Build From Scratch

LAB setup: Build from scratch

Prerequisites

- Host machine: 16 GB RAM minimum, 100 GB free disk recommended

1. vmware workstation pro
2. windows server 2016 iso
3. Oracle Linux 9 iso

Step 1: Create VMs

1. Install VMware Workstation Pro following the standard Windows application installation procedure.
2. Create 2 New virtual machines from ISO image in VMware Workstation Pro for Windows and Linux.
    a. During installation, set the network adapter setting to NAT. Add another network adapter and configure that as “bridged Network” for internet access from the Host subnet.
    b. Configure memory at least 4 GB and storage of 30 GB for each VM from the customize hardware option.
    c. Select the **split the virtual disk into multiple files**
    d. Configure a static IP address for both Windows and Oracle Linux in the same subnet. (such as: 192.168.200.0/24, VMware default for NAT network)

For windows:
IP address:192.168.200.10
subnet mask: 255.255.255.0
Gateway: 192.168.200.2

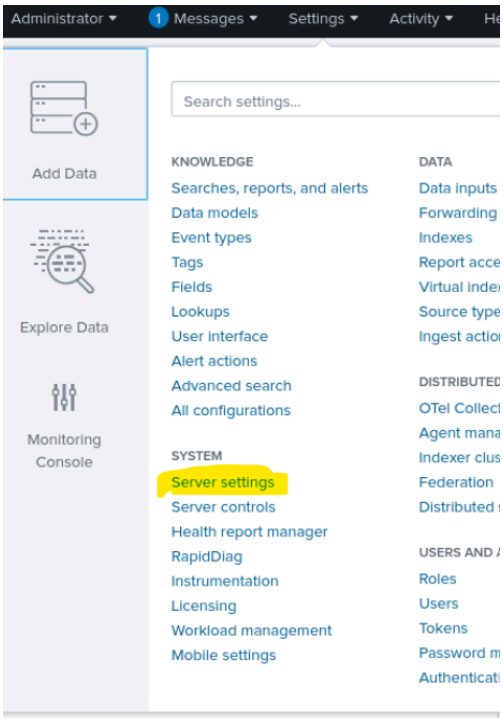
```
For Linux:
IP address:192.168.200.50
subnet mask: 255.255.255.0
Gateway: 192.168.200.2
```

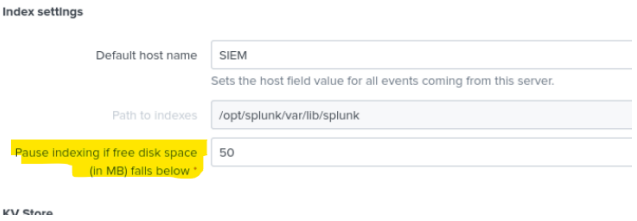
Step 2: Install & Configure Splunk in Oracle Linux VM

1. Download the trial version of Splunk Enterprise from URL below. https://www.splunk.com/en_us/download.html
2. Register with a valid business email account to download.

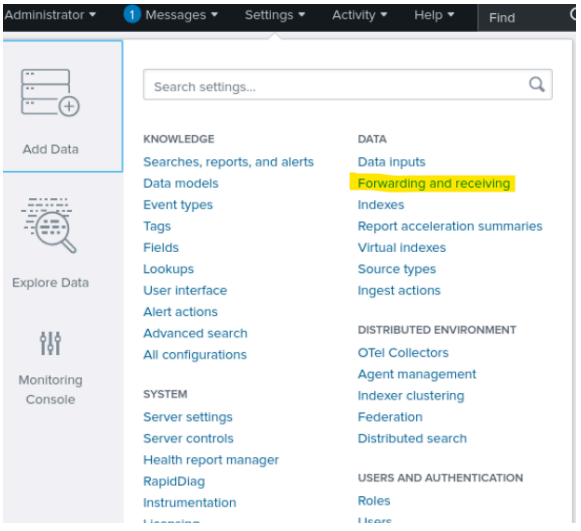
```
$ sudo su -
# cd /go/to/the/package/folder
# rpm -ivh splunk-9.0.2-17e00c557dc1-linux-2.6-x64.rpm
# /opt/splunk/bin/splunk start --accept-license --answer-yes --no-prompt
--seed-passwd splunkadmin

---comment---
Here --accept-license switch will skip eula,--no-prompt will disable interac
tive mode during installation,
and --seed-passwd will set the admin password as "splunkadmin" which w
ill be used later to login splunk web.
----comment---

#./splunk enable boot-start

--comment---
when server reboots, splunk will start automatically.
---comment---
```

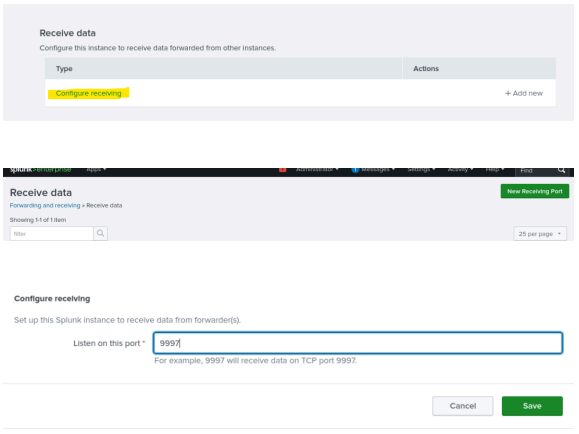
3. Go to Splunk web GUI at https://192.168.200.50:8000. Login with user: admin pass: splunkadmin
4. Go to settings > server settings > General settings
5. Change the Index settings as below.



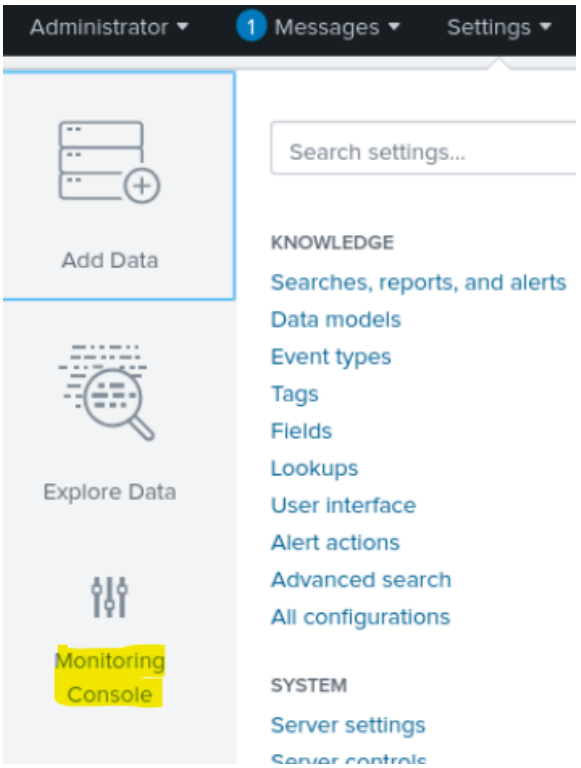


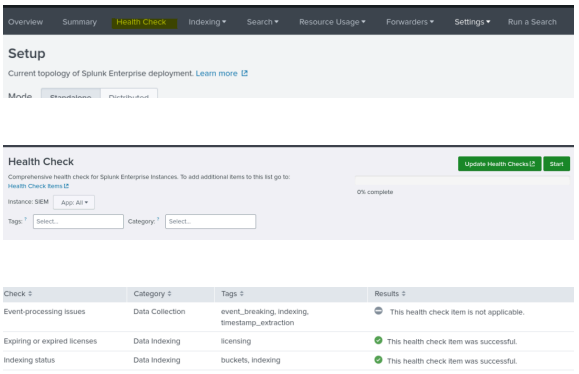
6. To enable ingesting logs, go to settings> forwarding and receiving> configure receiving> New receiving port > Listen on this port * > write 9997 > click on save





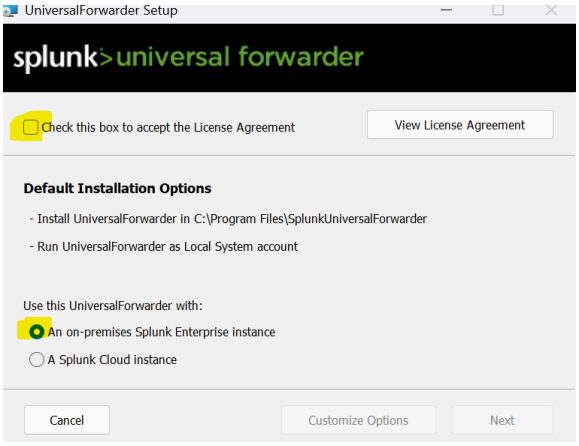
7. For best practice, go to settings > Monitoring Console > Health Check > Start. Splunk will start checking the health status of Splunk. Ensure none of the checks throw any critical (Red) errors.

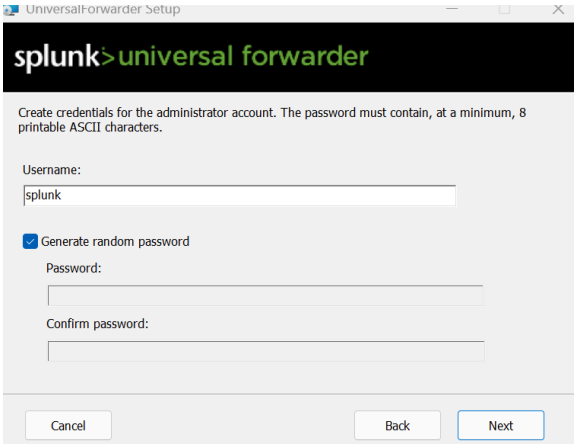


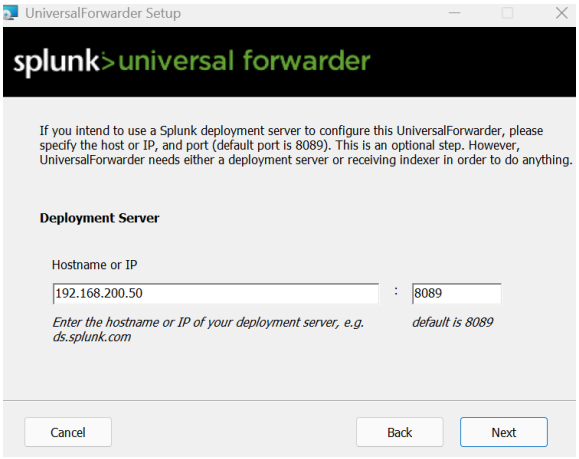


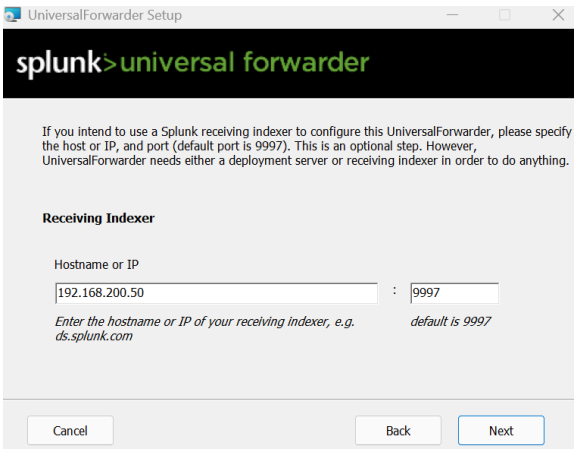
Step 3: Install and configure Splunk UFW/agent in Windows VM

1. Download the trial version of Splunk Universal Forwarder (UFW) from URL below.
   https://www.splunk.com/en_us/download.html
2. Register with a valid business email account to download.
3. Install Splunk UFW .msi following the standard Windows application installation procedure.
4. While installing UFW, ensure below configuration.









Step 4: Configure Atomic Red Team in Windows VM

You must have internet connectivity to execute the PowerShell download cradle. Make sure to run as Administrator.

Before running the atomic red team installation script, disable the Windows Defender completely.

```
PS C:\Users\pmics> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
PS C:\Users\pmics> IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/install-atomicredteam.ps1' -UseBasicParsing);
PS C:\Users\pmics> Install-AtomicRedTeam -getAtomics
```

Step 5: Configure log forwarding in Windows VM

1. Create Splunk_TA_windows>local folder as below. Then create new file inputs.conf on Windows host to send the logs.
   `C:\Program Files\SplunkUniversalForwarder\etc\apps\Splunk_TA_windows\local\inputs.conf`
2. Insert below configuration stanza into input.conf

```
[WinEventLog://Security]
sourcetype = WinEventLog:Security
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
index = windows
renderXml = false

[WinEventLog://Windows PowerShell]
sourcetype = WinEventLog:Powershell
#sourcetype=win:powershell
disabled=0
index = windows
renderXml = false
```

3. Enable necessary audit logs to detect attacks.

Step	Action	Tool
1	Enable global Audit Object Access	secpol.msc or auditpol
2	Enable Audit Registry sub-policy	gpedit.msc / auditpol /set
3	Set auditing on specific registry key	regedit → Permissions → Auditing
4	Verify events	Event Viewer → Security (4657, 4663)
5	Enterprise-level policy	GPO (gpmc.msc)

4. Restart UF

```
& "C:\Program Files\SplunkUniversalForwarder\bin\splunk.exe" restart
```

Step 6: Attack simulation in Windows VM

1. Run the command below from a Windows Server PowerShell session. Ensure to run this as admin each time a new PowerShell session is invoked.

```
Import-Module "C:\AtomicRedTeam\invoke-atomicredteam\Invoke-AtomicRedTeam.psd1" -Force
```

2. Run the attack simulation for brute force. First, get the prerequisites, if any.

```
Invoke-AtomicTest T1110.001 -GetPrereqs
```

3. Run the attack

```
Invoke-AtomicTest T1110.001
```

4. Run another attack simulation for registry key modification. First, get the prerequisites, if any.

```
Invoke-AtomicTest T1547.008 -GetPrereq
```

5. Run the attack

```
Invoke-AtomicTest T1547.008 -Force -Verbose
```

6. revert to the original state

```
Invoke-AtomicTest T1110.001 -Cleanup
Invoke-AtomicTest T1547.008 -Cleanup
```

Step 7: Attack detection in Splunk

1. Login to the splunk Search Head from oracle linux or windows VM.
   URL: https://192.168.200.50:8000
2. Go to Splunk app > search and reporting
3. Detect the alerts by the provided use case.

For BruteForce

```
index=windows
| search EventCode=4625
| search NOT Account_Name=*$
| bin _time span=1m
| stats count as failed_attempts by Source_Network_Address, Account_Name, _time
| where failed_attempts >= 5
| table Source_Network_Address, Account_Name, failed_attempts, _time
| sort - failed_attempts
```

For Registry modification

```
index=windows | search EventCode=4657
| rename Account_Name as User, ComputerName as Host, Object_Name as Registry_Path, Object_Value_Name as Modified_Object
| where NOT match(User, "(?i)^(SYSTEM|LOCAL SERVICE|NETWORK SERVICE)$")
| where NOT match(User, ".*\\$$")
| table _time, User, Host, Registry_Path, Modified_Object, Process_Name
```