Cyber Intelligence Analysis Platform

Final Report

Pier-Luc St-Onge
Antoine Lemay
José M. Fernandez
École Polytechnique de Montréal

Prepared By:
École Polytechnique de Montréal
2900, boul. Édouard-Montpetit
Campus de l'Université de Montréal
2500, chemin de Polytechnique
Montréal (Québec) H3T 1J4

Contractor's Document Number: CSSP-2012-TI-1033
Contract Number: 7182702
CSA: Rodney Howes, Portfolio Manager and CoP leader e-Security, DRDC Centre for Security Science, 613-943-2474

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Contract Report
DRDC-RDDC-2014-C108
April 2014

IMPORTANT INFORMATIVE STATEMENTS

CSSP-2012-TI-1033 Cyber Intelligence Analysis Sandbox was supported by the Canadian Safety and Security Program (CSSP) which is led by Defence Research and Development Canada's Centre for Security Science, in partnership with Public Safety Canada. Partners in the project are Royal Canadian Mounted Police, Natural Resources Canada, and École Polytechnique.

CSSP is a federally-funded program to strengthen Canada's ability to anticipate, prevent/mitigate, prepare for, respond to, and recover from natural disasters, serious accidents, crime and terrorism through the convergence of science and technology with policy, operations and intelligence.

POLYTECHNIQUE
MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE

Cyber Intelligence Analysis Platform

Final Report

Contract No. 7182702 – Royal Canadian Mounted Police

Prepared by:

Pier-Luc St-Onge, ing.jr

Antoine Lemay, ing.jr.

José M. Fernandez, ing., Ph.D. , Director – Information Systems Security Research Lab

Table of content

Introduction

This is the final report for the research and development project between the Royal Canadian Mounted Police (RCMP) and l'École Polytechnique de Montréal. The principal objective for this project was to produce a "blue-print" for a Cyber Intelligence Analysis Platform (CIAP), which has advanced capabilities to study sophisticated cyber threats in a secure environment. In this report, a "how to guide" detailing all the key steps to build a CIAP that automates the execution and analysis of complex malware samples is presented. The CIAP follows the design implemented at l'École Polytechnique de Montréal's SecSI Cyber Security Laboratory, which has been used to emulate and study real world botnets at scale in an isolated environment. In particular, the SecSI's cluster has generated a 3000 node Waledac botnet, which enable researchers to understand its complex command and control infrastructure used operate it.

Section 1 - Hardware and Software for the CIAP

Section 1.1 - Overview of the SecSI Laboratory

In this section we will start by providing an overview of the hardware and software components used in the actual SecSI laboratory at l'École Polytechnique de Montréal in order to provide a better understanding of the key elements required to build a CIAP.

Computational Core

The core of the SecSI laboratory (see figure 1) is made up of 7 Blade Centers each of which are loaded with 14 Blades (also referred to as "nodes"). Each Blade Center contains a "thin server" dedicated to the management of each of the 14 nodes through an Advanced Management module (AMM), which can be accessed through a web interface. From this interface each individual node can be remotely controlled. Each node has 4 Ethernet ports (eth0, eth1, eth2 and eth3) used to connect to the experimental and control network.

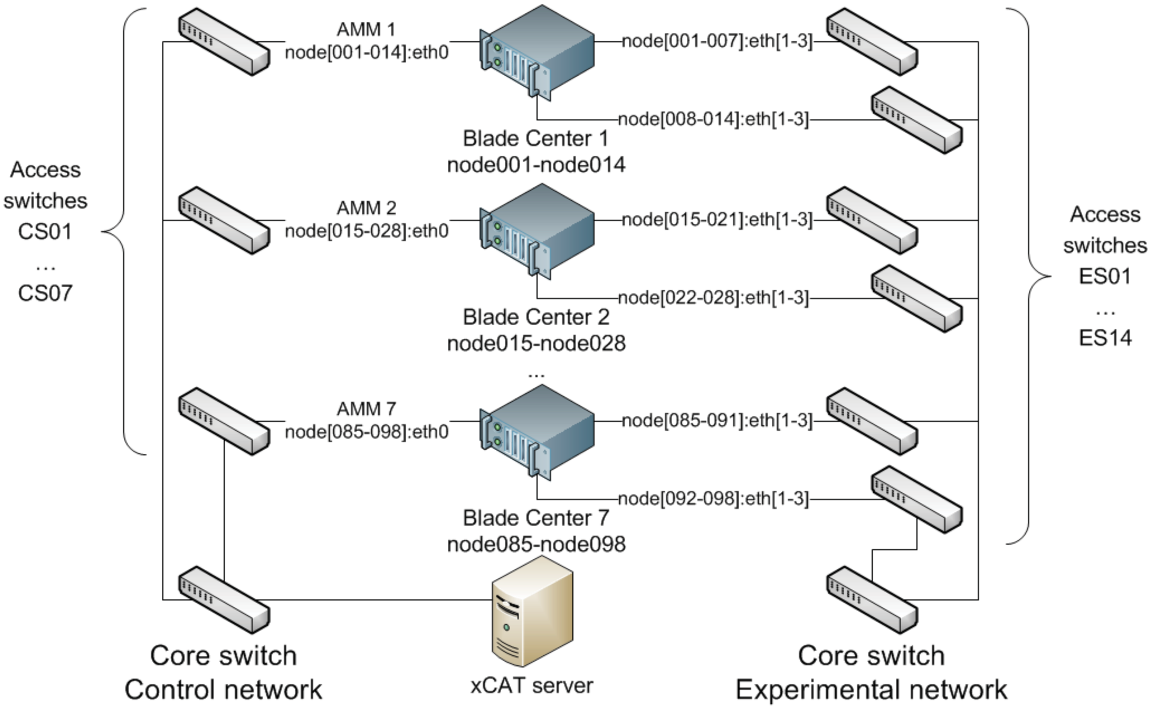

Figure 1 Global view of the SecSI laboratory cluster. "CS" stands for "control network switch", and "ES**" stands for "experience network switch". Total: 421 network connections for 98 nodes**

Separated Networks

A good practice when running malware on a test cluster, even if it is completely isolated from the Internet and the corporate network, is to create an additional layer of protection by segmenting the cluster into an "experimental network," where the malware will be executed, and a "control network," which will manage the setting up of the malware, collecting the data to be analyzed and cleaning up the experimental network post-infection (see figure 1).

Experimental Network

All nodes in the Blade Centers are loaded with VMWare ESX Server and are connected to the access switches for the experimental network through Ethernet ports “eth1”, “eth2” and “eth3” (see figure 1). The ESX Server software enables each node to set up virtual machines, which will be used to set up a target network for the malware to be studied. Because the access switches only have 24 ports, half of the nodes in a Blade Center are connected to one switch and the other half to another. This actually provides better flexibility in segmenting the experimental network. By unplugging one access switch from the experimental network’s core switch, it is possible to completely isolate one half of a Blade Center from the whole experimental network.

Control Network

The control network uses an Extreme Cloud Administration Tools (xCAT) Server to manage the setting up, execution and cleaning up the execution of malware on the virtualized target network on the experimental side. The xCAT Server connects to each node in each Blade Center through 7 separate access switches (see figure 1). The “reserved” port “eth0” is used to manage the VMware ESX server on each node.

From the xCAT server the researcher can force a hard-reboot of any node, reset infected virtual machines to their original state or, if needed, reinstall everything. The SecSI laboratory makes the hypothesis that as long as no virtual machine obtains by itself a connection to the control network then that network will remain effectively isolated from the experimental network.

Detailed Description of the “nodes”

The nodes are Blade Center servers from IBM. In groups of 14, they are packed together in Blade Center chassis. The main specifications for each node are:

- 1 quad-core CPU;
- 8 GB of RAM;
- 2 SAS hard-drive in RAID 0, in order to maximize both the bandwidth and the storage capacity;
- 4 Gigabit Ethernet ports: eth0 is reserved for the control network and eth1, eth2 and eth3 are reserved for the experimental network.

In the SecSI laboratory, the nodes were automatically installed and mostly configured on boot time. In fact, the “xCAT Server” forces the nodes to hard-reboot. Then, the nodes get from the xCAT Server all the tools needed to install the VMware ESX on them using a specific kickstart file, VMware installation packages and configuration scripts.

For the future, more RAM per node is highly recommended and 8-core or 10-core processors should be used. The local storage was an acceptable solution back in 2008 and 2009, but centralized storage over the network (SAN, NAS, etc.) is now the preferred option. In addition, reserving 4 gigabit ports for the experimental network is recommended: one port should be dedicated to outputting the traffic between the virtual switches inside a node. Moreover, by adding one or two 10-Gigabit port(s) and/or fiber-channel ports enough bandwidth would be available to enable the use of a centralized storage system.

The Frontal Node

The frontal node in the SecSI laboratory is the xCAT server, which is mainly used as file server. It consists of the following:

- 1 quad-core CPU;
- 4 GB of RAM;
- 2 SAS hard-drive in RAID 1. The 137 GB volume contains a Linux OS and the virtual machine templates (VMDK files). This is not enough for some usage.
- 1 SAN in RAID 5 which presents a 10 TB volume sliced in 2 TB partitions. This is used for backups and storage of recorded network traffic;
- Three gigabit ports in a load-balancing and redundancy mode;
- Unfortunately, neither power redundancy nor backup was available.

Since xCAT is compatible with Red Hat Enterprise Linux, and since CentOS is the same open-source operating system (OS), SecSI has chosen CentOS (version 5.4 for x86_64 CPUs) mainly because it was well supported by the university and because it is free.

The chosen version of xCAT was the latest in late 2009 (version 2.3.1). In order to optimize the deployment of the virtual cluster the following process was used:

1. Deploy ESX 4.1 on all 98 nodes. Virtual switches and VMware port-groups are built during the installation of each ESX node.
2. Deploy virtual machine templates to each node over the control network. This is done by using vmkfstools over NFS. Since the frontal node hosts all the templates, it is better to limit the deployment to 7 nodes at the same time (7 xCAT commands, each running in a separate BASH shell). Half a day was needed at this step.
3. Deploy all the VMs on each node at the same time. This means 98 nodes working in parallel, each cloning 240 GB worth of VMs locally. Half a day was needed at this step.

In the xCAT configuration, some xCAT database tables were filled:

- **Table nodelist**: a list of all the physical nodes, all the templates and all the virtual machines. They are called “xCAT nodes”. Some custom “xCAT groups” are associated to each of these “xCAT nodes”. For example, node097 may have the groups “esx097, blade13, bc7, compute, all”, which describe the type of “xCAT node” (compute), the targeted Blade Center (#7), the targeted node in the Blade Center (13/14) and the overall node number (97). A “workstation” template to deploy to node097 would have the groups “tw097, workstation_t, esx097, blade13, bc7, template, vm, all”. Finally, the groups “workstation, vm09, esx097, blade13, bc7, vm, all” define the 9th virtual machine as a “workstation” on node097. All these groups help to deal with the thousands of xCAT nodes defined in table “nodelist”;

- **Table vm**: for each xCAT group of VM, one can define the needed RAM, CPU and VMware virtual switch port-groups. The number of port-groups is the number of virtual Ethernet ports that will be generated for each VM that belongs to the given xCAT group. For example, “workstation, 1, 256, office” would create all “workstation” VMs with 1 CPU, 256 MB of RAM and one (1) Ethernet port connected to the “office” network.
- **Tables mac** and **hosts**: xCAT provides a DHCP server for the control network, so the MAC address of each physical node (eth0) must be defined in the “mac” table. In the case of VMs, a custom wildcard has been defined in order to generate automatically different MAC addresses to the Ethernet port(s) of all VMs.
- Few other xCAT tables are also filled with information regarding the management of the cluster: **networks**, **mpa**, **mp**, **passwd**, etc.

Finally, the following files were adapted:

- File **esx.pm**: functions “rmvm” and “mkvms” were adapted in order to remotely delete and create VMs (or templates), respectively. In the case of function “mkvms”, one of its main roles is to create the VMX file with the help of the xCAT tables’ content;
- The kickstart template: this template file is used to generate a kickstart file for each physical node in order to install and configure VMware ESX;
- The “post” installation scripts configures the virtual switches;

Next Steps

The previously described architecture was configured in late 2009. Things have changed in the system administration domain since then: different hardware and software, different storage solutions (local vs centralized), etc. The following sections will present what should be the architecture for the CIAP.

Section 1.2 - Hardware and Software needed for the project

What is needed?

In order to produce a realistic environment that will enable malware samples to evolve naturally, the target experimental network requires multiples layers. These includes both a corporate and operational network layer as well as an emulated fake “Internet” to trick the malicious code into believing that it is “in the wild.” This will require both a VMware cluster and a SCADA network.

The whole SCADA and corporate network should be divided into multiple isolated “sites” with recovery technologies to enable full restoration to the original pristine state. The power sources, networks and storage systems should be fully redundant. If properly isolated or protected by a firewall, a KVM over IP system may provide a remote access to the servers. Finally, VMware Viewers over thin clients can enable researchers to study the evolution of malware samples in “real time” by connecting directly to infected VMs on the experimental network.

The team from l'École Polytechnique de Montréal has suggested the following list of software and hardware to purchase in order to build the CIAP at its smallest scale.

Software Licenses

Early in the project, VMware products have been chosen over all other solutions since it is quite the reference in the industry, and because management tools for other solutions (KVM, Virtual Box, etc.) are not as advanced as those from VMware. All the needed licenses are described in Table 1.

Table 1 Licenses for the new cluster

Licences	Justification
VMware vSphere 5 Standard (15 licenses)	These are licenses for the ESXi nodes. Historically, all those licenses were bought for 12+ nodes of 1 CPU (any number of cores per CPU). In the end, the laboratory could buy fewer nodes of multiple CPUs.
VMware vCenter Server 5 Standard for vSphere 5 (1 license)	While it is possible to do basic management with the free VMware vSphere Client connected directly to the ESXi node, it becomes obviously impossible to manage a cluster without the real tool: a vCenter Server. This software comes with a "free" database (SQL Server Express) and everything works on a Windows Server. This server could be standalone, but could also be a **virtual machine residing in a node it is managing**. This is quite useful if the system administrator wants to keep the vCenter Server on a High Availability (HA) system.
Windows Server x86_64	This is for the VMware vCenter Server and also for the VMware vSphere Client: they only work on Windows. For some reasons, a 64 bits version is needed.

The Workhorse Server

The Blade Center solution with a centralized SAN was too expensive for the budget available. Moreover, the response time of IBM Canada was too long for the project's deadline. The solution was to order a single big server from an OEM provider. Here are the main specifications of the "workhorse server" which is designed to host a few hundred VMs:

Table 2 Specifications of the Workhorse Server

Specification	Justification
VMware certified hardware	To avoid compatibility problems.
Fast local storage 2 * SSD 120 GB (RAID 1) 8 * SSD 480 GB (RAID 5) Hot-swappable 1 * SAS/SATA RAID card, 6 Gbps, with battery	This server will be a VMware ESXi server. It only requires a very small amount of fast storage space (120 GB is more than enough). The massive storage volume (3.36 TB) will be used as a local "datastore" for the VMs. SSDs have been chosen in order to minimize the boot time of all the VMs booting at the same

Specification	Justification
	time. While 3.36 TB seems low, it means an average of 33.6 GB per VM (for 100 VMs), which is more than enough for Windows XP images. With VMware view, it is possible to manage a “golden image” and let the users work on a “linked clone”, which is much smaller than a complete clone of the massive VMDK files (the virtual discs). The RAID card has a battery in case of power outage. But, in fact, since the storage is made of SSDs, the cache on the RAID card is probably not needed. Benchmarks must be done to optimize the system.
A lot of fast RAM 24 * 16GB (384GB), DDR3, 1600Mhz, ECC	This means an average of 3.84GB for each of the 100 VMs.
Many cores, many threads 2 * Intel E5-2680 8-cores each, 2.7 Ghz, 20 MB cache	These are neither the fastest CPUs nor the slowest ones. But since the system is mainly I/O bounded 16 cores or 32 threads are enough.
Dual 10-Gigabits copper Ethernet ports	For the control network. It is fast enough to use the server as an iSCSI NAS in a future configuration.
4 Gigabit Ethernet ports	For the experimental network
Rackmount 2U	For future expansion, this server could be replicated.
Integrated BMC with IPMI – KVM over IP – Intel Remote Management Module 4 1 Ethernet port	This module manages the server, like the IBM AMM. There is an Ethernet port to connect to this module remotely.
Redundant hot-swappable cooling fans	Could be useful in the long term
Redundant power supplies	Mandatory in case of partial power outage.
3 years warranty, on-site service, next business day.	Bare minimum for the kind of system and application.

Since it is an ESXi server, another computer must be used as a client. It could be a laptop or a workstation. The laptop could host VMware vSphere client, VMware Workstation or a remote desktop client that will connect to a Windows VM that runs all the needed tools to manage the server.

At this point, the xCAT Server may not be necessary, since there is only one physical node (server) to manage. But, the xCAT Server itself could be a virtual machine that manages the other VMs and is required for any future expansion of the CIAP. Otherwise, VMware also provides tools that could manage groups of VMs. Finally, there are official VMware Perl and Java SDKs for the development of custom management tools. In any case, all these tools and SDKs would work with the vCenter Server.

Important Hardware Components

Ideally, the CIAP should be build on a stand alone rack loaded with the following hardware components:

Table 3 List of hardware for the rack

Hardware item	Justification
Tripp-Lite PDUMH30HVATNET Single-Phase Auto Transfer Switch / Switched PDU, 30A	The switched PDU is useful for remote power on, power off and power cycling (hard reboot) a specific server. The "auto transfer" feature is for the non power-redundant devices in the rack. So, the PDU has the power redundancy built-in. May one of the two power inputs fail, the other will take over. Rack-mountable for better rack compatibility.
Tripp-Lite PDUMH30HVNET Single-Phase Switched PDU, 30A	The switched PDU is useful for remote power on, power off and power cycling (hard reboot) a specific server. This second PDU should be plugged in a different power source than those of the previous PDU. Power redundant devices should be plugged in both PDUs. Rack-mountable for better rack compatibility.
19" USB KVM Console	If none is already available.
Startech 16 Port Multi-User Cat5 Matrix IP KVM Switch 16 * USB/VGA to KVM IP cables	This is the KVM over IP system for remote access to the server. All the cables are ordered for future usage.
Netgear Prosafe M5300-28G, Managed Switch, 24 Gigabit ports, 2 * 10-Gigabits ports	The 24 gigabit ports could be used for researchers to perform real time monitoring and analysis (19 ports), the management client computer (1 port), the PDUs (2 ports) and the KVM over IP (1 port), the integrated BMC port on the server node (1 port). One 10-Gigabits (10G) port could be used for the control network of the ESXi node. The other 10G port could be used to collect data in real time.

A second Ethernet switch could be used to isolate all the administration traffic: the PDUs, the KVM over IP and the node's integrated BMC module. The Netgear switch should only be used to manage the VMware services.

Now, all this hardware and software must be installed and properly configured in order to do analyze sophisticated cyber threats.

Section 2 - How to Install and Configure Everything

As seen in the previous section, the proposed software and hardware components are significantly different from what is used at l'École Polytechnique de Montréal's SecSI laboratory. In fact, the xCAT Server may not even be necessary to manage this scaled down version of the

CIAP. However, it should be noted that future expansion of the infrastructure to enable the study of malware samples directly on “bare metal” will require xCAT.

Section 2.1 - Hardware installation

Electrical Network

Figure 2 shows the electrical network of the ordered system.

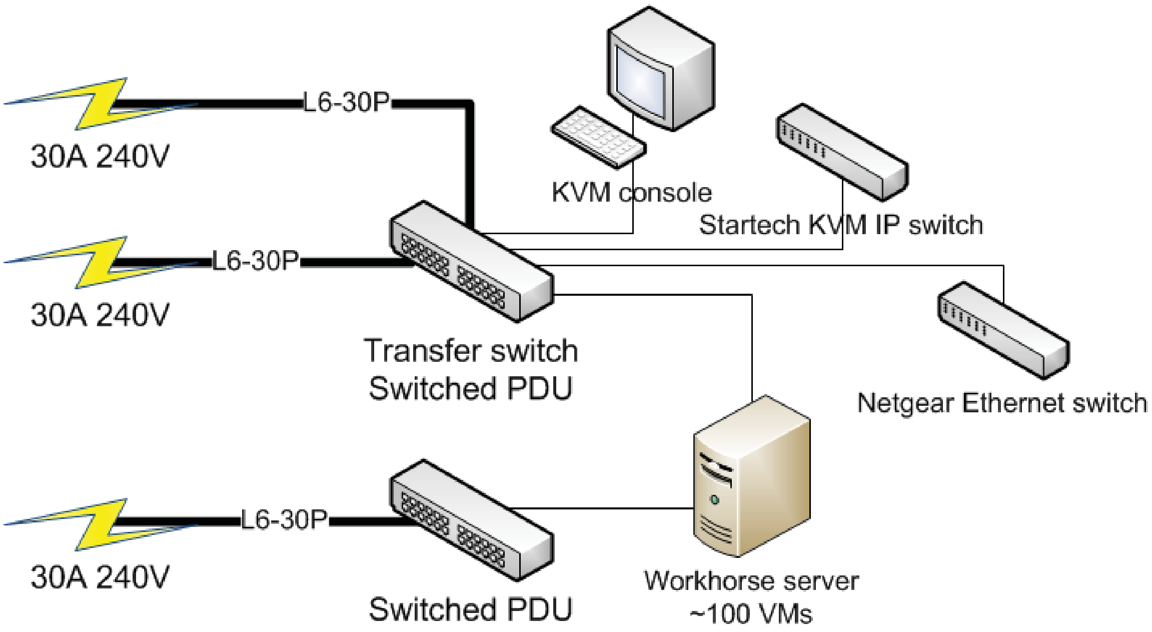

Figure 2 Electrical Network for all devices including the workhorse server node

Like in the SecSI laboratory, there is no battery backup, but the 30A 240V sources may come from big UPS (Uninterrupted Power Supplies) in the future.

Ethernet Network

Figure 3 shows two control networks: one for the administrators and one for the researcher network.

Having a second physical network switch would allow avoiding this tight sharing of the Netgear switch. Furthermore, this would create an air-gap between the management (Mgt) and the researcher network.

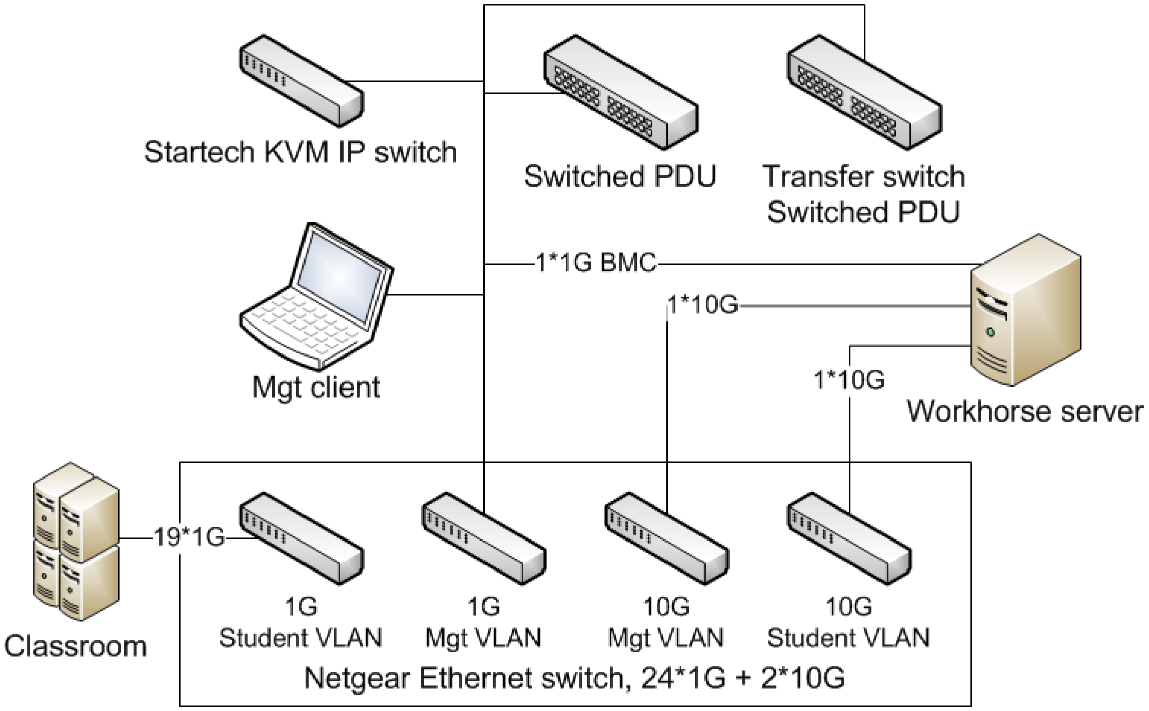

Figure 3 Ethernet Network with 2 VLANs, one for the administrators (Mgt), one for the researchers

Section 2.2 - Software installation and configuration

Installation of VMware ESXi

VMware ESXi installation is quite straightforward for any system administrator:

1. Get a certified compatible server. The workhorse server contains pieces of hardware known to be supported by VMware ESXi. The official compatibility guide could be found here: http://www.vmware.com/resources/compatibility/search.php
2. Choose a fully qualified domain name (FQDN) for the server;
3. Choose its static IP address if there is no DHCP server on the control network. Make sure you know the MAC address of the management port you want to use;
4. Install VMware ESXi, follow the instructions on screen. The hypervisor (ESXi) should be installed on the 120 GB volume (the RAID 1). The main “Datastore” should be on the 3.36 TB volume (the RAID 5);
5. When booted, press F2, enter the root password;
6. Go to Configure Management Network;
7. Go to Network Adapters, and choose the vmnic* for the management port;
8. Go to IP Configuration. Use a DHCP server or set manually the IP Address, the Subnet Mask and the Default Gateway.
9. Go to DNS Configuration. A DHCP server may provide all the necessary information. Without the DHCP server, set a Primary DNS Server IP address and a Hostname (from the FQDN);

The server is now ready, but uses an evaluation license.

Initial Configuration of VMware ESXi

It may be useful to have the SSH service available. From the main screen of ESXi:

1. Press F2, enter the root password;
2. Go to Troubleshooting Mode Options;
3. Go to Enable SSH and press Enter;
4. You may want to ping to the server, and try to connect to it: ssh root@IP_Address;
5. To show the license information from the command line:
    a. Type: vim-cmd vimsvc/license –show ;
6. To see the Datastores and the mounted volumes, type: df -h ;
7. Create virtual switches and port-groups with esxcfg-vswitch. It is also possible to configure these virtual switches and port-groups in VMware vSphere Client on a Windows machine.

You need an external Windows 64 bits machine in order to continue the configuration of the cluster:

1. Connect that Windows machine on the control network;
2. Go to: https://IP_Address of the ESXi server;
3. Download and install the vSphere Client;

Installation and configuration of VMware vCenter Server and vSphere Client

With the vSphere Client, you have to create the vCenter Server in a virtual machine:

1. Create a virtual machine with the vCenter Server's minimum requirements:
    a. http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003882
2. Install Windows Server 64 bits and set its static IP Address;
3. Allow remote desktop connections, if needed;
4. Install vCenter Server:
    a. Install the provided SQL Server Express or install a full version of Microsoft SQL Server or Oracle;
    b. Then install and configure vCenter Server to work with the database.
5. Install vSphere Client on the vCenter Server;
6. Next steps:
    a. Link the vCenter Server to the ESXi node;
    b. Install the licenses for vCenter Server and for the ESXi node;
    c. Configure all the management networks on the ESXi server;
    d. Create accounts for administration access for the researchers;

At this point, it should be possible to configure anything at any scale. For example: create dozens of VMs, starting and stopping them, etc. The tools to do all that could be xCAT or any custom tool built with the provided SDKs from VMware.

References

Here are some key references that are easy to use, follow and adapt.

- VMware Compatibility Guide, http://www.vmware.com/resources/compatibility/search.php
- VMware API and SDK Documentation, http://www.vmware.com/support/pubs/sdk_pubs.html
- VMware vSphere SDK for Perl Documentation, http://www.vmware.com/support/developer/viperltoolkit/index.html
- VMware vSphere SDK for Java, http://communities.vmware.com/community/vmtn/developer/forums/java_toolkit

- xCAT main documentation page, http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Main_Page
- xCAT HowTos page, http://sourceforge.net/apps/mediawiki/xcat/index.php?title=HowTos
- xCAT Basic Install DHCP, http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Basic_Install_DHCP
- xCAT example with iDataPlex nodes, http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_iDataPlex_Cluster_Quick_Start
- xCAT DNS setup, http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_iDataPlex_Cluster_Quick_Start#Setup_DNS
- xCAT Virtualization with VMware, http://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_Virtualization_with_VMWare

- Sumavi xCAT Administrators Guide, http://sumavi.com/books/xcat-administrators-guide
- Sumavi xCAT guide for VMware, http://sumavi.com/chapters/vmware

Section 3 - Setting up the Experimental Environment

The current section supposes that the installation and the configuration has been done and completed, as described in the last section. This also means that the management tools for the VMs deployment are also installed or developed (with VMware SDKs) and configured.

Section 3.1 - Prepare the Experiment

Preparing the experiment means to prepare the network and the templates.

Prepare the Network and Templates

As seen in Figure 3, there is a physical part of the network, but there is also a hidden part of the network that must be configured with vSphere Client or, on the command line, with esxcfg-vswitch.

As a reminder, the workhorse node has 4 allocable gigabit Ethernet ports. While not shown in Figure 3, some of these ports are reserved for connecting to the physical SCADA infrastructure. Inside of the node, those ports must be connected to virtual switches. These switches will then contain “port-groups”. Finally, the virtual machines will be connected to the appropriate port-group(s) in order to communicate with the external hardware.

For example, on the command line:

- Create new virtual switch: esxcfg-vswitch -a switchName
- Link to physical port eth3: esxcfg-vswitch -L vmnic3 switchName
- Add port-group “office”: esxcfg-vswitch --add-pg=”office” switchName

A part of the research is to monitor all the traffic. It is possible to configure the virtual switches in order to monitor all the traffic that goes through it. The same could be done with external managed switches, as long as it is supported by the firmware. A virtual machine could be configured to monitor all the traffic if it has a connection to all the different switches.

Now that the network is properly set, it is possible to create and configure the templates or “golden masters”. The templates are originally virtual machines created directly on the workhorse node. Of course, the VMs are hosted directly on the node, in its internal datastore (the 3.36 TB volume).

Section 3.2 - Monitoring Experiments and Extraction Techniques

It is highly recommended to load the targeted virtual machines that will be infected with the tools required to measure and collect the experimental data that will be used to analyze the malware sample. In particular, the “target” VMs should be loaded with “sensor” software, such as SysInternal tools as well as tcpdump. These tools, along with native commands, can be run using automated scripts to send results to a specific server during the execution of the experiment.

However, it should be noted that from experience some malware have been designed to check to see if the machines that they are about to infect has been “sensored” with tools such as procmon, in which case the malware sample will lay dormant or even deleted itself. In order to defeat such anti-reverse engineering mechanism the target machines can simply use the natives commands net, netstat and tasklist to monitor the behavior of the malware sample.

If the researcher seeks to gain a comprehensive understanding of the malware a “post-mortem” forensic analysis offers the best option for a deep dive dissection of the sample. In this case, none of the target machines are “sensored.” Instead, after the experiment is over all of the virtual machines RAM and hard drives are collected for forensic analysis. Advanced commercial tools such EnCase can then be used to reproduce each step of the infection and propagation of the malicious code.

Section 3.3 - Deploying the Virtual Machines

As a reminder, the SecSI laboratory, thanks to its hardware infrastructure, has to deploy the virtual machines in two steps: 1) deploy all the appropriate templates from the xCAT Server to the 98 nodes, and 2) create all the virtual machines locally on each node.

In the case of the current CIAP (described in section 2), all the templates should reside locally on the workhorse node, especially on the 3.36 TB volume (the main datastore). In other words, the first step of deployment is already done manually by the template maintainer. Finally, deploying virtual machines in this case only requires a local cloning.

In the long term, the workhorse node should be used as an iSCSI NAS. Even with an infrastructure containing a centralized storage, creating virtual machines from templates would still count as a local copy, which is fast because the cloning would actually not run over the network.

In the case of xCAT, the following commands should create the requested VMs:

- List of xCAT nodes and groups: **mkvm** node1,node2,group1,group2
- List of crossed xCAT groups: **mkvm** group1@groupA,group2@groupA

Make a snapshot of all virtual machines. In the case of a major infection all over the place, a simple, low cost, reset to the previous snapshot would be possible.

Finally, the last step consists of powering up all the virtual machines:

- List of xCATnodes and groups: **rpower** node1,node2,group1,group2 **on**
- List of crossed xCAT groups: **rpower** group1@groupA,group2@groupA **on**

Section 3.4 – Running Cyber Threat Scenarios

A comprehensive portfolio of cyber threat scenarios can be generated using the following techniques to infect the experimental network with malware samples:

- Plug an infected USB key on one of the computers running vSphere Client. The targeted virtual machine must have a virtual USB controller and be configured to execute the autorun.inf configuration file;
- Attack the network directly through open services using an exploit or penetration testing kit such as Backtrack. This can be partially or fully automated;
- Attack the network directly through phishing/spear-phishing emails with infected attachments using an exploit or penetration testing kit such as Backtrack. This can be partially or fully automated;
- Attack the network indirectly by setting up a water-holing website using an exploit or penetration testing kit such as Backtrack. The action of the targeted machines visiting the compromised website can be partially or fully automated;
- Mount an infected ISO file as a virtual USB device or a virtual DVD on the VM.

Section 3.5 - Stop the Experiment and Collect the Results

In some cases, stopping the experience may mean to hard-stop all infected and non trustable VMs. The hard-stop may be done with xCAT ("rpower <node-list> off") or with the VMware Client. Pausing may also be a good option if one wants to retrieve some information directly into the RAM of the VMs.

In other cases, the VMs could just wait for a custom message ("cleanup yourself") in order to "clean up" themselves by killing the vulnerable malware. Otherwise, a simple soft reboot of the VM could bring it back to a "clean" state while preserving all the local experiment results. It all depends on the kind of malware used. Of course, if there is a USB key plugged somewhere, simply remove it before restarting the VMs.

At this point, the experiment is stopped. Some results may be in the network traffic that is recorded on a dedicated server, which is considered "not infected". As previously said, some VMs may store internally their local results; you just have to gather all the results by pulling the information from all VMs or by letting the VMs push their results to a dedicated server.

The xCAT server can be scripted to collect all infected virtual machines for in-depth forensic analysis to carefully dissect the actions of the malware sample on the experimental network.

Section 3.6 - Reset the Experiment to a Clean State

Since all virtual machines were saved as a snapshot, reverting them back to the snapshot is a good way to reproduce exactly the same experiment, but with different types of attacks and parameters. If a server was monitoring the network traffic during the experiment the recorded pcap files should be archived before running another experiment. At this point, the CIAP is ready to execute another malware sample.

Section 4 – Enriching Cyber Intelligence

The information extracted from the malware sample should include domain names and/or IP addresses associated with its command and control infrastructure. This provides valuable information not only on the communication channels used by the operator to interact with the malware but more importantly on the potential identity of the threat actor behind an operation. By enriching the extracted information with open source intelligence a profile of the threat actor and/or group can be constructed.

In particular, the extracted domain names from the sample can be analyzed using DNS lookup commands such as dig and nslookup. Understanding the linkages between the FQDN and IP address will be a first step in building a profile of the threat actor's *modus operandi*. It is know from experience that the frequent change of a domain name's associated IP address is a clear sign of botnet behavior.

The IP addresses and domain names extracted and obtained through DNS lookups can be further investigated using the whois command to gather intelligence from various the WHOIS databases. This information will provide new clues on the threat actor's identity. In particular, malware operators often reuse the same email address when registering domain names for their communication infrastructure.

The IP addresses obtained can be furthermore associated with Autonomous System Numbers (ASNs) and more importantly with specific geographic locations. Botnet operators often use servers in geographic regions where there is little risk of prosecution from law enforcement officials. Similarly, certain ASN are more frequently by cyber criminals and spies then others.

By collating and analyzing all the gather intelligence from open source databases, and by further enriching this information with other sources of intelligence, a concise portrait of the malware operator or group can emerge.