May1638 Final Report

Team members:
Steff BisingerChris Read
Sophia Preston Megan Solleveld
Kristin Clemens

Project Design

Project Constraints

Naming Conventions and Terminology

Blue Team	The users in the Operations who are trying to keep services running or to bring them back online.
Event	An isolated incident that results in a specific problem which compromising the network or services
Event Supertype	Main categories of the Events, consisting of ● Environmental/Physical Disaster ● Human Element ● Technical Failure ● Political Threat/Sabotage ● Advanced Persistent Threat
Hypervisor	A piece of computer software, firmware or hardware that creates and runs virtual machines.
ISEAGE	The Internet Scale Event And Generation Environment (ISEAGE) is a security *testbed* “designed and dedicated to creating a virtual Internet for the purpose of researching, designing, and testing cyber defense mechanisms. The ISEAGE *testbed* provides a controlled environment where real world attacks can be played out against different configurations of equipment.” [1]
ISERink	A virtual laboratory environment that allows users an opportunity to undertake hands-on activities focused on networking, cyber security, and penetration testing. To users, it appears as if their network, which uses public address space, is directly connected to the Internet.[2]
Operation	A set of *Events*.
Player	Anyone participating in a *Game*.

[1] Internet-Scale Event and Attack Generation Environment http://www.ece.iastate.edu/research/centers-institutes-and-laboratories/internet-scale-event-and-attack-generation-environment/

[2] IT-Adventures: ISERink (A Cyber Playground) http://www.it-adventures.org/iserink/

Process Testing	To exercise the processing logic of the system to expose errors in attack operations and network model, and to ensure that the system delivers all functionality.
Red Team	The group of computers attacking the Blue Team's network in the Operation.
Tabletop [Exercise]	A discussion-based session where team members meet in an informal, classroom setting to discuss their roles during an emergency and their responses to a particular emergency situation. A facilitator guides participants through a discussion of one or more operations. Also referred to as a ***Game***.
Testbed	An environment consisting of software, hardware, and/or networking components that may be used to test modules and applications in an isolated fashion while behaving as though it is part of a larger program.
vCenter	A VMware product that provides a web interface for managing *virtual machines* running on a *hypervisor*, such as ESXi.
Virtual Machine	A Virtual Machine "is an operating system OS or application environment that is installed on software which imitates dedicated hardware. The end user has the same experience on a virtual machine as they would have on dedicated hardware."[3]
vSphere	A VMware product that packages vCenter and ESXi (a *hypervisor*) among other things.
White Team	The group running the operations, providing support.

[3] http://searchservervirtualization.techtarget.com/definition/virtual-machine

Facts, Assumptions, and Mandated Constraints

1. ISEStorm will exist in an isolated environment while still maintaining limited access to the public internet.
2. ISEStorm will not have access to any real data and information stored on university systems.
3. Events and operations may be designed in part by ITS or other future users of ISEStorm.
4. Any authentication system used will need be approved (and possibly designed) by ITS.

System Requirements

Functional Requirements

1. User shall be able to interact with the virtual machines via SSH/RDP.
2. User shall have the ability to input information to create operations.
3. User shall have a mechanism for initiating an operation.
4. The web application shall allow a user to create an event.
5. The web application shall allow a user to create an operation.
6. The web application shall allow a user to assign events to an operation.
7. The web application shall allow a user to initiate an operation.
8. The web application shall allow a user to initiate an event.

Nonfunctional Requirements

1. Testability
    - 1.1. The product shall provide a framework for testing Events and Operations.

2. Extensibility
    - 2.1. The system shall have a modular design for a “plug and play” introduction of new Events to the system

3. Maintainability
    - 3.1. The product shall have sufficient internal and external documentation to permit future developers to maintain the final product.

4. Reusability
    - 4.1. System should be able be reused for other tabletop exercises.

5. Usability
    - 5.1. ITS users should be able to learn how to use the system fairly easily.

6. Operability
    - 6.1. System should be in reliable, functioning condition.

7. Scalability
    - 7.1. System should be able to handle added work load.

Functional Decomposition

ISEStorm will consist of a virtualized version of Iowa State University's technical systems and a web application that interacts with the virtual system. The virtual model will be on the ISU ISERink. It will be used for tabletop exercises that we will create in which the different sections of the IT department will have to interact to solve problems in this virtual environment in real-time. These exercises will include everything from servers failing to a full-scale assault on the network and will help the IT department to develop plans for real life disasters. ISERink was created by the ISEAGE Research Group at Iowa State University as a platform for virtually modelling network environments for the purposes of training and penetration testing.

System Analysis

ISEStorm provides a usable testbed with various operations to replicate plausible operations for Iowa State University's Information Technology Services. By using a replication the system, a viable solution was created by the team to promote problem solving and knowledge-share amongst the ITS members while still keeping services online and without hampering the actual infrastructure.

Implementation

Input/Output Specification

Input consists of using a keyboard, mouse, Internet access, and interface to access the web application. Once the White Team chooses the particular events, chosen from the different event supertypes, the output will consist of a series of mock disasters that the Blue Team must resolve in order to bring services and/or access back online.

User Interface Specification

ITS members who are part of the White Team set up operations via the ISEStorm web application. The White Team builds an operation by choosing a series of events to occur in a specified order at specified times (or, optionally, manually or automatically triggered). The Blue Team uses remote desktop and/or SSH to mitigate the effects of the events created by the White Team as part of the operation.

Hardware Specification

Our hardware consists of one server: a Dell R610. No other hardware is required.

Software Specification

We are using ESXi and VMware vCenter to create and manage the virtualized systems. We are also using ISEFlow, a program designed by our advisor, Doug Jacobson, to control network traffic on the virtualized network. In addition, we have created a Django web application in Python that allows the White Team to interact with the virtualized servers. This app will run scripts that will trigger network outages and attacks that the Blue Team will have to resolve.

Testing

Process

Our team shared our plans with our client early on asking for feedback. In particular, we sent the client screen sketches of our web application for their approval.

In addition to testing our concept on our target audience, we also implemented two different test suites to ensure that our code is of the highest possible quality. The first uses the testing library built into Django. We use this primarily to test backend functionality such as creating and deleting Events.

Our second test suite uses a software tool called Behave. This is a behavior-driven development tool which means that the tests created using this approach take the form of scenarios or use cases. These tests are written in plain English such that they are easy to read and follow. They also serve as a form of documentation within the code base itself.

Both test suites are run by our continuous integration server whenever a new code is added to the codebase. This protects our codebase from being tainted by errors and poor code as Github will not allow us to merge branches that are failing these test suites. Thus, we have a stable master branch.

Results

The client approved our screen sketches and were very receptive to our concept. Thus, we proceeded with implementation.

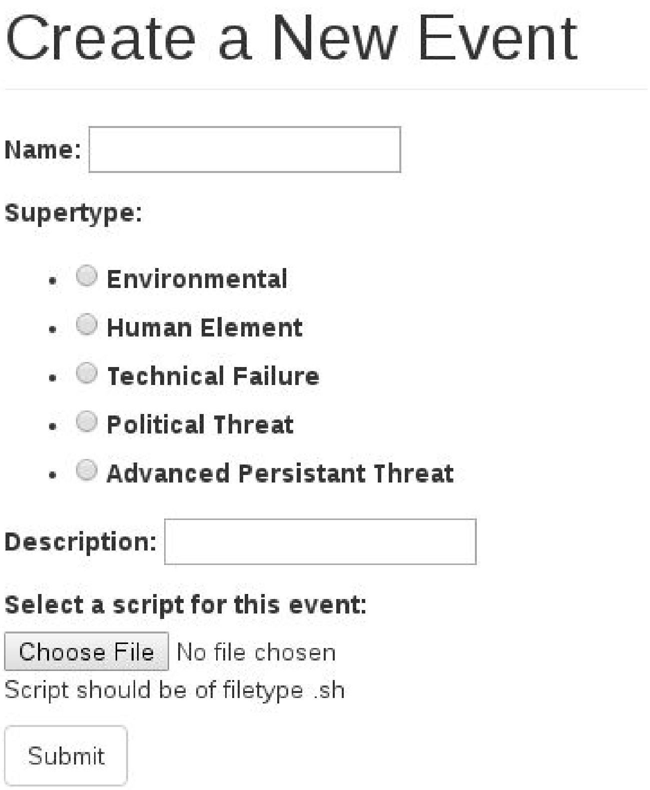
Our testing suites continue to help us to create better code that is much less likely to contain bugs. This means that we have a good codebase to pass off to our client that will not need much refactoring later on.

Appendix I: Operations Manual

Directions for White Team

Creating an Event

1. To create an event, a White Team member will go to the ISEStorm web application in their browser.
2. The White Team member will then click the Event menu and select New Event.
3. Now that they are on the New Event page, the White Team member will be presented with a form that looks like this:



Create a New Event

Name:

Supertype:

- ○ Environmental
- ○ Human Element
- ○ Technical Failure
- ○ Political Threat
- ○ Advanced Persistant Threat

Description:

Select a script for this event:

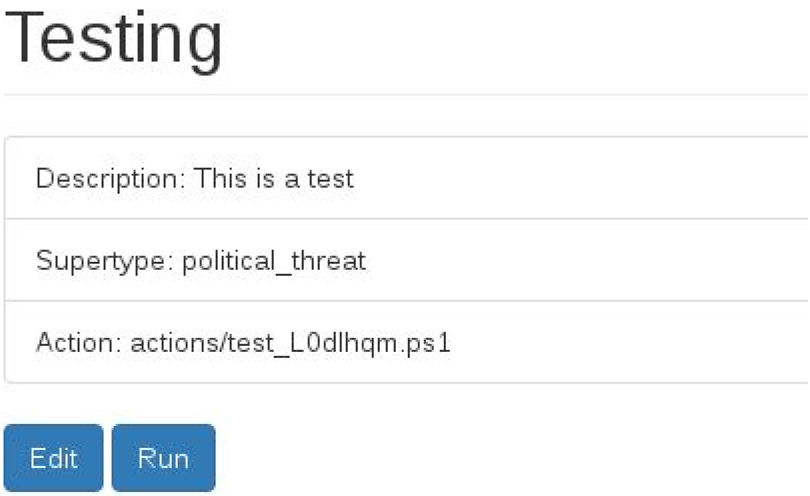
Choose File No file chosen

Script should be of filetype .sh

Submit

4. The White Team member will fill in the relevant information, including uploading a bash script to run, and click the Submit button.

5. If the event is successfully created, they will be redirected to the View Event page for their newly created event. The page should look like this:



Testing

Description: This is a test

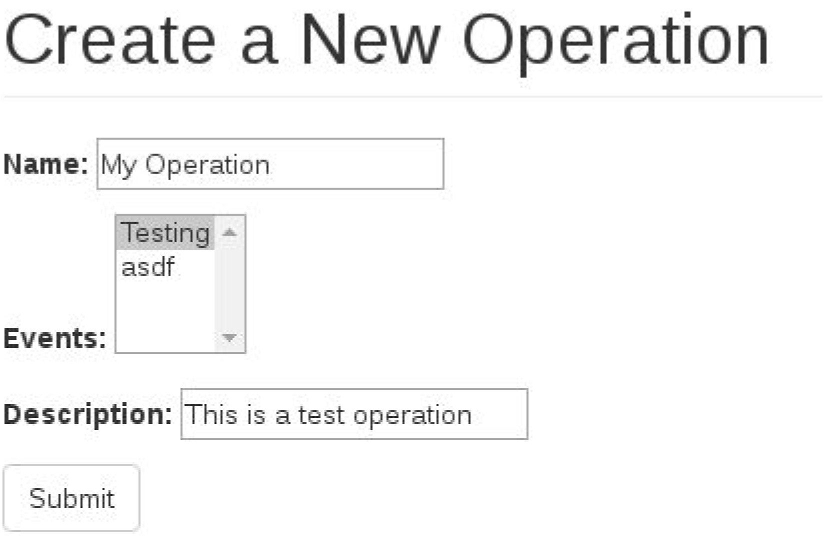
Supertype: political_threat

Action: actions/test_L0dlhqm.ps1

6. If an error occurs while creating the event, the White Team member will be presented with the New Event page again and informed via an alert that something went wrong.

Creating an Operation

1. The White Team member, while viewing the ISEStorm web application in a web browser, will click the ‘Operation’ menu and select ‘New Operation’
2. They will be presented with a form that looks like this:



Create a New Operation

Name: My Operation

Events: Testing
asdf

Description: This is a test operation

3. On this page they will name their Operation, select one or more Events to include in this Operation, and provide a description of the Operation. They will then click Submit.
4. On successful creation of the Operation, the White Team member will be redirected to the View Operation page for the new Operation.
5. On a failed attempt to create an Operation, they will be informed that an error occurred via an alert on the page.

Running a Single Event

1. While viewing the ISEStorm web application, the White Team member will click the Event menu and then select List Events.
2. The White Team member will locate the Event they wish to run in the list of events and click View.
3. On the View Event page, the White Team member will click the Run button.
4. If the Event's action is successfully started, the button on the View Event page will turn green.
5. Otherwise, the page will display an alert explaining why it failed.

Running an Operation

1. While viewing the ISEStorm web application, the White Team member will click the Operation menu and select List Operations.
2. From the list of operations, the White Team member will locate the Operation they wish to run and click the corresponding View button.
3. The White Team member will then be shown a page with the details of the Operation including a list of Events.
4. The White Team member will then click the Run buttons next to the Events in the order they wish to run the Events.
5. As with running a single Event, if the Event's action is successfully started, the corresponding button will turn green, and if running the event fails, an alert will be displayed.

Directions for Blue Team Members

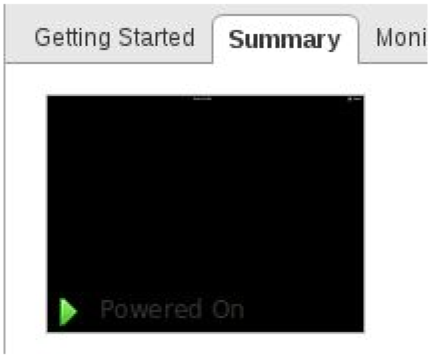
Blue members should have access to the virtual machines on the Blue Net via vCenter, so that they can do things they would normally need physical access for. vCenter also provides a console for interacting with virtual machines. For now, this is how Blue Team members will get console access to the machines on Blue Net.

Turning Virtual Machine On/Off

1. The Blue Team member will log into our vCenter server, which is currently located on the ISU network behind the NAT, using their web browser.
2. The Blue Team member will then click ‘VMs and Templates’ in the menu on the left-hand side.
3. In the resulting menu, the Blue Team member will locate and select the virtual machine they wish to power on/off.
4. They will right-click the virtual machine and select Power > Power On/Off from the right-click menu.

Getting a Console on a Virtual Machine

1. The Blue Team member will log into our vCenter server using their web browser.
2. The Blue Team member will then click on ‘VMs and Templates’ in the menu on the left-hand side.
3. In the resulting menu, the Blue Team member will locate and select the virtual machine they want to access via the console.
4. They will then select the Summary tab in the central section of the page.
5. Next, assuming the VM is turned on, they will click the box that looks like this:



6. This will launch a console on the machine in a new tab within the Blue Team member’s browser.

Appendix II: Alternate Designs

We could have tried to replicate the ISU network with physical servers, but this would have taken more resources and provided little to no benefit. It would also be more difficult to reset such an environment after running Events. Thus, it made sense for us to go for a virtualized approach that would take a single physical server and allow our client to reset the servers on the network using snapshots.

We could also have tried to use a different hypervisor. There is much debate among system administrators about the merits of various hypervisors like HyperV, but because ESXi is what the original ISERink uses, there are numerous tutorials, and one of our group members has some experience with it, ESXi was the logical choice.

As for storing, serving, and running the operations, we had a choice between creating a web application and creating a desktop application. We chose to create a web application, because it would be easy to host on our ISERink and would be easy to get off the ground quickly. This would allow us to get to the work of creating the operations faster and perhaps create a larger quantity of them so that the IT department can train for more situations.

Appendix III: Other Considerations

Hardware

We originally had intended to use a repurposed server from ITS; however, the server was significantly older than we had anticipated (8 to 10 years old) and had no local storage and would not likely support the version of ESXi that we were using. Luckily, we had another server that we were able to use as a backup plan.

Scheduling

We learned that it is important to start coding early on.

Working With A Client

Keeping in contact with the client is extremely important, especially if the client is not very invested in the project. Getting responses from the client was extremely difficult and time consuming; some emails never got a response. Cohesiveness between various client contacts was also an issue. People were assigned and reassigned and left the project numerous times, leaving us unsure of who to contact, and the current contact unsure of what was happening.

Appendix IV

Code

Our code will be available on Iowa State University’s Gitlab server following the presentation.