1 - NCAE CYBERGAMES GAMEPLAN

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

This document overviews competition strategies that will be used by my team during NCAE Regionals on March 15th. This includes a general 15 minute plan for all services at the beginning of competition as well as tools for intrusion detection, service troubleshooting, and backups during the competition.

TABLE OF CONTENTS

2 - SERVICE UPTIME & SSH

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

SERVICE UPTIME VS SECURITY

- service points come purely from service uptime - taking too long to bring up the service loses points
- there are no extra points for taunting red team and them nuking your service (who'da thunk?)
- bring up the service early with reasonable security, then increase security and make changes as you go

SSH

note: all placeholders are enclosed in `<brackets>`, and the <> are not part of the command, just an indicator that you fill in the blank.

- SSH enables you to access another machine via command line
  - this is helpful for remote access to a machine or if multiple people are working on the same machine
- SSH into another machine:
  - `ssh username@remote_host <command>`
  - `ssh -p <port number> username@remote_host`

3 - COMPETITION IP CONFIGURATION CHEATSHEET

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

BLUE TEAM

SERVICE	INTERNAL/EXTERNAL	IP ADDRESS
Router + INTERNAL GATEWAY	Internal - ether2 interface	192.168.2.1
Router	External - ether1 interface	172.18.13.2
Backup	Internal	192.168.2.15
Shell/FTP	External - DHCP	172.18.14.2
Web Server	Internal	192.168.2.5
Database	Internal	192.168.2.7
Internal DNS	Internal	192.168.2.12
Kali VMs	Internal	192.168.2.4[1-6]
Kali VMs	External - DHCP	(172.18.15.2[1-6])

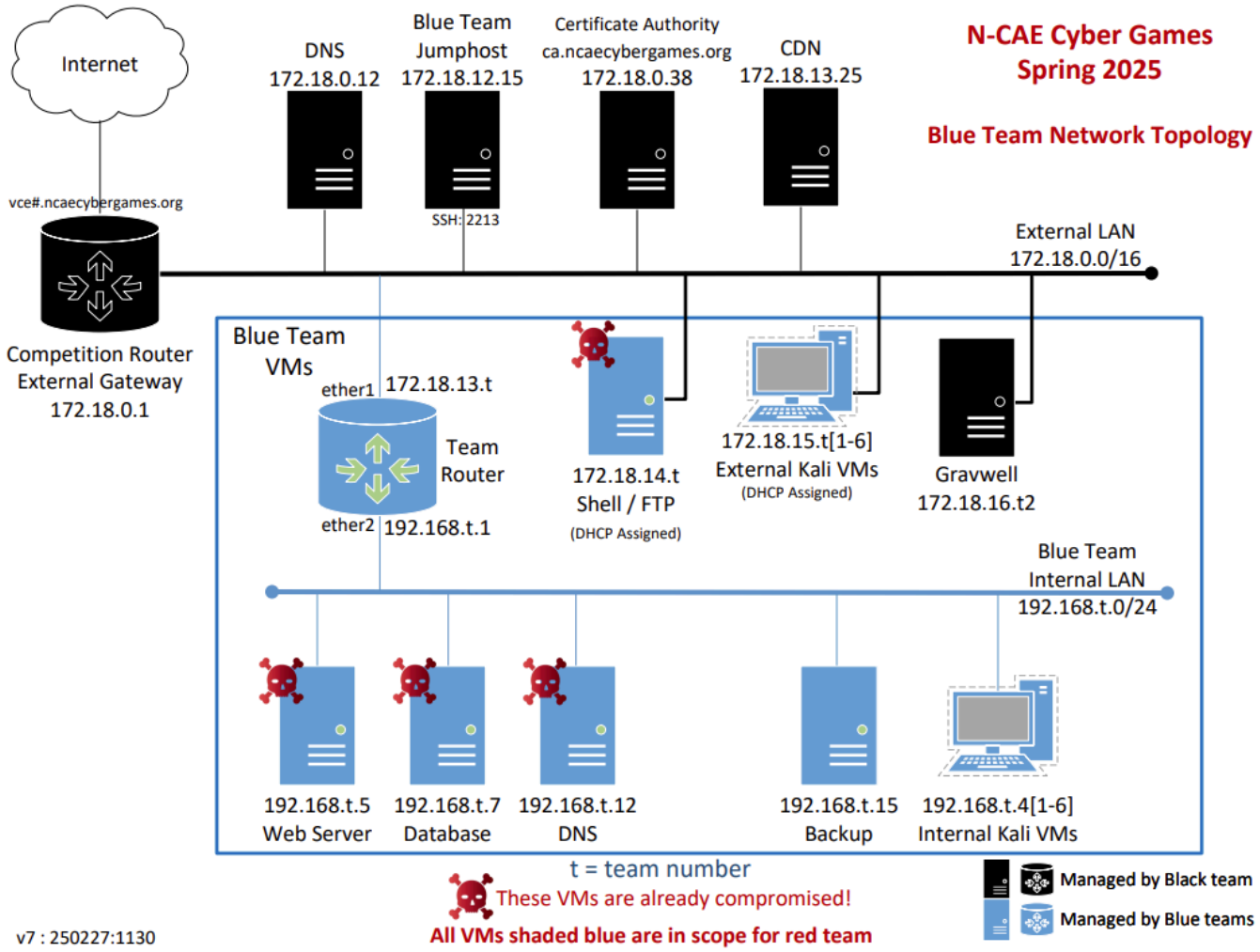
BLACK TEAM

SERVICE	INTERNAL/EXTERNAL	IP ADDRESS
Competition Router External Gateway vce#.ncaecybergames.org	External	172.18.0.1
External DNS	External	172.18.0.12
Blue Team Jumphost	External	172.18.12.15
Certificate Authority ca.ncaecybergames.org	External	172.18.0.38
CDN	External	172.18.13.25
Gravwell	External	172.18.16.22

4 THINGS REQUIRED TO CONNECT TO THE INTERNET

1. IP ADDRESS- see above tables | **2. NETMASK** - 255.255.255.0 or /24 for internal

3. GATEWAY - 192.168.2.1 | **4. DNS SERVER** - 8.8.8.8 8.8.4.4, if it's up: 192.168.2.12


N-CAE Cyber Games
Spring 2025
Blue Team Network Topology
Internet
vce#.ncaecybergames.org
Competition Router
External Gateway
172.18.0.1
DNS
172.18.0.12
Blue Team
Jumphost
172.18.12.15
SSH: 2213
Certificate Authority
ca.ncaecybergames.org
172.18.0.38
CDN
172.18.13.25
External LAN
172.18.0.0/16
Blue Team
VMs
ether1 172.18.13.t
Team
Router
ether2 192.168.t.1
172.18.14.t
Shell / FTP
(DHCP Assigned)
172.18.15.t[1-6]
External Kali VMs
(DHCP Assigned)
Gravwell
172.18.16.t2
Blue Team
Internal LAN
192.168.t.0/24
192.168.t.5
Web Server
192.168.t.7
Database
192.168.t.12
DNS
192.168.t.15
Backup
192.168.t.4[1-6]
Internal Kali VMs
t = team number
These VMs are already compromised!
All VMs shaded blue are in scope for red team
Managed by Black team
Managed by Blue teams
v7 : 250227:1130

4 - 15 MINUTE SERVICE HARDENING PLAN

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

- [ ] basic recon: see what's scored on the scoreboard
  - [ ] check linux version: `cat /etc/os-release`
  - [ ] check if your service is running: `sudo systemctl status <service name>`
- [ ] change root password: `sudo passwd root`
- [ ] create new user with sudo privilege
  - [ ] add new user:
    - [ ] Debian/Ubuntu/Kali/RHEL: `sudo adduser <username>`
    - [ ] Arch/OpenSUSE: `sudo useradd -m -s /bin/bash <username>`
  - [ ] grant sudo privileges:
    - [ ] Debian/Ubuntu/Kali: `usermod -aG sudo <username>`
    - [ ] RHEL/CentOS/Arch/OpenSUSE: `usermod -aG wheel <username>`
      - [ ] make sure `%wheel ALL=(ALL) ALL` is uncommented in `/etc/sudoers`
  - [ ] set default shell: `sudo chsh -s /bin/bash newuser`
  - [ ] disable root account: `sudo passwd -l root`
  - [ ] verify sudo privilege: `su - <username> && whoami`
- [ ] check users (users have uid > 1000): `cat /etc/passwd`
- [ ] check sudo privileged users: `sudo cat /etc/sudoers` or edit with `sudo visudo`
- [ ] create a local backup of `/etc` to a local directory (`see 7 - BACKUPS below`)
- [ ] statically assign IP (`see separate IP configuration documentation`)
- [ ] **configure service!** (`see separate service configuration documentation`)
- [ ] back up and take screenshots of config files (`see 7 - BACKUPS below`)
- [ ] prevent root logon for SSH (but don't block all SSH users if they're scored)
  - [ ] open SSH config file: `sudo nano /etc/ssh/ssh_config` (or sshd_config)
  - [ ] append or change the following lines: `PermitRootLogin no` and `AllowUsers <your username>`
- [ ] configure host-based firewall (`see 6 - FIREWALL`)

5 - 15 MINUTE KALI HARDENING PLAN

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

- [ ] basic recon: see what's scored on the scoreboard
  - [ ] check linux version: `cat /etc/os-release`
  - [ ] check what service you have running: `sudo systemctl status <service name>`
- [ ] change root password: `sudo passwd root`
- [ ] create new user with sudo privilege
  - [ ] add new user:
    - [ ] Debian/Ubuntu/Kali/RHEL: `sudo adduser <username>`
    - [ ] Arch/OpenSUSE: `sudo useradd -m -s /bin/bash <username>`
  - [ ] grant sudo privileges:
    - [ ] Debian/Ubuntu/Kali: `usermod -aG sudo <username>`
    - [ ] RHEL/CentOS/Arch/OpenSUSE: `usermod -aG wheel <username>`
      - [ ] make sure `%wheel ALL=(ALL) ALL` is uncommented in `/etc/sudoers`
  - [ ] set default shell: `sudo chsh -s /bin/bash newuser`
  - [ ] disable root account: `sudo passwd -l root`
  - [ ] verify sudo privilege: `su - <username> && whoami`
- [ ] check users (users have uid > 1000): `cat /etc/passwd`
- [ ] check sudo privileged users: `sudo cat /etc/sudoers` or edit with `sudo visudo`
- [ ] create a local backup of `/etc` to a local directory (`see 7 - BACKUPS below`)
- [ ] statically assign IP (`see separate IP configuration documentation`)
- [ ] back up and take screenshots of config files (`see 7 - BACKUPS below`)
- [ ] harden root logon for SSH (but don't block all SSH users if they're scored)
  - [ ] open ssh config file: `sudo nano /etc/ssh/ssh_config` (or sshd_config)
  - [ ] append or change the following lines: `PermitRootLogin no` and `AllowUsers <username>`
- [ ] configure firewall (`see 6 - FIREWALL`)
  - [ ] in the beginning while the VM is not being used, prevent all incoming traffic
  - [ ] allow traffic when actually using the VMs

6 - FIREWALL

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

UFW - Ubuntu/Debian/Kali

- check firewall status: `sudo ufw status verbose`
- turn on firewall: `sudo ufw enable`
- enable required port: `sudo ufw allow <port number/protocol>`
- enable required IP: `sudo ufw allow from <ip address>`
  - e.g. `sudo ufw allow 23/tcp from 192.168.1.12`
- deny traffic if needed: `sudo ufw deny <port number/protocol>`
- deny all incoming traffic: `sudo ufw default deny incoming`
- delete a rule: `sudo ufw delete <rule>`
  - e.g. `sudo ufw delete allow 22/tcp`

firewalld - CentOS/RHEL

- check status: `firewall-cmd --state`
- enable firewall: `systemctl enable firewalld`
- start firewalld: `systemctl start firewalld`
- allow required service: `firewall-cmd --permanent --add-service=<service name>`
  - e.g. `firewall-cmd --permanent --add-service=ssh`
- allow required port: `firewall-cmd --permanent --add-port=<port number/protocol>`
  - e.g. `firewall-cmd --permanent --add-port=8080/tcp`
- deny specific traffic if necessary
  - `firewall-cmd --permanent --remove-port=<port number/protocol>`
  - `firewall-cmd --permanent --remove-service=<service name>`
- deny all incoming traffic: `sudo firewall-cmd --set-default-zone=drop`
- list services: `firewall-cmd --permanent --list-all`
- reload to apply changes: `firewall-cmd --reload`

iptables - reference separate documentation

7 - TARBALL BACKUPS

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

directory and file names for this example:

- backup directory on local machine: `/etc/mybackupdir`
- backup directory on backup server: `/home/ourbackupdir`
- tar ball filename: `ball.tar.gz`
- file being backed up: `/etc/config/service.conf`
- backup server's IP address: `192.168.1.255`
- `strip component level = 2` for `/etc/mybackupdir/ball.tar.gz`, since there are 2 directories listed before the file we want to backup (/etc and /mybackupdir get stripped)

on backup server

- make a directory for all backups: `sudo mkdir /home/ourbackupdir`
- power off the backup server whenever it's not in use

send backup from local machine

- [ ] make local backup directory: `sudo mkdir /etc/mybackupdir`
- [ ] install tar and gzip: `sudo apt install -y tar gzip`
- [ ] create a tar.gz file copy of the files: `sudo tar -czvf /etc/mybackupdir/ball.tar.gz /etc/config/service.conf`
- [ ] send the backup to the backup server with SCP: `scp /etc/mybackupdir/ball.tar.gz root@192.168.1.255:/home/ourbackupdir`

retrieve backup from backup server

- [ ] retrieve the backup: `scp root@192.168.1.255:/home/ourbackupdir/ball.tar.gz /etc/mybackupdir`
- [ ] switch into original local directory you want to move the backup to: `cd /etc/config`
- [ ] extract the backup: `sudo tar -xzvf /etc/mybackupdir/ball.tar.gz --strip-components=2`

8 - FILE PERMISSIONS

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

chmod value	permission
`0`	--- (No permissions)
`1`	--x (Execute only)
`2`	-w- (Write only)
`3`	-wx (Write & Execute)
`4`	r-- (Read only)
`5`	r-x (Read & Execute)
`6`	rw- (Read & Write)
`7`	rwx (Read, Write & Execute)

chmod symbol	meaning
`r` (4)	Read permission
`w` (2)	Write permission
`x` (1)	Execute permission

- change file or directory permissions with `chmod` (change mode)
- basic syntax:
  - `chmod <permission level for owner, group, everyone> <file or directory>`
  - e.g. owner has all permissions, others have read: `chmod 744 /var/html/www/index.html`
- for passwd file: `chmod 644 /etc/passwd`

9 - PROCESS RECONNAISSANCE

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

- list logged-in users
  - `who`
  - `w`
  - `last -a`
- check users
  - `cat /etc/passwd` (users have uid > 1000)
  - `sudo cat /etc/sudoers` (users with sudo)
- check running processes
  - `ps aux --sort=-%cpu | head`
- check processes running as root
  - `ps aux | grep root`
- check hidden processes
  - `ps -ef | grep '[]'`
- check cronjobs
  - `cat /etc/crontab`
- check services
  - `ls /etc/init.d`
- system-wide settings
  - `/etc/bash.bashrc`
  - `/etc/profile`
- check command history
  - `cat ~/.bash_history`
- check command history for a specific user
  - `sudo cat /home/<username>/.bash_history`
- list active connections
  - `netstat -tulnp`

    - `ss -tulnp`
    - `ss -peanuts`
- shell camping (not recommended because it annoys red team)
    - `kill -9 <PID>`
    - `pkill -u <username>`
    - INSTEAD: check logs to see what red team is doing and then make changes based off of that
- change passwords
    - `passwd <username>`
- delete suspicious user accounts
    - `userdel -r <username>`
    - do not delete the nobody user! (system-level user running processes with minimal permissions to limit damage in case of being compromised)

10 - LOGS

Author: Hope Tan, LU NCAE Cybergames Team Captain '24-25

note: these may be different depending on Linux flavor and other configurations.

- do a service reload if you reconfigured files
  - `sudo systemctl reload <service>`
- check the service status
  - `sudo systemctl status <service>`

tools to search logs:

- journalctl - troubleshoot services
  - `journalctl -xeu <service>`
- ausearch - search and filter audit logs

ausearch command	description
ausearch -m [EVENT_TYPE]	search by event type (e.g., USER_LOGIN, SYSCALL, AVC)
ausearch -ua [UID]	search by user ID
ausearch -ul [USERNAME]	search by username
ausearch -p [PID]	search by process ID
ausearch -x [EXECUTABLE]	search by executed command
ausearch -k [KEY]	search by custom audit key
ausearch -hn [HOSTNAME]	search by hostname

specific file paths:

- Linux:
  - general system logs: `/var/log/syslog`
  - authentication logs: `/var/log/auth.log`
  - check failed logins: `sudo last -f /var/log/btmp`
  - check all logins: `sudo last -f /var/log/wtmp`
- firewalls and IDS/IPS:
  - iptables:

        - firewall logs: `/var/log/iptables.log`
- web servers:
    - Nginx:
        - access logs: `/var/log/nginx/access.log`
        - error logs: `/var/log/nginx/error.log`
    - Apache:
        - access logs: `/var/log/apache2/access.log`
        - error logs: `/var/log/apache2/error.log`
- web applications:
    - PHP:
        - error logs: `/var/log/php/error.log`
        - troubleshoot syntax: `php -l <filename.php>`
- databases:
    - MySQL:
        - error logs: `/var/log/mysql/error.log`
    - PostgreSQL:
        - error and activity logs: `/var/log/postgresql/postgresql-{version}-main.log`