Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 19 Bratislava 4

Looking for vulnerabilities in large networks (nmap)

Autor: Michal Greguš
Predmet: Bezpečnosť informačných technológii
Akad. r.: 2022/23

Obsah

Motivácia a ciele zadania

V súčasnosti musia obvykle systémový administrátori alebo devops inžinieri zodpovedať za správne fungovanie, chod a ako aj za bezpečnosť mnohých aplikácii, systémov a služieb. Správa týchto služieb dokáže byť časovo náročná a preto najmä za účelom dosiahnutia čo najvyššej miery bezpečnosti používajú títo odborníci automatizované nástroje na identifikáciu zraniteľností v daných službách.

Avšak tieto nástroje často nie sú schopné identifikovať všetky zraniteľnosti nachádzajúce sa v testovaných službách. Dobrým príkladom je napríklad zraniteľnosť v rámci knižnice WolfSSH s CVE-2022-32073 napriek tomu, že táto zraniteľnosť v NVD (national vulnerability database) CVSS skóre až vo výške 9.8 čo predstavuje kritickú zraniteľnosť [3, 4].

Vzhľadom na fakt, že daná zraniteľnosť bola zverejnená len v lete Júly tohto roku a navyše sa nachádzala len v jednej verzii aplikácie, pokiaľ nám je známe v súčasnosti nie je možné ju identifikovať prostredníctvom známych automatizovaných nástrojov na identifikovanie zraniteľností. Preto sme sa rozhodli využiť nmap scripting engine na vytvorenie vlastného skriptu pomocou ktorého bude možné identifikovať zraniteľnú verziu služby na skenovanom systéme, a tak poskytnúť varovanie administrátorovi služby, ktorý následne môže vykonať jej aktualizáciu, a tak dosiahnuť bezpečné fungovanie danej služby [1, 2, 10, 11].

Nástroj Nmap

Nmap je voľne dostupný open-source nástroj na skenovanie sietí a ich bezpečnostné auditovanie. Mnohí administrátori ho využívajú na inventarizáciu siete, monitorovanie aktívnych hostov a služieb na nich bežiacich prípadne na identifikáciu konkrétnej verzie bežiacej služby na cieľovom systéme [1, 2].

Nmap scripting engine

Samotný nástroj disponuje množstvom rôznych funkcionalít avšak jednou z jeho extrémne dôležitých súčastí je aj nmap scripting engine, ktorý umožňuje používateľom vytvoriť si vlastné skripty v jazyku LUA. Tieto môžu byť následne spustené nástrojom nmap a tak flexibilne rozšíriť jeho funkcionalitu prípadne zautomatizovať vykonávanie niektorých úloh. Medzi možné príklady rozšírenia funkcionality patria napr. schopnosť identifikovať konkrétnu proprietárnu službu bežiacu na vybranom porte, prípadne detekcia backdoorov na cieľovom systéme ale aj exploitovanie vybranej zraniteľnosti [1, 2].

Pre nástroj nmap je momentálne dostupná široká škála skriptov, ktoré sú priamo súčasťou balíčku nástroja nmap a možno ich využiť na rôzne účely. Keďže sa jednotlivé skripty medzi sebou výrazne líšia z hľadiska funkcionality a dopadu na cieľovú službu tak za účelom ich klasifikácie disponuje každý skript poľom, ktoré ho zaraďuje do vybranej kategórie skriptov [1, 2].

Tabuľka 1 Kategorizácia nmap .nse skriptov [1, 2]

Typ kategórie	Opis kategórie
1. Auth	Skripty využívané na bypass autentifikačných údajov
2. Broadcast	Skripty využívané na host discovery pomocou broadcastov na lokálnej sieti
3. Brute	Využívané na brute force útoky formou hádania prihlasovacích údajov na vzdialené serveri
4. Default	Základné skripty používané s prepínačom -sC alebo -A
5. Discovery	Skripty slúžiace na získanie dodatočných informácii pomocou žiadostí k verejne dostupným registrom, SMTP službám a tak pod.
6. DOS	Tieto skripty môžu spôsobiť denial of service cieľovej služby
7. Exploit	Účelom týchto skriptov je aktívne exploitovať vybranú zraniteľnosť
8. External	Tieto skripty môžu odosielať dáta third-party databázam, príkladom *whois-ip*
9. Fuzzer	Tieto skripty sú nesmierne náročné na potrebnú šírku komunikačného pásma nakoľko odosielajú randomizované dáta s cieľom identifikovať neobjavený bug alebo zraniteľnosť v cieľovom SW
10. Intrusive	Tieto skripty nesmú byť klasifikované ako safe, lebo napr. nadmerne vyťažujú CPU cieľového stroja alebo môžu spôsobiť pád služby
11. Malware	Slúžia na otestovanie, či cieľový stroj nie je infikovaný nejakým malwarow alebo neobsahuje nejaký backdoor
12. Safe	Tieto skripty by nemali spôsobiť pád služby alebo odoslať obrovské množstvo dát na sieť, rovnako by nemali ani exploitovať akékoľvek zraniteľnosti. Skripty, ktoré tieto podmienky nespĺňajú by mal byť uvedené v kategórii intrusive.
13. Version	Tieto skripty sú rozšírením nmap funkcionality slúžiacej na detekciu verzie služby na ich spustenie je možné použiť prepínač -sV
14. Vuln	Tento typ skriptov sa snaží identifikovať prítomnosť konkrétnej zraniteľnosti na cieľovom stroji

Pri skenovaní hostov resp. portov konkrétneho hosta je nutné zvoliť vhodnú kategóriu skriptov. V prípade penetračného testu alebo skenu kritickej produkčnej infraštruktúry nie je vhodné používať intrusívne skripty, ktoré by mohli spôsobiť pád služby alebo v podstate byť ekvivalentom DOS útoku na danú službu. Ale naopak je vhodné použiť skôr skripty z kategórie safe, prípadne vuln ktoré často slúžia práve na overenie prítomnosti vybranej zraniteľnosti v cieľovom systéme alebo version, ktoré sa využívajú na identifikáciu verzie vybranej služby. Rovnako ak napríklad nechceme zverejniť údaje

o našom skenovaní resp. zanechať čo najmenšiu stopu je vhodné vyhnúť sa skriptom, ktoré zverejňujú informácie počas svojej činnosti. Príkladom je napr. whois skript, ktorý musí poskytnúť informáciu o cieľovej IP adrese regionálnemu whois registru [1, 2, 11].

Okrem skriptov, ktoré sú štandardne súčasťou nástroja nmap si môže v podstate každý používateľ nástroja napísať vlastné skripty, prípadne prevziať dostupné existujúce skripty od iných autorov. Tu je dôležité podotknúť, že nmap skripty nie sú spúšťané v rámci žiadneho sandboxu a preto spúšťanie skriptov od neoverených autorov v prípade, že obsahujú škodlivý kód môže viesť k poškodeniu stroja na ktorom sú spúšťané prípadne k úniku citlivých informácii. Z tohto dôvodu sa odporúča využívať len skripty z overených zdrojov alebo minimálne vykonať kontrolu obsahu resp. preskúmať činnosti vykonávané v rámci používaných skriptov od neoverených autorov pred ich spustením [1, 2, 10, 11].

V rámci nášho zadanie plánujeme vytvoriť skript, ktorý bude pomocou fingerprintingu identifikovať zraniteľnú službu a preto vytvoríme skript, ktorý bude spadať do kategórie vuln, resp. safe [1, 2, 10, 11].

Typy nmap skriptov

Ďalší typ kategorizácie nmap skriptov je na základe ich typu resp. na základe fázy skenovanie cieľového systému v ktorej sa využívajú. Pričom niektoré skripty môžu patriť aj do niekoľkých z nasledujúcich kategórii súčastne [2, 10, 11]. Kategórie skriptov [2]:

1. **Prerule skripty –** tieto skripty možno identifikovať tak, že obsahujú *prerule* funkciu. Sú spúšťané pred samotnou skenovacou fázou nástroja nmap, teda predtým, ako nmap nadobudol akékoľvek dáta o cieľoch skenovania. Tieto skritpy možno využiť na vygenerovanie zoznamu IP adries cieľových hostov na základe doménového mena cieľa. Vtedy pomocou tohto skriptu dokáže nástroj nmap využiť DNS-ZONE-TRANSFER na získanie rozsahu IP adries a následne ich pridať do zoznamu hostov nad ktorými bude vykonané skenovanie.
2. **Host skripty –** samotné skripty sú spúšťané na každý cieľový host, ktorý spĺňa podmienku *hostrule* funkcie. Tento typ skenu sa vykoná až po tom čo nmap vykonal host discovery, port scanning, version detection aj OS detekciu na cieľovom stroji a teda možno vyhodnotiť splnenie podmienky *hostrule*.
3. **Service skripty –** identifikovať ich možno na základe toho, že obsahujú *portrule* funkciu, ktorá určuje, voči ktorej službe by mal byť skript spustený. Teda samotný skript je spúšťaný voči službe, bežiacej na konkrétnom porte na cieľovom stroji.
4. **Postrule skripty –** tieto obsahujú *postrule* funkciu a sú spúšťané až potom čo nástroj nmap dokončil skenovanie všetkých cieľových hostov. Obvykle sa využívajú už len na vhodné formátovanie výstupu nástroja nmap. Toto umožňuje pri skene viacerých hostov napr. vypísať zoznam všetkých strojov, kde je použitý rovnaký ssh kľúč, alebo kde beží rovnaká služba.

Keďže v rámci nášho zadanie je cieľom identifikovať vybranú zraniteľnú službu, tak budeme v rámci zadania vytvárať skript z kategórie service teda s využitím portrule funkcie.

Vytvorenie vlastného .nse skriptu

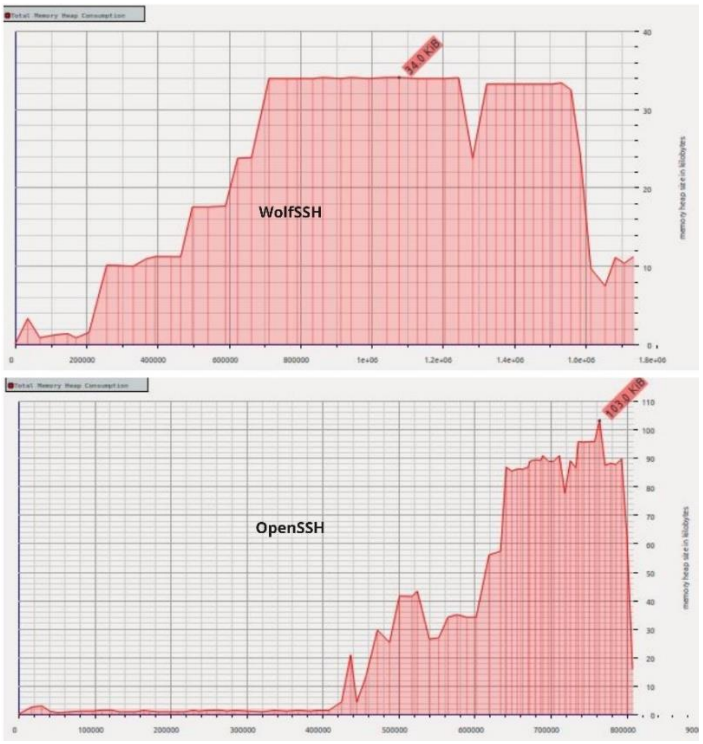
Na vytvorenie vlastného .nse skriptu je možné použiť skriptovací jazyk LUA. Na obrázku je možné vidieť napísanú kostru portrule skriptu, do ktorej možno doplniť akcie ktoré majú byť vykonané aby bolo pomocou daného skriptu možné detegovať vybranú zraniteľnosť. Od tohto hrubého návrhu skriptu sa budeme odvíjať v rámci implementácie vlastného .nse skriptu. Následne je možné vytvorený skript skúšať a debugovať spustením nástroja nmap napríklad pomocou nasledovného príkazu: nmap –script=skript.nse *ip_adresa* -p *cislo portu* **–script-trace -dd** [10, 11]**.**

```lua
--importovane kniznice jazyka LUA pripadne nse kniznice
local string = require "string"
local stdnse = require "stdnse"

--metadata skriptu - popis, informacie o autorovi, licencia
description = [[
Attempts to determine if service is vulnerable to CVE-2022-32073
]]
author = "Michal"
license = "GPL 2.0"
categories = {"vuln"}


--podmienka, ktora musi byt splnena aby sa vykonali prikazy v ramci funkcie action
--v nasom pripade moze byt podmienka napr. ci na vybranomporte bezi sluzba ssh pripadne ssh vo verzii 2
portrule = shortport.http

--funkcia action ma dva argumenty host a port, ktore v ramci nej mozme pouzit na vykonanie potrebnych akcii
--za ucelom urcenia ci je skenovana aplikacia zranitelna na vybrane CVE
action = function(host, port)

  --Vykonanie potrebnej akcie napr. fingerprinting sluzby

  --formatovanie vystupu vystup obsahuje priamo vypis ci je alebo nie je aplikacia zranitelna na dane CVE
  local out = {}
  table.insert(out, string.format("Host : %s (%s)", host.ip, host.name))
  table.insert(out, string.format("Port : %s", port.number))
  table.insert(out, string.format("Host is vulnerable to CVE-2022-32073"))

  return stdnse.format_output(true, out)
end
```

Obrázok 1 Kostra nmap portrule .nse skriptu v jazyk LUA, ktorá môže byť up ravená podľa potreby na vykonanie želaného skenu [10, 11].

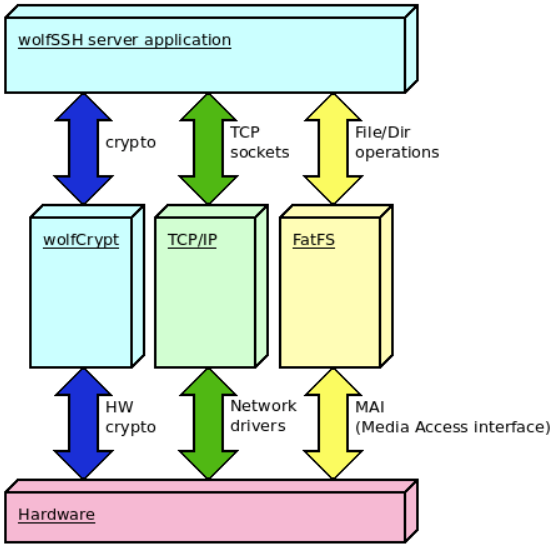
WolfSSL a WolfSSH

WolfSSL je vnorená SSL knižnica vhodná pre vnorené systémy a systémy s obmedzenými výpočtovými zdrojmi. Je to open-source, royalty-free knižnica s výbornou cross-platformovou podporou. A aj napriek malej veľkosti dosahuje vysoké pracovné rýchlosti a podporuje súčasné štandardy ako TLS 1.3 a SSL 3.0. Pričom využíva Wolfcrypt kryptografickú knižnicu, ktorá bola validovaná a spĺňa štandard FIPS140-2 [6, 7, 8].



Obrázok 2Porovnanie využitia pamäti knižniciou WolfSSH s OpenSSH. Knižnica WolfSSH spotrebúva podstatne menej operačnej pamäti. Obrázok upravený z https://www.wolfssl.com/wolfssh-sftp-performance/

Súčasťou tejto knižnice je aj vnorený SSH server. WolfSSH server je secure shell server napísaný v jazyku C podporujúci SSHv2. Pričom jednou z jeho hlavných predností je, že umožňuje spustenie na rôznych platformách. Taktiež poskytuje podporu pre rôzne typy šifrovania ako Poly1305, ChaCha20 alebo NTRU. Okrem secure shell funkcionality server podporuje aj službu FTP resp. SFTP. Táto služba umožňuje prácu s adresármi a súbormi [6, 7, 8].



Obrázok 3 Architektúra aplikácie s využitím FatFS umožňujúca vzdialenú správu s využitím secure shell ako aj vzdialenú prácu z FS vďaka SFTP. Obrázok voľne prevzatý z https://www.wolfssl.com/wolfssh-with-fatfs/

Identifikovaná zraniteľnosť vo WolfSSH

V rámci služby SFTP v knižnici WolfSSH vo verzii 1.4.7 bola identifikovaná zraniteľnosť typu integer overflow (CWE 190). Táto zraniteľnosť sa nachádzala v rámci funkcie wolfSSH_SFTP_RecvRMDIR. Tejto zraniteľnosti bolo priradené CVE-2022-32073 s CVSS skóre (verzia 3) 9,8 čo predstavuje kritickú zraniteľnosť. Po hlbšej analýze bolo však pôvodné CVSS skóre znížené na hodnotu 7,5 teda stále dosahovalo pomerne vysokú hodnotu. Dôvodom bolo najmä, že daná zraniteľnosť mala dopad na všetky tri elementy CIA triády a navyše exploitácia zraniteľnosti nevyžadovala žiadne špeciálne privilégiá ani sa nevyznačovala nejakou zásadnou mierou sofistikovanosti [3, 4, 5, 6].

Táto zraniteľnosť bola opravená 28.2.2021 vo vyššej verzii WolfSSH knižnice a na jej odstránenie bola nutná aktualizácia verzie knižnice. Zraniteľnosť bola spôsobená nevhodnou kontrolou veľkosti premennej typu word32 (int32) ako možno vidieť na obr. Ako možno vidieť zo zobrazeného code snippetu a ako aj vyplýva z definície CWE 190. Tak v SW implementácii je vykonávaná operácia, ktorá predpokladá, že výsledná hodnota bude väčšia ako pôvodná hodnota resp. v tomto prípade, že ak bude súčet *sz* a *idx* > *maxSz* tak bude splnená podmienka. Avšak v tomto prípade môže byť výsledná hodnota aj záporná, keďže v prípade príliš veľkého vstupu dôjde k pretečeniu maximálnej hodnoty premennej typu word32 (int32) a teda dosiahnutiu neželanej akcie v rámci kontrolného toku vykonávaného programu [5, 6].

```
if (sz + idx > maxSz) {
if (sz > maxSz - idx) {
    return WS_BUFFER_E;
}
```

Obrázok 4 Oprava zraniteľnosti integer overflow v rámci funkcie RecvRMDIR [6]

V prípade knižnice WolfSSH a jej identifikovanej zraniteľnosti v rámci funkcie SFTP_RecvRMDIR bolo možné obísť kontrolu maximálnej dĺžky dátovej časti paketu s príkazom na odstránenie priečinku pomocou SFTP. Čím bolo teoreticky možné dosiahnuť úpravu resp. zmazanie priečinku v rámci služby SFTP pri odoslaní paketu s príliš veľkou dátovou časťou. Napriek tomu, že je útočník schopný vymazať priečinok na cieľovom systéme, tak daná zraniteľnosť neumožňuje kontrolovať, ktorý priečinok zmaže resp. nie je priamo možné zvoliť si cieľový priečinok. Toto značne limituje možnosti potenciálneho útočníka a bolo to aj jedným s dôvodov zníženia CVSS skóre pre danú zraniteľnosť [6].

Doposiaľ neexistuje žiadny verejne dostupný exploit pre danú zraniteľnosť. A rovnako sa mi ani nepodarilo nájsť informácie, že by sa niekomu podarilo túto zraniteľnosť úspešne exploitovať. Jedným z dôvodom môže byť, že zraniteľnosť bola prítomná len v spomenutej verzii aplikácie a bola pomerne rýchlo identifikovaná a odstránená v nasledujúcej verzii.

Návrh riešenia

V tejto časti predstavíme návrh riešenia, ktoré bude schopné identifikovať danú zraniteľnosť prostredníctvom nástroja NMAP, s využitím vytvoreného .nse skriptu. Vytvorený ,nse skript bude využívať fingerprinting na identifikáciu služby a na základe jej verzie vyhodnotí prítomnosť zraniteľnosti. Konkrétne bude schopný identifikovať či je konkrétna služba na ktorú bude spustený zraniteľná na CVE-2022-32073, teda či je to bežiaca služba WolfSSH v 1.4.7. Pričom zároveň musí tento skript patriť do kategórie *safe* skriptov a v prípade spustenie nespôsobiť znefunkčnenie skenovanej služby [2, 3, 4].

Spôsob detekcie zraniteľnosti

Keďže samotná zraniteľnosť nevychádza z konfigurácie WolfSSH servera alebo samotnej služby SFTP v rámci neho a ani jej exploitovaniu nemožno predísť zmenou konfigurácie WolfSSH serveru. Preto sme sa rozhodli, že bude postačujúce vykonať fingerprinting (identifikáciu verzie) danej služby na to aby sme na základe jej verzie (1.4.7) bolo schopný prehlásiť, že je zraniteľná na CVE-2022-32073 [3, 4].

Python program

Pre zjednodušenie procesu skenovanie resp. identifikovania prítomnosti zraniteľnosti CVE-2022-32073 na vybranej bežiacej službe na cieľovom stroji sme sa rozhodli vytvoriť python program, ktorý pomocou funkcie subprocess.run() spustí nmap s vytvoreným skriptom a tak a následne pre používateľa vypíše výstup na základe ktorého bude schopný zistiť, či je vybraná služba zraniteľná.

```
45  def main():
46      args = process_arguments()
47      port_specif = "-p" + str(args.port)
48
49      print('Running a scan on "'+args.address+':'+str(args.port)+'" for CVE xy')
50
51      try:
52          compl_process = subprocess.run( ["nmap", "--script","wolf_ssh.nse",port_specif,args.address], capture_output=True, check=True)
53          print(compl_process.stdout)
54
55          is_vulnerable(compl_process)
56      except:
57          print("Failed to execute nmap scan!")
58          return 1
59
60      print('Scan succesfully completed!')
61
62
63      return 0
```

Obrázok 5 Code snippet python programu. Spustenie nástroja nmap s vybranými argumentami pomocou funkcie run().

Použitie programu

Výsledný program je možné spustiť na ľubovoľnom stroji na ktorom je nainštalovaný python verzie 3 (samozrejme aj potrebné knižnice subprocess, argparse...) a zároveň nástroj nmap. Následne je možné spustiť program z príkazového riadku prostredníctvom príkazu:

python3 program.py -a 192.168.100.21

```
└─$ python3 program.py --help
usage: Custom nmap vuln scanner [-h] -a IP adress/range [-p Port]

Program checks whether a service running on host is vulnerable for CVE-2022-32073 in WolfSSH

options:
  -h, --help            show this help message and exit
  -a IP adress/range, --address IP adress/range
                        Enter the IP adress of the target machine or range of IPs to be scanned
  -p Port, --port Port  Enter the Port to be scanned. Default port is 22 222
```

Obrázok 6 Použitie vytvoreného python programu na skenovanie cieľovej služby.

Bibliografia

1. Nmap scripting engine (NSE): Nmap network scanning. Nmap Scripting Engine (NSE) | Nmap Network Scanning. (n.d.). Retrieved November 13, 2022, from https://nmap.org/book/man-nse.html
2. Usage and examples: NMAP network scanning. Usage and Examples | Nmap Network Scanning. (n.d.). Retrieved November 13, 2022, from https://nmap.org/book/nse-usage.html#nse-categories
3. CVE-2022-32073 Detail NVD. WolfSSH v1.4.7 was discovered to contain an integer overflow via the function wolfSSH_SFTP_RecvRMDIR. Retrieved November 13, 2022, from https://nvd.nist.gov/vuln/detail/CVE-2022-32073
4. Vulnerability details : CVE-2022-32073. CVE. (n.d.). Retrieved November 13, 2022, from https://www.cvedetails.com/cve/CVE-2022-32073/
5. CWE - 190 : Integer overflow or wraparound. CWE 190 Integer Overflow or Wraparound. (n.d.). Retrieved November 13, 2022, from https://www.cvedetails.com/cwe-details/190/cwe.html
6. wolfSSL. (n.d.). Asan SFTP fixes by ejohnstown · pull request #360 · Wolfssl/Wolfssh. GitHub. Retrieved November 13, 2022, from https://github.com/wolfSSL/wolfssh/pull/360
7. wolfSSL. (n.d.). Wolfssl/Wolfssl: The Wolfssl Library is a small, fast, portable implementation of TLS/SSL for embedded devices to the cloud. Wolfssl supports up to TLS 1.3! GitHub. Retrieved November 13, 2022, from https://github.com/wolfSSL/wolfssl
8. wolfSSL. (n.d.). Wolfssl/wolfssh: Wolfssh is a small, fast, portable SSH implementation, including support for SCP and SFTP. GitHub. Retrieved November 13, 2022, from https://github.com/wolfSSL/wolfssh
9. Introduction. 1. Introduction - wolfSSH Manual. (n.d.). Retrieved November 13, 2022, from https://www.wolfssl.com/documentation/manuals/wolfssh/index.html
10. *Script writing tutorial: Nmap network scanning*. Script Writing Tutorial | Nmap Network Scanning. (n.d.). Retrieved November 20, 2022, from https://nmap.org/book/nse-tutorial.html
11. *Extending nmap with lua*. Blog post RSS. (n.d.). Retrieved November 20, 2022, from https://citizen428.net/blog/extending-nmap-with-lua/