ISEMan: A management and deployment interface for lab-based activities within ISERink

Alex Luehm
Iowa State University, luehm@iastate.edu

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Other Computer Engineering Commons

Recommended Citation
Luehm, Alex, "ISEMan: A management and deployment interface for lab-based activities within ISERink" (2018). Creative Components. 7.
https://lib.dr.iastate.edu/creativecomponents/7

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
ISEMan: A management and deployment interface for lab-based activities within ISERink

by

Alex Wolfgang-Nagel Luehm

A Creative Component submitted to the graduate faculty in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Electrical and Computer Engineering
Information Assurance

Program of Study Committee:
Doug Jacobson, Major Professor

Iowa State University
Ames, Iowa
2018

Copyright © Alex Wolfgang-Nagel Luehm, 2018. All rights reserved.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Terminology</td>
<td>2</td>
</tr>
<tr>
<td>2.1 vSphere</td>
<td>2</td>
</tr>
<tr>
<td>2.2 ISERink</td>
<td>3</td>
</tr>
<tr>
<td>2.3 ISEHost</td>
<td>4</td>
</tr>
<tr>
<td>2.4 ISELab</td>
<td>5</td>
</tr>
<tr>
<td>3. Motivation</td>
<td>7</td>
</tr>
<tr>
<td>4. Management Structure</td>
<td>8</td>
</tr>
<tr>
<td>4.1 Virtual Lab Structure</td>
<td>9</td>
</tr>
<tr>
<td>4.1.1 Course Structure</td>
<td>9</td>
</tr>
<tr>
<td>4.1.2 User Structures</td>
<td>10</td>
</tr>
<tr>
<td>4.1.3 Networking Structure</td>
<td>11</td>
</tr>
<tr>
<td>4.1.4 Machine Deployment</td>
<td>13</td>
</tr>
<tr>
<td>4.2 IP Assignment</td>
<td>14</td>
</tr>
<tr>
<td>4.3 VM Configuration</td>
<td>17</td>
</tr>
<tr>
<td>5. Using ISEMan</td>
<td>20</td>
</tr>
<tr>
<td>5.1 ISEMan Setup</td>
<td>20</td>
</tr>
<tr>
<td>5.2 Roster setup and use</td>
<td>21</td>
</tr>
<tr>
<td>5.3 Object deployment</td>
<td>24</td>
</tr>
<tr>
<td>5.4 IP Assignment</td>
<td>27</td>
</tr>
<tr>
<td>5.5 Ansible</td>
<td>29</td>
</tr>
</tbody>
</table>
6. Future work ................................................................. 30

6.1 DHCP boot strapping .................................................. 30

6.2 Proxied access to vSphere ......................................... 30

6.3 AD management ....................................................... 30

6.4 Persistence across courses ......................................... 31

7. Conclusion ................................................................. 32
Abstract

ISERink is an isolated virtual environment built within the vSphere virtualization platform in which users can safely perform various cyber security exercises without fear of damaging real-world machines. In the past it has been successfully used in cyber defense competitions to provide a network setting similar to that of the real Internet, while safely containing any rogue malicious traffic. Recently the ISERink environment has been deployed within the academic setting alongside ISELab to provide students with a safe and controlled environment in which to practice building, securing, and attacking networks in a structured lab setting in conjunction with guided lectures.

While ISERink has been of significant assistance in teaching a wide range of security concepts, there exists considerable overhead in the initial ISELab setup and in the deployment of student machines and network resources. As a result, students are exposed to concepts outside the scope of a typical security course and spend more time dissecting the nuances of vSphere and ISERink than on actual lab material. In addition, the deployment of template machines by TAs and instructors requires extensive forethought and still requires further configuration by the student users.

ISEMan provides an easy and intuitive way to configure a vSphere datacenter to interface with an existing ISERink and to manage, deploy, and configure templated student machines, networks, and network services.
1. Introduction

ISEMan is a django web application that has been integrated with the pyVmomni Python vSphere API and the Ansible configuration deployment engine to provide instructors and TAs with the ability to rapidly create and deploy virtual lab environments configured to operate with ISERink. Templates are able to be deployed across entire courses with networking and operating system settings preconfigured and ready to use. ISEMan is meant to be easily configured and able to adapt to future changes in the ISELab and ISERink infrastructures, should any arise.

To start, a brief discussion concerning terminology used throughout this document is provided in section 2, while the motivation behind ISEMan is presented in section 3. Next, section 4 describes the lab architecture used by ISEMan in order to achieve the desired lab environment. Section 5 provides step-by-step instructions for creating and deploying a new lab environment, deploying new VMs, and performing post-deployment configuration.

Finally, section 6 provides the reader with future goals and uses for the ISEMan project with the summary and any concluding statements presented within section 7.
2. Terminology

In order to provide the reader with enough background knowledge to comprehend the later sections dedicated to the motivation, design, and use of ISEMan, the following terms are defined and provided with descriptions of how each component relates to ISEMan.

2.1 vSphere

vSphere is a cloud computing platform that is used to unify instances of the VMware ESXi hypervisor. While instances of ESXi can be run as stand alone machines vSphere allows network storage to be configured and shared among the individual ESXi hosts, along with a centralized interface for configuring host networking and for virtual machine deployment and management.

Each individual ESXi host is responsible for providing computing and virtual networking for the locally deployed virtual machines. Hosted VMs can have the underlying disk images housed anywhere within vSphere while resources such as CPUs and RAM are provided by the host ESXi machine.

From an organizational perspective, VMs are housed within resource pools on their ESXi host which are in turn used to dictate the maximum resource resources allowed for the contained VMs. Additionally, VMs are also assigned a folder within the aggregated vSphere environment that allow VMs to be logically grouped and to allow permissions to be granted to vSphere users in an organized manner.

Users access VMs via virtual consoles through the vSphere graphical user interface (a Flash or HTML5 web applications). User authentication and group definitions are managed via an external system such as Active Directory, while the permissions associated with each user account or user group are dictated within the vSphere environment through predetermined User/Group roles.
Networking is managed through vSwitches (virtual switches) defined within each individual ESXi host. Each vSwitch can contain an arbitrary number of port groups which in turn can be assigned unique VLAN ids in order to provide unified or segmented virtual networks which are then used by provisioned the VMs. vSwitches can either be bridged to one or more physical NICs located on the ESXi hosts or kept isolated from external networks to provide private user networks.

### 2.2 ISERink

An ISERink is collection of virtual machines hosted within a standalone ESXi running the ISEAGE software which works to mimic the behavior and appearance of the Internet within an air gaped setting. A collection of machines called "Boards" each emulate a subset of IP ranges by rewriting and queuing packets as they are "routed" from Board to Board before arriving at their final destination, whether it be to an internal "mimicked" or external "real" IP. Each Board provides what is known as a "Blue" network which is used by machines located outside the ISERink to gain network access.

Other machines within ISERink provide services such as DNS and proxied internet access to the outside world. Internal DNS services can be used to provide students with domains with which to experiment and to provide name resolution for internal network services such as e-mail and http. Isolation from the outside world is provided by two "Keyhole" machines with reverse facing firewalls. These prevent any unauthorized internal traffic from reaching the external internet and prevent all external traffic from reaching the internal network. Access to the outside world is allowed via a Squid proxy which allows only traffic on ports 21 (FTP), 80 (HTTP), and 443 (HTTPS) to be forwarded. All other traffic is treated as internal-only traffic.

An ISERink typically has three Blue networks, each with 15 IP ranges, however the number of Blue networks and IP ranges assigned to each can be modified to suite the needs of each class. These modifications are entirely supported within ISEMan, but are regarded as outside the scope of what ISEMan is able to automatically configure. ISEMan is meant to integrate with a preconfigured ISERink.
Within the context of this document, an ISERink is housed on an independent ESXi node outside the vSphere instance running the ISELab. Multiple ISERinks may be connected to an ISELab at any time, and indeed is the assumed scenario as it is expected that each course running within an ISELab will have its own independent ISERink. Each ISERink that is in use will then have one physical connection per Blue network in the ISELab (as shown in figure 2.1).

![Figure 2.1 ISERinks and ISELabs](image.jpg)

### 2.3 ISEHost

An ISEHost is an ESXi node within vSphere that is used by ISELab to host VMs used by students and networking services. Together, groups of ISEHosts provide computational resources to the VMs deployed within a specific course. ISEHosts are only assigned to one course and are typically grouped together within a folder (figure 2.2).

![Figure 2.2 ISEHosts assigned to courses](image.jpg)
Each ISEHost provides VMs with a "Blue" network (vSphere port group) with an uplink to a corresponding network within an external ISERink (as in figure 2.1), multiple internal-only networks (unique VLAN ids) that are used to provide students with private networks for use behind optional firewalls, and a "TAP" network that performs port-mirroring for the local "Blue" network (figure 2.3).

![ISEHost VMs and networks](image)

Figure 2.3  ISEHost VMs and networks

### 2.4 ISELab

An ISELab is a vSphere center that is used to collectively host and manage student virtual machines and networks across multiple courses. Courses within an ISELab are allocated a number of ESXi hosts (ISEHosts, as in figure 2.1) which are used to host VMs for the specified course. Each VM is assigned a unique IP address that is hosted within the ISERink. Network traffic is passed from each ISEHost via a physical network connection to an ISERink where it is then routed to either the appropriate virtual destination or through a Keyhole air gap to the outside world.

As mentioned above, user authentication and group definitions are provided via an external AD while user and group roles are defined within ISELab. ISEMan requires that AD users and groups be configured before use. Two user groups are assumed to exist, one for student users and one for
administrative users. In addition, a group role with permissions for basic network usage is assumed to have been created within ISELab.
3. Motivation

ISERink has been successfully used in previous years to provide students in cyber security courses with a controlled environment in which to conduct experiments and gain hands-on experience with offensive and defensive security tools. Each student was assigned an IP range from within an ISERink and would use this range for any lab experiments.

During the initial years of use, students were responsible for building and configuring machines in ISELab by hand, needing to install VMs from installation images, configure network settings and download, install, and configure any other software required for use in lab. Only then could the actual lab assignments be carried out. While this provided valuable knowledge and practice to users concerning virtualization and networking configuration, it technically fell out of scope for many security courses and tended to overwhelm new and aspiring students. Students would end up spending more time configuring machines than on actual lab work.

In later years, preconfigured template images were generated and shared with students where all required packages had been installed and configured beforehand. While this alleviated issues in configuration errors, students still tended to have issues when configuring any static IPs and ensuring that the correct network connections were established within ISELab. In addition, students would tend to deploy templates simultaneously at the beginning of lab, causing the ISELab to become overwhelmed and sometimes cause corrupted deployments.

Thus stems the desire to be able to systematically deploy templates with properly configured networking settings before lab time and the ability to remotely push configuration updates for later labs. These issues are all handled with the use of ISEMan.
4. Management Structure

After observing consistent issues with machine configuration and deployment, the following deployment, networking, and configuration schemes have been developed to assist in lab creation and are the driving factors behind ISEMan. These schemes are presented below for greater understanding, but are automatically performed by ISEMan - little to no direct configuration need be carried out on ISELab.

In order to provide a concrete example, the demonstration course "cpreTEST" is considered in the following discussion. It contains the ISEHosts "isehost03" and "isehost08", two course administrators, "Julie Rursch" and "Alex Luehm", with one student user "Nic Losby". "isehost03" is connected to the Blue1 network in an ISERink on physical nic "vmnic4", while "isehost08" is connected to the Blue2 network in the same ISERink on nic "vmnic4". An Ubuntu 16.04 desktop is to be deployed to each user.

It is assumed that an external authentication system such as Active Directory is used and has been integrated into ISELab. In keeping with the aforementioned scenario, student users are grouped into the "cpreTEST" group, while course administrators are within the "cpreTEST-ta" group. ISELab has been configured such that the user role "Class_Network_User" exists with the "Assign network" role, as outlined in figure 4.1.

![Class_Network_User role](image)

Figure 4.1 Class_Network_User role
4.1 Virtual Lab Structure

There are four phases in the deployment of resources required to properly configure a virtual lab and to begin provisioning of VMs. First, a management framework for VM management and propagating permissions must be deployed for each course. Next, student resources and permissions must be deployed and configured. Afterwards, port groups and VLANs must be deployed and configured for use with VMs, and lastly machine templates may be generated and then deployed.

The next subsections describe the results of each phase.

4.1.1 Course Structure

Firstly, resources within vSphere must be created and configured such that objects generated later inherit the correct permissions and can be easily located. Each course contains a single top-level vSphere folder with five child folders used to store VMs and templates: admins, hidden, meta, students, and templates (figure 4.2).

![Figure 4.2 vSphere course folder structure](image)

Course administrators and TAs (AD group "cpreTEST_ta") have propagating administrative permissions on this root folder, providing them with full administrative access to any course-related templates, network services (meta), and student machines (figure 4.3).

![Figure 4.3 Course administrator propagating permissions](image)

Course administrator machines are housed within the "admins" folder for ease of access while student machines are stored within either the "students" or "hidden" folders. The "hidden" folder is used to store machines that are designated for a particular student, but which should not be accessible via a virtual console (such as in a network enumeration lab).
Machines to be used as templates in automated deployments are housed within "templates" which only administrators have access to. Finally, the "meta" folder contains special one-off machines used for basic network services such as DHCP, DNS, mail, packet archiving, etc...

On each ISEHost assigned to a course, a similar structure is used with resource pools to similarly organize VMs and assign permissions (figure 4.4). The noted exception is that templates are only stored on one ISEHost - the Primary ISEHost - for ease of location. Again, the root resource pool is assigned the Administrative role for the course administrator group.

![ISEHost resource pool structure](figure 4.4)

### 4.1.2 User Structures

After course resources have been deployed, users are assigned an arbitrary ISEHost such that each ISEHost receives a distributed number of students and the required folders and resource pools are deployed within the "students" or "admin" directory/resource pools (figure 4.5).

![User resources deployment](figure 4.5)

At this point it is important to notice that these structures have implicit administer permissions granted for the corresponding user (figure 4.6). These permissions are in addition to those propagated from the root course structures. These implicit permissions allow users to perform localized administrative actions if required.
4.1.3 Networking Structure

Network deployment occurs both physically and virtually. Before the appropriate vSwitches and port groups can be created, the physical nic connecting each ISEHost to its corresponding Blue network in an ISERink must be determined (reference figure 2.1). It is here assumed that through physical observation, both "isehost03" and "isehost08" are connected to their corresponding Blue network within an ISERink on their respective "vmnic4" interfaces.

To provide access to the ISERink network, each ISEHost is configured with a standard vSwitch "vSwitch1" that is bridged to the corresponding physical nic (figure 4.7).

Within this vSwitch, two port groups are configured for machines hosted on the local ISEHost to connect to: "cpreTEST_blue1" and "cpreTEST_blue1_tap" (figure 4.8). The former is a standard port group and provides standard switching capabilities, allowing machines to communicate with the ISERink.
The later serves as a local network TAP by mirroring any traffic being sent across the switch. This is accomplished by enabling "promiscuous mode" on the group (figure 4.9). It should also be noted that the two port groups on "vSwitch1" have no VLAN ids set (group 0).

![Figure 4.9 isehost03 blue1 tap settings](image)

A secondary vSwitch called "Internal" is created and not bridged with an external nic. For each user, a port group with a unique VLAN id is created and thus can be used as an internal network in situations that require NATing or a network-based firewall. These port groups have the "Class Network User" role applied to each corresponding user account. This allows networks to be manually switched on VMs.

![Figure 4.10 iseh0st3 Internal networks](image)

Port groups are organized within vSphere folders such that propagating permissions similar to those used for VMs can be applied as needed. A root folder "cpreTEST" is created with two child folders "cpreTEST_external" and "cpreTEST_internal". The "cpreTEST_blue" and "cpreTEST_blue_tap" networks for each ISEHost are housed within the "cpreTEST_external" folder, while private user groups created on the "Internal" vSwitch are kept within the "cpreTEST_internal" folder (figure 4.11).

The root folder "cpreTEST" is assigned the propagated administrative role for the "cpreTEST_ta" group, and the child folder "cpreTEST_external" is assigned the "Class Network User" role with
the "cpreTEST" group such that standard users have the ability to access the TAP and Blue networks as well.

4.1.4 Machine Deployment

After the management infrastructures at the course and user levels have been deployed and after the required networking structures have been properly created and configured, templated machines can be deployed to users.

Machines are deployed via a linked-clone in order to save on disk space with minimal degradation of performance. A template machine, or base image, must first be configured as desired by the course administrators by installing any required software and performing any custom configuration. After the template machine has been tested and verified for correct configuration, a snapshot is taken from within vSphere, which will be used as a reference image during deployment.

These template machines are located within the "template" vSphere folder and the primary ISEHost’s "template" resource pool. The naming structure \texttt{<course>_<template>_<name>} (figure 4.12) is used to distinguish these VMs from ordinary VMs and are not used for any lab activities. Their sole purpose is to generate subsequent machines to be used by course users. By allowing templates to persist across semesters and being updated as additional resources are required, they can be used to provide consistent machines over a long period of time.

Deployed machines are located within the "students" or "admin" folder and resource pool on the corresponding ISEHost and follow the naming convention \texttt{<course>_<netID>_<name>} (figure
Due to the previously configured permissions, newly created VMs should automatically inherit course administrator and standard user permissions.

Of course VMs can be deployed by vSphere templates and provisioned by student users, or created from scratch by student users - the course structures, user structures and networking configuration would not be required to change to accommodate this, but would not fall under the management of ISEMan.

### 4.2 IP Assignment

In the past, network configuration has always been an issue as IPs typically must be statically assigned to ensure that each user has machines that have IPs within their designated range. Students would be responsible for manually configuring the static IP and proxy on each machine, and with the wide variation in operating systems and previous networking/Linux experience, there inevitably emerges groups of students spending alarming amounts of time over trivial configura-
tion issues that lay outside the scope of lab. Even deployed machines would need to be manually configured, as each VM would end up preconfigured with the same IP address.

Fortunately this issue is solved with the help of the Dynamic Host Configuration Protocol. DHCP servers allow network devices to obtain an IP assignment from a pool of predefined IP addresses through negotiation with a network server. This is done during VM boot, allowing machines to be deployed and then later automatically receive an IP assignment. These servers offer the ability to statically map interface MAC addresses to specific IPs, providing a persistent one-to-one binding. This, coupled with dynamic MAC address specification upon VM deployment, allow machines to be deployed and automatically configured to a predefined address for each student.

In order to contain any ISELab-specific configuration within the ISELab setting and to reduce needless traffic within ISERink, DHCP servers are hosted within the ISELab, one per ISEHost. Each server is responsible for the IP ranges provided by the corresponding Blue network within the ISERink. In this manner, all DHCP traffic is constrained to the physical ISEHost. In accordance with the course resources deployment scheme, DHCP servers are located within the ”meta” folder, and within the ”meta” resource group on each ISEHost (figure 4.14).

![figure 4.14 DHCP servers configured per onst](image)

Care must be taken during the DHCP server configuration, as by default a DHCP server only allows IP leases from one IP range to be issued per NIC. This implies that a machine providing IP leases for \( n \) number of IP ranges would require \( n \) NICs, each configured with its own static IP address.

While more NICs could be added during VM creation virtual NICs are used instead, allowing one NIC to respond to requests for multiple IP ranges. Using a network interface configuration
similar to that in figure 4.15, only one NIC is required per DHCP server. One "true" entry is made, giving the machine a true primary IP assignment with a defined gateway to be used for server-generated traffic while each subsequent virtual interface is configured with an IP from each managed IP range with which to respond to incoming DHCP requests.

```
auto lo
iface lo inet loopback

auto ens160
iface ens160 inet static
  address 64.39.3.250
  netmask 255.255.255.0
  gateway 64.39.3.254
  dns-nameservers 199.100.16.100

auto ens160:0
iface ens160:0 inet static
  address 64.39.3.253
  netmask 255.255.255.0

auto ens160:1
iface ens160:1 inet static
  address 33.96.5.253
  netmask 255.255.255.0

...  
```

Figure 4.15 virtual interfaces virtual configuration

Once the virtual interfaces are configured, the actual MAC-IP mappings can be formed within the DHCP server configuration file. In this instance, it is assumed that the `isc-dhcp-server` application from the Ubuntu package manager is used. As can be seen in figure 4.16, a common DNS server can be configured for all subnets by wrapping the DNS option and all subsequent subnets within a `shared-network` space. Each managed IP range requires a separate `subnet` section that specifies the network range to respond to and an upstream gateway. Within each `group` section, individual hosts can be assigned static IP based upon MAC address. These mappings are updated before each template deployment.

It should be noted that two IP ranges are intended to be reserved, one used for network services such as e-mail, web sites, and DNS hosting, and the second for hosting the "YellowSnow" collection of vulnerable operating systems.
Virtual machine configuration is done either before deployment, meaning modifications and configurations to the underlying machine exist within template images that are deployed via linked-clones, or after deployment via remote configuration. It is favorable that most configuration be done during the template creation stage as it is easier to perform testing and validation to ensure a properly configured VM. However later configuration can be performed after deployment through the use of Ansible.

Ansible is a simple yet powerful deployment engine that uses OpenSSH and Python to remotely call arbitrary processes or dynamically generate configuration files. Lists of steps to perform are contained in YAML-structured files called "Playbooks" (figure 4.17), which are compiled into Python byte-code that is copied to the remote machine via SSH and then locally executed. This allows for complex procedures to be easily written, executed, and then updated for future use.
Figure 4.17 Ansible Playbook for initial config

Machines to be controlled by Ansible are listed within a `hosts` file, where they can be grouped for easy reference (figure 4.18). The `hosts` file is updated at the same time as the DHCP server config files to ensure that newly deployed machines are immediately controllable. The IPs for every machine deployed from a single template are grouped together allowing all templates to be configured simultaneously from the same Playbook.

Figure 4.18 Ansible hosts file

To provide Ansible remote access, template machines are preconfigured with a master RSA public key on the root account while keeping the private key on a control server (the machine
running Ansible). This allows Ansible to remotely access machines without the need for a local root password, and without providing students with the ability to infiltrate one-another’s VMs.

The development of Ansible Playbooks is outside the scope of ISEMan, however Playbooks can be used within ISEMan to perform VM updates, push out package installs or required binaries for use in class, or to be used to configure network resources such as e-mail or DNS servers.
5. Using ISEMan

ISEMan takes the aforementioned management scheme and automates the implementation of said structure to allow consistent and reliable course, user, and machine deployment. Two modules are used - Roster for course, user, template, and machine information management and deployment, and Ansible, which allows "Playbooks" to be ran against deployed machines to provide further configuration.

The following sections provide details for ISEMan setup, configuration, and use.

5.1 ISEMan Setup

As mentioned above, ISEMan is partitioned into two parts - the Roster, which is intended to be ran from outside the ISELab environment (or via an internal machine with non-proxied internet access) and Ansible, which is required to be ran from within ISELab. These constraints are on the underlying network access provided by the hosting machine and so does not affect setup.

The only requirements for ISEMan are Python3 (including virtualenv, plumbum and pip3), Ansible, and a Linux desktop, so most vanilla Linux desktops can be used (Ubuntu16.04 Desktop is used in the example images below). It is assumed that all required ISERinks are already operational and properly connected to the ISEHosts designated for each course. It is also assumed that the external user management service has been configured and that ISELab users have already been added and segregated into either a student or course administrators group.

Steps that follow are for setting up the django development server. Clone the repository from https://git.ece.iastate.edu/infaslab/iseman into a local directory and within the "iseman" directory create a new Python virtual environment. Next, "activate" the environment so that any dependent python packages will be isolated to the local project (figure 5.1). vEnv activation is required in every new terminal. The first time running, the aforementioned Python packages will
need to be installed. This can be done via pip and the "requirements.txt" file. Once this has been done, change directory into the "app" directory and start the local server to start ISEMan.

```
cd iseman
virtualenv vEnv
source vEnv/bash/activate
pip -r requirements.txt
cd app
python3 manage.py runserver
```

Figure 5.1 Initial steps

Note that DHCP servers are assumed to be pre-deployed and have the iseman.rsa key (found in the "app" directory) installed under the root account as well (within the authorized_keys file). Depending on the number of IPs to be supported, each DHCP machine must either have the prerequisite number of NICs, or have one nic with the needed virtual interfaces configured (see ifconfig_gen.py within the "app" directory).

5.2 Roster setup and use

Once ISEMan is running is is time to create a new course. In a web browser, navigate to localhost:8000 where you will be greeted with the ISEMan homepage (figure 5.2). Select "Roster" to be taken to the Course listing page.

Figure 5.2 ISEMan homepage

From here, click the link "Create a new Course". You should be presented with a form to provide ISEMan with information about the vSphere and AD environments (figure 5.3). An account and password with administrative privileges in vSphere should be provided. The student AD group
should have all students enrolled in the course, with TAs and course administrators in the TA group.

![ISEMan new course form](image1)

Figure 5.3 ISEMan new course form

![Courses](image2)

Figure 5.4 ISEMan new course

A new course should now be listed on the Courses page (figure 5.5). Click on the course name to proceed to the course summary page. As you can see, not very much can be done until ISEHosts have been added. Click on the "New ISEHost" button to add an ISEHost to the course.
Provide information concerning the new ISEHost to the form and select "Save". The vSwitch provided for the "External switch" entry should be bridged to the ISERink via physical cabling, while the "Internal switch" should have no physical nic - this switch will be used for private user networks. Since this is the first ISEHost to be added, mark this as "primary" (figure 5.6).

You should be redirected to the newly added ISEHost’s summary page (figure 5.7). Notice that no IPs are available for this ISEHost. Click on the "Associated IPs” link to be brought to the IPs summary page. From here it is possible to inform ISEMan which IPs are hosted by the corresponding Blue network in ISERink. Either utilize the manual IP specification or bulk upload feature to associate IPs with this ISEHost (figure 5.8).

Repeat this process for as many ISEHosts as are planned to be configured for this course. Notice that the first IP added is marked as not available (figure 5.8. This is because ISEMan reserves one IP address from the entire course to be used for network services.

Now that IPs have been made available, users can be added. Navigate to the Course summary page and use either of the "New user" buttons to either manually add users, or to add users in bulk (similar to the IPs). Please note that if both the "Base IP” and "ISEHost” fields are left blank
on the manual form (figure 5.10), the new user will be assigned an ISEHost and IP such that the number of users throughout the various ISEHosts is well distributed.

Repeat to add users as needed.

5.3 Object deployment

Once sufficient users and ISEHosts have been added, the course management structures can be created within vSphere. This will create any required folder, resource pools, and networks/port groups while applying the permissions required for proper user access. This is done in a non-destructive manner, allowing existing infrastructure to be reused.

To generate and deploy the infrastructure for the course, click on the "Deploy course" button towards the bottom of the Course Summary page. The page may become unresponsive while the operation takes place (a spinner should pop up showing that the page is busy), but do not re-click the button or close the page. Once the structures have been generated in vSphere, the page will reload. The spinner should stop spinning.
Verify that the desired structures have been generated within vSphere via the web interface. The structure should be similar to that which was described within the "Management Structure" section (reference figures 4.2, 4.4, and 4.11).

Also on the Course Summary page, use the "Deploy all users" (figure 5.12) to deploy user folders, resource groups, and private networks to vSphere. Again, confirm via the web interface (reference figure 4.5).
Machines can now be created (outside of ISEMan) within the "templates" directory, configuring required packages and operating system settings (proxy) as needed. Once a machine is ready to be deployed, navigate to the Course Summary page in ISEMan and click the "Refresh templates" button (figure 5.13). This will scan the "templates" folder in vSphere and populate any found templates in ISEMan.

Templates that have been populated within ISEMan can be deployed in what is called a "Deployment" to selected users at a specified IP address. Deployments store a snapshot of the deployed machine, allowing it to be redeployed at a later time, even if the original template has been modified (but not deleted).

From the list of templates, select a desired template and click "New deployment" (figure 5.14). A new page will appear, where information concerning the new deployment can be provided. Here,
the user can provide at what IP the new machine should be created at, what name the new machine should have (will be appended to the string "course\_netid" when deployed to users), and whether the new machine should be placed on the blue network (ISERink), or on the user’s private internal network (such as in a firewalled situation).

After providing the required information, click the "Deploy to selected" button (figure 5.15) to have the machine replicated and deployed to each of the selected users. The operation is done via linked-clones, however it could still take a considerable amount of time - do not navigate away from the page.

Once it has finished, notice that there is a new Deployment entry on the Course Summary page. This deployment can now be redeployed to users previously without the machine, even after the template has been modified. Redeployments utilize automated snapshots to ensure that consistency can be ensured across redeployments. The template can be used to create a new deployment (if further configuration is required) without fear of its previous variants being lost.

### 5.4 IP Assignment

Now that machines have been deployed, the final step is to configure the DHCP servers. This is done by downloading a zip file containing the dynamically generated DHCP configs and a con-
figuration script written in Python. This must be done from within a management computer preconfigured within ISELab. From the Course Summary page, click "Deploy DHCP". When prompted to download the zip file, place it in a directory where you can easily find it.

Once downloaded, unzip the contents of the file. Mark the `deploy_dhcp.py` script as executable, and execute from within the terminal (figure 5.17). This will connect to the dhcp server on each ISEHost, copy over the generated configuration files, clean out the old DHCP leases, and restart the DHCP server. Any newly deployed machines should then soon receive an assigned IP.
5.5 Ansible

An automatically generated Ansible `hosts` file can be downloaded in a manner similar to the DHCP configurations and used in future Playbook development. At this point in time, ISEMan is not able to run the Ansible Playbooks itself.
6. Future work

While ISEMan is able to manage, deploy, and configure networking for templated machines, there are yet more features that would allow for a more fine-tuned deployment engine.

6.1 DHCP boot strapping

As of yet, ISEMan relies upon a previously configured set of DHCP servers. These servers must have already had networking interfaces configured and a previously known RSA key installed in order to be configured by ISEMan.

Use of templates with a statically assigned IP and previously installed RSA keys or a PXE server would allow ISEMan to deploy and configure these machines itself.

6.2 Proxied access to vSphere

Currently, ISEMan is required to sit outside of the ISELab environment due to limitations in the underlying pyVmomi library. Transparent proxying is not possible for vSphere requires TLS/SSL on port 443, rendering `iptables` and other host-based proxying useless.

Provided ISEMan can communicate with an external vSphere through either a proxy server or a dedicated internet connection, ISEMan could be hosted entirely within vSphere.

6.3 AD management

ISEMan relies upon an externally managed authentication server for permissions and access rights. Users and groups must be created on this server in order for the permissions that ISEMan assigns to come into play.

Auto-configuration of this server would dramatically improve ISEMan’s independence from outside resources.
6.4 Persistence across courses

By design, ISEMAn does not allow Machines or Users to be associates with more than one Course. This is to simplify ISEMAn use and development, as it provides a clear relationship between models.

With further development, ISEMAn would be able to migrate Machines and Users between courses, allowing students to bring previously created networks and environments into future classes.
7. Conclusion

All in all, ISEMan is a proof-of-concept, showing that automated course management and deployment is entirely feasible. Development with the Django web framework allows ISEMan to be accessed from any operating system, providing flexibility in requirements and use cases.

ISEMan could easily be modified to work in industry settings where replicable environments for testing or user training is required, or in the setup of network services in Cyber Defense Competitions.