the first thing that attackers do is look for low hanging fruit

1) change passwords, use different passwords for users/boxes

shared passwords create multiple security holes

be aware that once they have enough of a foothold, attacker no longer

needs to know your passwords

change your passwords often

In order to do this as efficiently as possible, there should be a master list of

passwords that is created before the competition. This list needs to be large

enough to allow for different systems to have different passwords which can be rotated

thoughout the competition. This rotation should occur at a fixed, predetermined interval.

By having a complete list of passwords prior to the start, it should be simple enough to write

which machine is using which password and for what round next to the password.

2) who's on there and who shouldn't be

tail -n 30 /etc/passwd

disable uneeded accounts

remove admin rights from accounts that don't need it

Obviously we won't be able to tell who needs admin rights and who doesn't before we have an understanding

of how the network works. The key here is to simply notate which users have certain permissions so

that we can go back and validate these users against required functionality.

Going along with this is checking your own OS, see the strike team guides for specific commands to identify

kernel arch and version

We can also see which services are running on our machine

sudo service --status-all

which processes are using the most cpu

top

what network connections look like

netstat -natp

netstat -tulpn

What network's we're connected to

ifconfig

inspect routing table

What's on the network?

To get a better understand of our network we can turn to nmap

nmap -F <host> # runs a quick scan of the 100 most common ports

nmap -sP 10.0.0.0/24 # scans the network listening for machines that respond to patching

nmap -A 1<host> # detect OS and services

common initial attacks are ssh\_login on linux and smb/psexec on windows

3) check ports 22 and 445

Similar to the user situation we won't know what machines need what ports until we understand the network

but keep track of what's open where so that we can close everything extra once we know.

your host firewall should be blocking port 445 unless it is absolutely necessary

change default ssh login credentials

look for dropped ssh keys, or ssh configuration that allows multiple users to use the same key

ls -la ~/.ssh

~/.ssh/authorized\_keys

if there's a key in .ssh that was added recently

when you look at authorized\_keys it gives you the key and the user associated with it if the same

key can log in to multiple users

they're getting in regardless but we would like to know how they're getting out

4) set up network monitoring

don't just focus on stopping post exploitation

in order for them to do stuff, information needs to come in and out

find out how it's happening and stop it

be aware of multiple ways to exfil data and what to look for

tcp

reverse shells

When something in your machine tries to call back to the attacker

look for high ports on your machine, odds are you're running standard services

so high ports making connections out of the network are signs that it's calling home

inbound shells

inbound connections to ports that aren't running known services

http

if it's just a website with a strange nonsense string on it then that could be c2

don't ask me how it works, i have no idea, but i know that it happens

dns tunneling

large number of requests to strange subdomains

block those domains

whitelist outbound ports

force web traffic through an authenticated proxy server to deny SYSTEM from exiting

now that we've taken closed the front door, close up holes in your walls

6) work on patching machines

ensure that those patches are applied and active

especially patch rce vulnerabilities

linux:

apt-get update

yum update

windows:

system security -> windows update -> view update history

-> restore hidden updates

check for updates

wmic qfe get hotfixid | find "<#>"

6) firewall

strict inbound and outbound rules

focus on whitelisting what we need

key to success is to defeat their persistence

7) how do they keep getting in?

indicators of compromise

hash values: Ssdeep and other fuzzy hash tools that might be able to match commercial malware

ip addresses: it's always good to block bad ips, but it's easy for attacker to change

domain names: little harder to change, necessary for dns tunneling

network artifacts:

uri patterns

c2 info in network protocols

This stuff comes from wireshark, once you are able to narrow down which addreses are malicious

you can try to watch the traffic in wireshark

distinctive http user agent strings

if you're using a proxy or some sort of custom request generator then they might not have bothered

to fill in the entire long user agent string

distinctive smtp mailer values

host artifacts:

registry keys and values: regshot

files with names that are similar to real windows files

especially stuff in system32, syswow64, program files, etc

stuff that will run at startup

alternate data streams in files

what processes are connecting to the internet and shouldn't

which processes are taking up way too much memory (process injection/reflective loading)

how to identify stuff that's bad

linux

ps -aux

netstat -napt

netstat -tulpn

lsof

suid programs

find <directory> -perm -4000

service --status-all

look through /proc/<pid> directories

chron

windows

sysinternals

procmon

tcpmon

autoruns

streams

regshot

task manager

Continually monitor network and conduct active defense throughout

8) watch for dirty tricks and a few of your own

static arp entries and arp poisoning

check service status

red team likes to stop services

and delete important files

so have a backup or know how to redo it

remove notepad.exe (default spawn process for cobalt strike and metasploit)

a note on shapshots

Reset to initial snapshot

vulnerable to default but it might manage to kick them off

penalty???

don't want to rever to snapshot where they already were in the network