE	National Cybersecurity Center of Excellence
20	NIST	National Institute of Standards and Technology
21	OS	Operating System
22	PEM	Privacy Enhanced Mail
23	PKI	Public Key Infrastructure
24	SME	Subject Matter Expert
25	SQL	Structured Query Language
26	SSL	Secure Socket Layer
27	STIG	Security Technical Implementation Guideline
28	TLS	Transport Layer Security
29	VLAN	Virtual Local Area Network
30	VM	Virtual Machine
31	VPN	Virtual Private Network