CHISEL - Networking Challenges

*le CTF est un jeu consistant à exploiter des **vulnérabilités** affectant des logiciels de manière à s'introduire sur des ordinateurs pour récupérer des "flags" preuves de **l'intrusion**.

But du projet

Le projet est composé de plusieurs *CTF** sur 3 thématiques différentes:

- Privilege escalation
- SSH Tunneling
- Proxying Tunneling

Quels sont les différences entre un bon H4ck3r et un mauvais H4ck3r ?

- Bonne culture générale des failles qui existent
- Connaissance des outils existants pouvant détecter les failles potentielles
- Être patient et progresser à son rythme

Explication des thématiques

Privilege escalation

Passer d'un simple user à un accès administrateur (root).

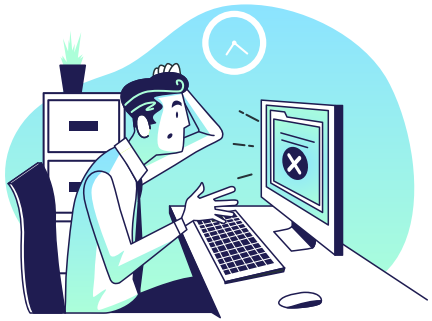
SSH Tunneling

Ouvrir le terminal en local grâce à un partage des ports

Proxying Tunneling

Passer à travers un proxy par le biais d'un serveur HTTP

Difficultés rencontrées - VS - Solutions apportées



Problèmes

- Manque de connaissances sur les technologies de découvertes de failles
- Manque de méthodologie de recherche de flags

Solutions

- Apprentissage des méthodes de découvertes de failles (forums, expériences étudiantes)
- Premiers CTF très enrichissants pour apprendre comment maîtriser la bonne méthodologie

Technologies utilisées

NMap

Scan de ports sur une ip

Dirb

Scanner web d'URL

Hydra

Cracker de mdp

Postman

Plateforme pour communiquer à une adresse Web / API

nfs

Protocole de partage de fichiers d'une machine sur un réseau

ftp

Protocole de partage de fichiers via TCP/IP

Technologies utilisées

SSH

Connection au shell d'un machine via clé SSH ou mdp

Rockyou

Wordlist connue pour les CTF

Metasploit

Outils de détection de failles de sécurité

Burp Suite

Outil de test de pénétration Web

Nc (netcat)

Utilitaire pour ouvrir des ports

17/27

Flags au total

Fun With Haskell

- Scan nmap (ssh + http) + scan dirb (/submit)
- Haskell reverse shell
- Récupère ssh key + connection to ssh -> 1er flag
- Privilege escalation avec une faille Flask (set FLASK_APP env var to fichier python os.system("bash")
- Root -> 2ème flag

Muso Troglodytarum

- Scan nmap (ftp + ssh + http) + Scan dirb (/assets)
- Dans /assets/style.css -> /l3_B4n4N13r_D3s_MOnT4gN3s
- Postman this URL (to desactivate javascript) and get /intermediary.php?hidden_directory=/L3s_Fru1ts_s0nt_c0NNus_Gen3raL3m3nt_s0Us_l3_N0m_D3_B4N4n3 in Location header received
- In the route, we download picture and “cat” picture gives us user + list of psswd -> ftp crack hydra
- Get blank code language -> ssh private key -> ssh -> 1er flag
- Switching ssh user thanks to hints (text)
- Use vim faille pour rentrer sur root -> 2eme flag

TekPedago

- Scan nmap (ftp + ssh + http) + Scan dirb (/ + /?view=tek)
- Par chance, nous avons trouvé que la route /flag.php existe. Le site est vulnérable au LFI. Nous avons utilisé une LFI qui permet de convertir le code source de la page php en base64 -> flag1
- LFI to access /var/log/apache2/access.log + use Burp Suite to send php reverse shell via “Referer” header -> flag2
- Privilege escalation `sudo /usr/bin/env /bin/bash` to access to root -> flag3
- Cron job in container -> reverse shell -> flag4

The Binding of Cyber

- Scan nmap (thousands of open ports) beginning from port 109 -> next to 23456
- Find nfs port and mount it -> gives us a locked zip file
- JohnTheRipper to crack the zip -> 1er flag + ssh key + open port hint (5000-6500)
- Script to check every ssh open port -> 5555 -> 2nd flag
- Nous n'avons pas trouvé le dernier flag, privilege escalation (3eme flag)

The Many-Face God

- Scan nmap (smb open = samba ftp server)
- Smbclient -N (le flag -N = no passwd) -> 1er flag + hints
- Showmount -e IP -> list nfs -> il y a un dossier dispo ! On récupère son contenu. Il y a plein de fichiers de config (notamment config redis). Grâce à “grep -r ‘pass’ .” on récupère un mdp. Et on tombe sur un port ouvert config dans le même fichier.
- Scan le port ouvert. telnet -> AUTH + mdp. C’est une db. Parmis toutes les keys, on trouve le 2eme flag dans “internal flag”.
- Dans la db se trouve une liste named “authlist” où se trouve un mdp et un hint par rapport à rsync en base64
- rsync rsync://rsync-connect@10.10.151.248/files/jaqen/ -> user.txt mais on peut pas y accéder car nous n’avons pas les droits
- On envoie notre id_rsa.pub dans les authorised_keys de la machine attaquée pour qu’on puisse accéder à la machine attaquée en ssh sans mdp (rsync -ahv ~/.ssh/id_rsa.pub rsync://rsync-connect@10.10.151.248:/files/jaqen/.ssh/authorized_keys) -> on a enfin les droits pour lire le fichier user.txt -> 3eme flag
- Y’a un dossier sus “TeamCity” qui tourne sur le port 8111 -> shell port forwarding sur le port 9090 car zeus-admin -> http://127.0.0.1:9090/login.html -> admin access = http://127.0.0.1:9090/login.html?super=1. On nous demande un token d’auth -> on fouille de nouveau dans les fichiers de conf où on trouve plein de token dans les logs ->6080862964149265811.
- Création d’un projet + build conf. + build step + command line -> ssh tunneling -> ssh -i .ssh/id_rsa.pub root@IP
- 4eme flag dans le root.txt

Silence

- Scan nmap (ftp + ssh + http) + Scan dirb (/index.htm, /hidden)
- Dans le /hidden, on a accès à un stats.zip -> on get un fichier .xlsx.gpg locked + clé pgp (.hidden_key) -> add la clé pgp “gpg --allow-secret-key-import --import .hidden-key” + decrypt le fichier “gpg -ClimbersStats.xlsx.gpg > ClimbersStats.xlsx” -> accès à un fichier de user / mdp
- Connection ssh avec le user magnus -> (“find / -type f -name "*flag*" -exec ls -l {} + 2>/dev/null”) /usr/share/httpd/web.txt = 1er flag
- Exec linpeas.sh -> shows that nfs is deployed -> to have access to the nfs, we need to use port forwarding -> mount -> user.flag = 2eme flag
- Nous n’avons pas réussi à trouvé le 3eme flag

Un Pepene

- Scan nmap (ssh + http) + Scan dirb (/phpmyadmin, /wordpress, /server-status, /blog, etc.)
- Wordpress est à jour et n'a pas de failles mais on peut savoir ?user=1 que le user admin existe -> brut force de login avec une wordlist pour le mdp -> on est sur le wordpress admin dashboard -> Reverse shell php -> 1er flag
- Nous n'avons pas réussi à récupérer le second flag. Nous sommes bloqués