Úvod do počítačovej bezpečnosti FEI STU

LAMP Security project

[Linux Apache MySQL PHP]

1 Environment setup

TARGET MACHINE – **LAMP-CTF8** [web server]

ATTACK MACHINE – **Kali Linux**

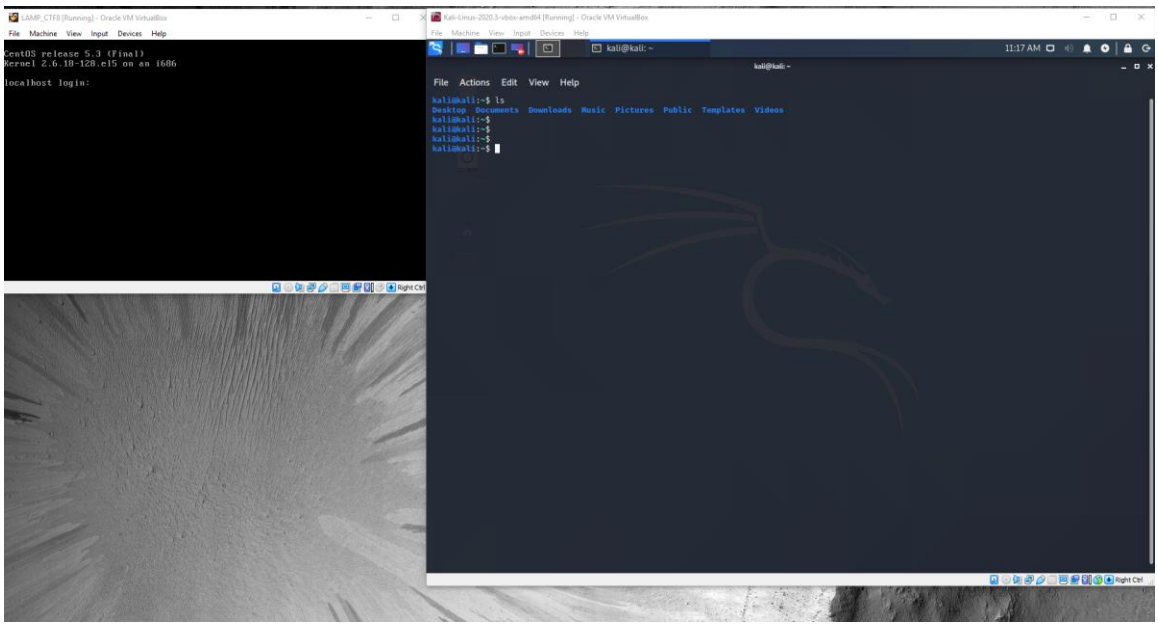


Figure 1 Environment setup – Oracle VM VirtualBox

Target aj attack platformy spúšťam cez VirtualBox od Oracle. Stiahol som LAMP server verziu CTF8. Prvým predpoklad úspešného útoku je poznať IP adresu targetu. Na to aby VirtualBox rozlíšil na sieti target a attack machiny musel som pre každú platformu nastaviť sieťový adaptér v nastaveniach VirtualBoxu.

2 Testovanie bezpečnosti serveru

2.1 Zistenie IP adresy targetu

Predpokladom tejto úlohy je, že attack machine Kali Linux je na rovnakej lokálnej LAN sieti ako target. V tomto prípade začneme tým že zistíme akú IP adresu má pridelenú attack machine.

```
kali@kali:~$ /sbin/ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.56.105  netmask 255.255.255.0  broadcast 192.168.56.255
      inet6 fe80::a00:27ff:fec8:ae6b  prefixlen 64  scopeid 0x20<link>
```

Zistili sme, že attack machine ma dynamickú IP **192.168.56.105**.

Vieme, že target je na rovnakej LAN, to znamená, že DHCP server mu pridelil adresu vo formáte 192.168.56.XXX. Musíme zistiť čísla subadresy XXX. Na to použijeme penetračný nástroj, napríklad **nmap**.

```
kali@kali:~$ nmap 192.168.56.1-255
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-28 11:52 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
```

Dostali sme response len z jedného zariadenia na tejto sieti a to pravé z target machiny. Teda target ma adresu IP **192.168.56.102.** Teraz sme schopný útočiť na server sieť.

2.2 Zistenie slabých a zaujímavých miest cez sieťové pripojenie

V tomto kroku sa pozrime bližšie na výstup z príkazu **nmap** pre target IP adresu.

```
Nmap scan report for 192.168.56.102
Host is up (0.00029s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.0.5
22/tcp   open  ssh         OpenSSH 4.3 (protocol 2.0)
25/tcp   open  smtp        Sendmail
80/tcp   open  http        Apache httpd 2.2.3 ((CentOS))
110/tcp  open  pop3        Dovecot pop3d
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open  imap        Dovecot imapd
443/tcp  open  ssl/https?
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
993/tcp  open  ssl/imaps?
995/tcp  open  ssl/pop3s?
3306/tcp open  mysql       MySQL (unauthorized)
5801/tcp open  vnc-http    RealVNC 4.0 (resolution: 400×250; VNC TCP port: 5901)
5802/tcp open  vnc-http    RealVNC 4.0 (resolution: 400×250; VNC TCP port: 5902)
5901/tcp open  vnc         VNC (protocol 3.8)
5902/tcp open  vnc         VNC (protocol 3.8)
5903/tcp open  vnc         VNC (protocol 3.8)
5904/tcp open  vnc         VNC (protocol 3.8)
```

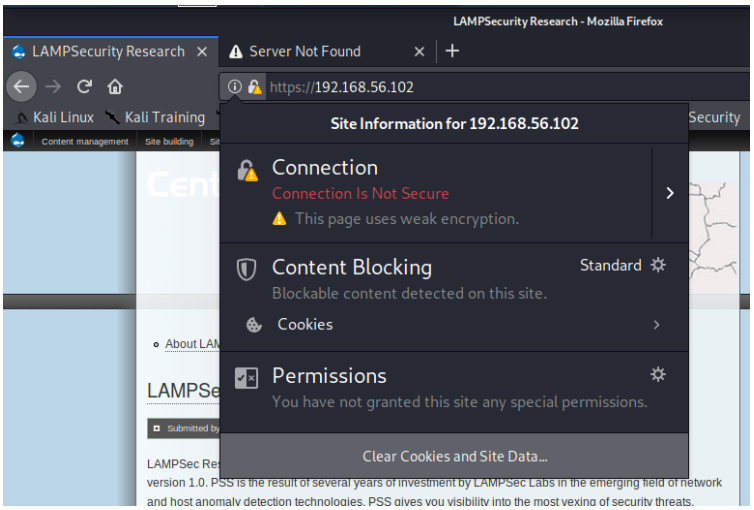
Už samotný výpis príkazu ***nmap -sV 192.168.56.102,*** hovorí veľa o bezpečnosti serveru a dáva nam možnosti na ktoré sieťové funkcionality sa môžeme zamerať. Každá sieťová funkcionalita komunikuje cez špecifický **port**. Napríklad SSH cez 22, VNC na 5901, web server na porte 80. Náš target ma cez firewall povolené takmer všetky sieťové funkcionality.

Teda hodnotím, že target machine mohol v produkciu zavrieť niektoré TCP porty pomocou firewallu.

2.3 Analýza sieťovej služby web serveru Apache na porte 80

Najkomfortnejšia služba pre nás je web server aplikácia na porte 80 lebo poskytuje GUI. Navštívime stránku 192.168.56.102 cez náš attack Kali Linux a vidíme komplexnú web stránku s množstvom textových vstupov, podstránok a súborov.

Ako prvé si všimneme, že Apache server nemá implementovaný šifrovací https protokol pomocou SSL a používa len http. Toto je jednoznačne slabina web aplikácie a je možné využiť nástroje a postupy na odchytávanie komunikácie.

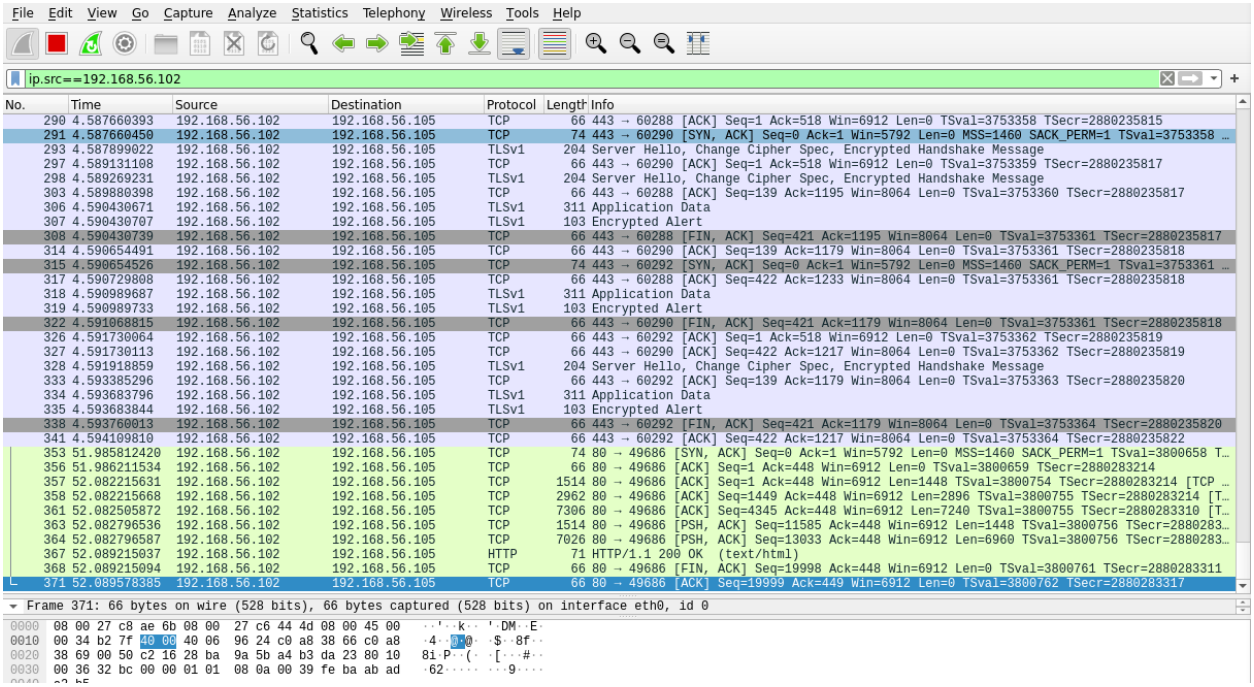


Môžeme predpokladať, že webová služba má ďalšie bezpečnostné nedostatky preto použijeme automatizované penetračné nástroje, ktoré Kali Linux ponúka.

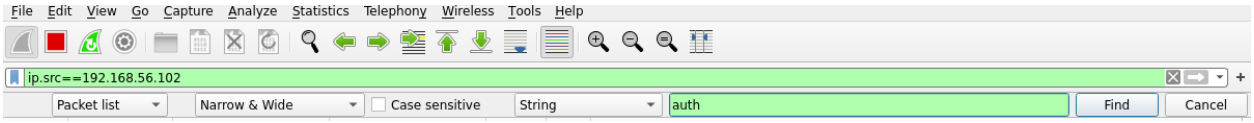
Tool **NIKTO** (***nikto -host 192.168.56.102***) deteguje množstvo potenciálnych slabín, avšak veľa z nich sa nedá reálne využiť. Avšak získame kompletné informácie o backend infraštruktúre a verziách.

2.4 Phishing pomocou WireSharku

Keďže sme zistili, že target používa nešifrovanú http komunikáciu, je tu možnosť odchytávať komunikáciu a získať citlivé informácie. Na odchytávanie sieťových TCP parketou som zvyknutý používať WireShark.



Použijeme filter pre sledovanie packetov len zo strany targetu ***ip.src==192.168.56.102.***

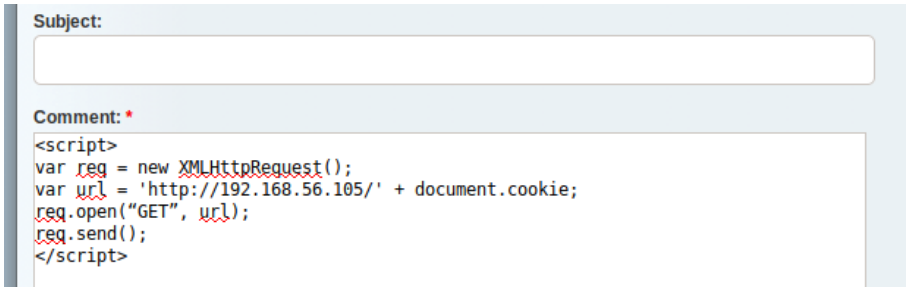


Vyskúšal som hľadať v paketoch stringy ktoré by mohli poskytnúť citlivé informácie, napríklad **auth, pass, pwd..** ale nepodarilo sami nájsť. Tak pokračujem podľa návodu.

2.5 Cross Site Scripting

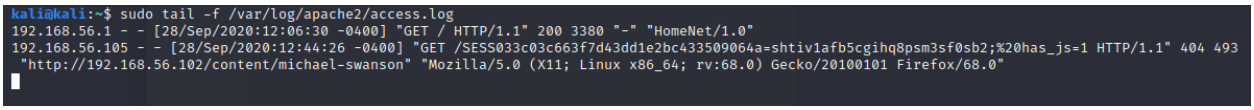
Web stránka obsahuje textové vstupy ako komentáre, príspevky, ktoré akceptujú formát HTML, čo je absolútna bezpečnostná chyba v tomto prípade. Vďaka tomu sme schopný napísať JavaScriptový exploit, ktorým získame cookies užívateľa prihláseného ako administrátor.

Exploit vyzerá takto:



Po uverejnení komentára sa HTML kód nezobrazí ale interpretuje a vykoná JS príkaz, ktorý odošle obsah cookies na IP adresu Attack machiny, na ktorom beží web server, ktorý odchytáva prijaté packety.

V momente, kedy administrátor načíta stránku na ktorej je exploit, prijmeme obsah cookies vďaka ktorým sa vieme prihlásiť ako administrátor.



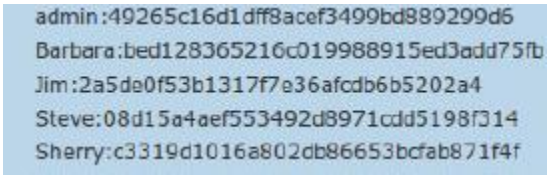
(SESS033c03c663f7d43dd1e2bc433509064a=shtiv1afb5cgihq8psm3sf0sb2)

Stačí sa na attack machine zmeniť v databáze obsah cookies a sme prihlásený ako administrátor. Tu by sa dala implementovať prídavná ochrana pri prihlasovaní.

2.6 PHP backdooring

Zistili sme, že administrátori môžu na stránke pridávať PHP funkcionality web aplikácie. To je neobvyklé a veľmi nebezpečné. Je to jeden z hlavných bezpečnostných faktorov celého systému.

Vďaka administrátorskému prístupu na web stránke môžeme pristupovať k databáze serveru. Nakoľko stránka obsahuje autentifikáciu členov a adminov, musí v databáze ukladať prihlasovacie mena a HASH kódy hesiel. Jednoducho pridáme PHP **select db_query('select name,pass from users');** a dostaneme

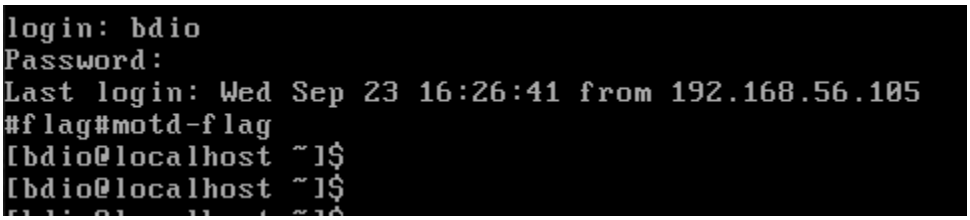


Vidíme, že používajú staré a slabé šifrovanie takže je šanca nájsť heslo medzi leaknutými heslami pomocou toolov ako **john**. A naozaj väčšinu hesiel sme schopný dekriptovať. Tieto heslá sú však len prihlasovacie heslá to online aplikácie užívateľov a administrátorov.

2.7 Server backend access

V tomto momente poznáme heslá pre web, je šanca, že niekto z administrátorov bude mať rovnaké heslo na web aj na system. Pomocou PHP backdooringu sme schopný nájsť systémové mena používateľov.

Administrator Barbara Dio má web heslo ***passw0rd,*** na backende existuje systemove meno bdio, čo bude Barbarin account. A nazaj Barbara spravila bezpecnostnu chybu, kedy pouzila rovnake heslo ako na web tak aj na system.



Teraz sme schopný vzdialeného prístupu priamo na backend pomocou SSH. Nanešťastie barbara nie je sudo user, takže nemáme ešte úplný prístup.

Vykonajte inštrukcie uvedené v návode (ctf4_instruction.pdf) a pokúste sa samostatne získať administrátorský prístup do testovaného systému. Zapamätajte si všetky zraniteľnosti ktoré v systéme boli a ktoré vám umožnili útok úspešne zrealizovať.

5. Pokúste sa vyriešiť aj ďalšie možné spôsoby pre útok popísané v závere dokumentu ako Other Unscripted Attack Vectors.

6. Bonus: Na základe získaných poznatkov sa pokuste navrhnuť ako systém zabezpečiť (aspoň niektoré zraniteľnosti) tak, aby ho nebolo možné kompromitovať.

```
kali@kali:~$ nmap 192.168.56.1-255
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-28 13:46 EDT
Nmap scan report for 192.168.56.101
Host is up (0.56s latency).
Not shown: 996 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
25/tcp  open   smtp
80/tcp  open   http
631/tcp closed ipp
```