VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SKENOVANIE PRIEMYSELNÝCH ZRANITEĽNOSTÍ SKUPINA 14

AUTOR PRÁCE **BC. PETER TAKÁCS, BC. ROBERT RUMAN, BC. MARTIN ŽIGRAI, BC. MICHAL ČESKÝ**

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Skenovanie priemyselných zraniteľností

Navrhnite a naprogramujte webovú aplikáciu, ktorá bude vykonávať skenovanie zariadení v sieti (napr. pomocou nástroja nmap). Na základe získaných informácií bude ďalej aplikácia vyhľadávať zraniteľnosti v CVE databázach [1, 2, 3]. Súčasťou aplikácie bude vlastná databáza, kde budú uložené informácie o vykonaných skenoch. Databáza bude obsahovať minimálne položky:

- časové razítko skenu,
- názov/ID skenu,
- pre každé zistené zariadenie:
  - IP adresa,
  - ID/Sériové číslo zariadenia,
  - typ zariadenia,
  - firmware zariadenia,
  - otvorené porty,
  - zistené zranitelnosti:
    - názov,
    - typ,
    - CVSS,
    - odkaz na zranitelnosť.

Výstupom projektu bude vyššie uvedená aplikácia, ktorá bude overená na aspoň jednom priemyselnom zariadení.

Projekt naprogramujte vo Vami zvolenom programovacom jazyku s využitím dostupných knižníc. Skenovanie bude primárne vykonávané na zariadeniach od spoločnosti Siemens, ktoré budú dostupné v laboratóriách alebo cez vzdialený prístup.

Obsah

1 Úvod

V súčasnej dobe, keď sa priemyselné zariadenia a siete stávajú čoraz viac prepojenými a digitalizovanými, bezpečnosť týchto systémov sa stáva prioritou. S cieľom identifikovať a zabezpečiť tieto systémy proti potenciálnym hrozbám, je nevyhnutné pravidelne skenovať ich zraniteľnosti a vyhľadávať možné slabiny v ich zabezpečení.

Hlavným cieľom tohto projektu je navrhnúť a naprogramovať webovú aplikáciu, ktorá by umožňovala skenovanie priemyselných zariadení, ktoré sú integrované v sieti. Vhodným nástrojom, ktorý by toto skenovanie vykonával je nástroj NMAP, ktorý je schopný identifikovať aktívne zariadenia a otvorené porty v sieti. Aplikácia pomocou získaných informácií bude následne schopná analyzovať a vyhľadať potenciálne zraniteľnosti týchto zariadení.

Tieto údaje budú poskytovať komplexný prehľad o bezpečnostnom stave siete a budú umožňovať správcom systémov rýchlo reagovať a implementovať opatrenia na zabezpečenie siete.

1.1 Ciele projektu

Výstupom projektu je vytvorenie kontrolnej štúdii, webovú aplikáciu, ktorá bude schopná skenovať priemyselné zariadenia pomocou nástroja Nmap a následne dokumentáciu k tejto aplikácii.

Cieľom kontrolnej štúdie je zoznámiť sa s problematikou zadania projektu a navrhnúť riešenie, ako vytvoriť danú webovú aplikáciu s databázou, ktorá bude slúžiť na skenovanie priemyselných zariadení.

Zámer samotného projektu je vytvoriť webovú aplikáciu, ktorá bude mať na hlavnej stránke pole na zadanie IP adresy zariadenia, na ktorom sa budú skenovať otvorené porty a následne pomocou nich odhaľovať bezpečnostné zraniteľnosti. Vložená IP adresa bude vložená do Python skriptu, kde pomocou knižnice Nmap, ktorá je vytvorená špeciálne pre Python, bude zariadenie oskenované. Po vykonaní skenovania budú výsledky skenu uložené do MySQL databázy. Vo webovom prostredí bude možné si zobraziť predošlé výsledky skenovania a pomocou týchto informácií bude aplikácia vyhľadávať zraniteľnosti v CVE databázach.

2 Kontrolná štúdia

2.1 Komunikačné zariadenia v priemysle

Priemyselné zariadenia sú kľúčovými komponentmi v mnohých odvetviach, vrátane energetiky, vodného hospodárstva, dopravy, chemického, ropného a plynárenského priemyslu. V týchto odvetviach je zamestnané veľké percento populácie a ohrozenie niektorých z týchto odvetví by mohlo mať závažné ekonomické a bezpečnostné následky. Ich úlohou je riadiť a monitorovať rôzne procesy a zhromažďovať dáta v reálnom čase z automatizovaných riadiacich komponentov. Napríklad v energetickom odvetví sa tieto zariadenia používajú na monitorovanie a riadenie prevádzky elektrární, distribúciu elektrickej energie a optimalizáciu výkonu.

Základné komponenty, ako sú SCADA, DSC, PLC, RTU a IED, sú dôležité pre úspešné fungovanie priemyselných systémov. SCADA systémy umožňujú monitorovanie a riadenie distribuovaných procesov, zatiaľ čo PLC sú zodpovedné za vykonávanie konkrétnych riadiacich úloh na základe zozbieraných údajov. RTU a IED potom umožňujú komunikáciu so vzdialenými zariadeniami a ich inteligentnú integráciu do celkového systému [4].

Výzvou pri komunikácii medzi týmito zariadeniami je zabezpečiť rýchlu a spoľahlivú výmenu údajov. Moderné bezdrôtové technológie a riešenia s kybernetickou bezpečnosťou sú nevyhnutné na zvýšenie dostupnosti a bezpečnosti týchto systémov. Špecifické sieťové protokoly, ako sú PROFINET, OPC UA, PROFIBUS a INTERBUS, sú navrhnuté tak, aby poskytovali efektívnu komunikáciu medzi priemyselnými zariadeniami a zabezpečili ich spoľahlivú integráciu do siete.

Takýto pokročilý prístup k riadeniu a monitorovaniu priemyselných procesov je kľúčom k zlepšeniu výkonnosti, bezpečnosti a efektivity v mnohých odvetviach. Rýchly vývoj technológií v tomto sektore znamená neustále zlepšovanie a inovácie, aby sa zabezpečilo, že priemysel bude schopný efektívne reagovať na budúce výzvy a požiadavky.

Komunikačné protokoly priemyselných zariadení sa delia do dvoch hlavných kategórií a tými sú protokoly komunikácie otvorených systémov a protokoly uzavretých systémov. Komunikačné protokoly uzavretých systémov nie sú verejne dostupné a nie sú interoperabilné s produktmi iných výrobcov. Komunikačné protokoly otvorených systémov slúžia na zabezpečenie komunikácie produktov, ktoré majú rovnaké funkcie, nezávisle od značky.

Tieto protokoly sa viažu na špecifické sieťové porty, ktoré definujú konkrétny proces alebo službu. Napríklad protokol PROFINET využíva UDP porty 34 962, 34 963, 34 964 a 53 247 pre rôzne účely ako je komunikácia, konfigurácia alebo diagnostika. Protokol OPC UA využíva porty 4840 a 4843 a protokol MODBUS využíva port 502 [5].

2.2 Skenovanie siete pomocou nástroja NMAP

Python Nmap je knižnica jazyka Python, ktorá umožňuje vývojárom programovo komunikovať s nástrojom na skenovanie siete Nmap. Samotný Nmap je výkonný open-source nástroj používaný na prieskum siete a bezpečnostný audit. Je určený na na odhaľovanie zariadení, služieb a zraniteľností v počítačových sieťach [6].

Python Nmap poskytuje rozhranie k funkciám nástroja Nmap a umožňuje vývojárom automatizovať úlohy skenovania siete, získavať výsledky skenovania a vykonávať rôzne činnosti prieskumu siete pomocou skriptov.

Spôsob, akým nmap pracuje, spočíva v odosielaní surových paketov IP (raw IP packet) s cieľom zistiť, či je cieľový hostiteľ dostupný v sieti, aké služby poskytuje, aký typ firewallov sa používa, aké verzie operačného systému sú spustené a mnoho ďalších charakteristík.

2.3 Zraniteľnosti priemyselných zariadení

Zraniteľnosti priemyselných zariadení predstavujú bezpečnostné riziká spojené s automatizovanými systémami a zariadeniami používanými v priemysle. Tieto zraniteľnosti môžu vychádzať z chýb v softvéri, nedostatočných bezpečnostných opatreniach alebo zastaraných technológiách. Identifikujú sa pomocou CVE (Common Vulnerabilities and Exposures) databáz, ktoré priradzujú unikátne identifikátory zraniteľností. CVSS (Common Vulnerability Scoring System) poskytuje škálu na hodnotenie závažnosti zraniteľností na základe rôznych kritéri.

2.4 CVE databázy

Databázy CVE, skratka pre Common Vulnerabilities and Exposures databases, sú úložiská, ktoré uchovávajú informácie o známych bezpečnostných zraniteľnostiach softvérových a hardvérových produktov. Tieto zraniteľnosti sú identifikované, katalogizované a je im pridelený jedinečný identifikátor nazývaný CVE ID. ID CVE pozostáva z predpony "CVE", za ktorou nasleduje rok a jedinečné číslo, napríklad CVE-2024-1234.

Databázy CVE sú základným zdrojom informácií na riadenie rizík kybernetickej bezpečnosti, informovanie o nových hrozbách a ochranu systémov a údajov pred potenciálnymi bezpečnostnými zraniteľnosťami a zneužitiami.

Ako systém CVE funguje:

Na program CVE dohliada spoločnosť MITRE Corporation s finančnými prostriedkami Agentúry pre kybernetickú bezpečnosť a bezpečnosť infraštruktúry (CISA), ktorá je súčasťou Ministerstva vnútornej bezpečnosti USA. Záznamy CVE sú stručné. Neobsahujú technické údaje ani informácie o rizikách, vplyvoch a opravách. Tieto údaje sa nachádzajú v iných databázach vrátane Národnej databázy zraniteľností USA (NVD), databázy CERT/CC Vulnerability Notes a rôznych zoznamov vedených dodávateľmi a inými organizáciami. V týchto rôznych systémoch poskytujú identifikátory CVE používateľom spoľahlivý spôsob rozpoznávania jedinečných zraniteľností a koordinácie vývoja bezpečnostných nástrojov a riešení [7].

2.5 CVSS

CVSS je skratka pre Common Vulnerability Scoring System. Je to spôsob hodnotenia a klasifikácie nahlásených zraniteľností štandardizovaným spôsobom, a jej cieľom je pomôcť porovnať Popzraniteľnosti v rôznych aplikáciách. CVSS vytvára skóre od 0 do 10 na základe závažnosti zraniteľnosti[8].

Tabuľka 1: Hodnotenie zraniteľnosti pomocou CVSS

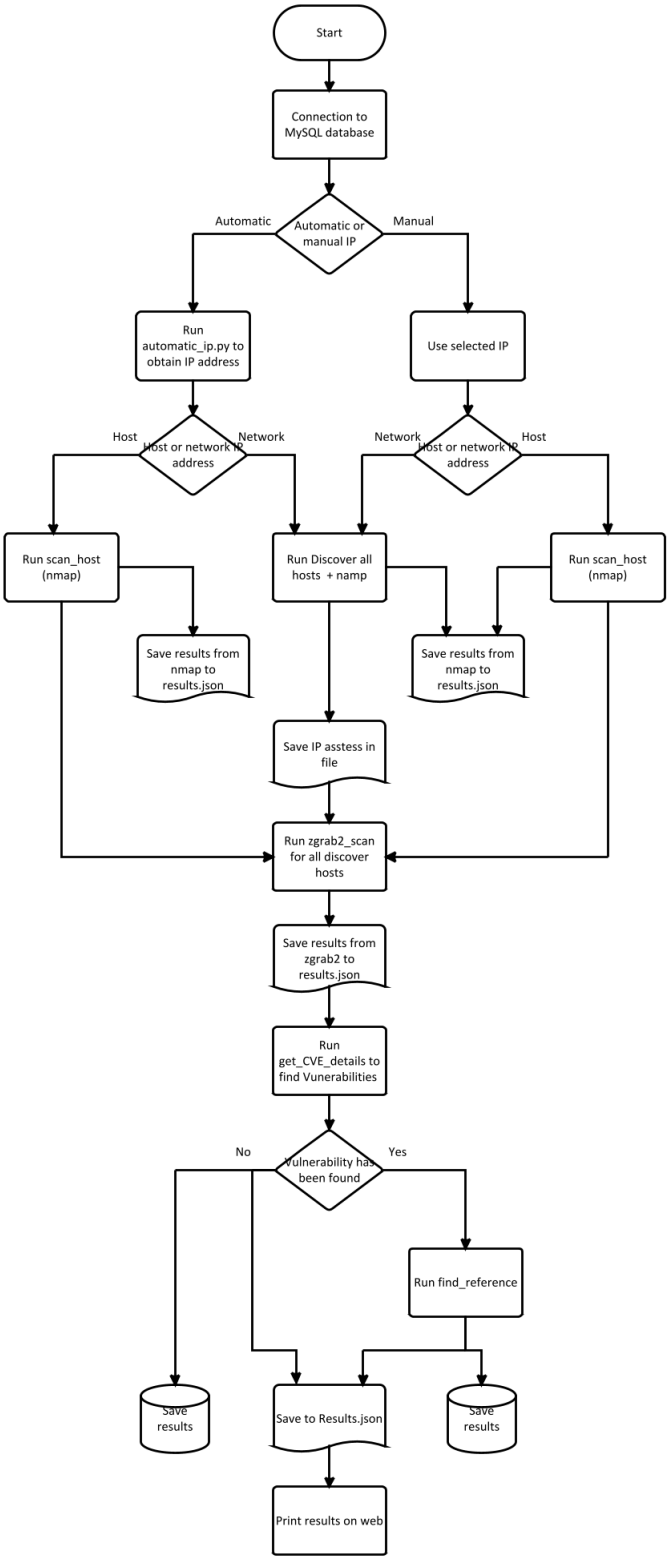
CVSS Base Score	CVSS Severity Level
0	None
0.1 - 3.9	Low
4.0 - 6.9	Medium
7.0 - 8.9	High
9.0 - 10	Critical

Pri generovaní skóre CVSS sa berie do úvahy viacero faktorov, medzi ktoré patria:

- **Attack vector (Vektor útoku)** – týkajúci sa spôsobu, akým môže útočník získať prístup k danému systému (možno priradiť štyri rôzne hodnoty - sieťový, susedný, lokálny, fyzický).
- **Attack complexity (Zložitosť útoku)** – ako ťažké je zneužiť zraniteľnosť (dve možné hodnoty sú nízka alebo vysoká).
- **Privileges required (Požadované oprávnenia)** – uvádza oprávnenia, ktoré útočník potrebuje na zneužitie zraniteľnosti:
  - Žiadne - žiadny prístup k žiadnym súborom alebo nastaveniam,
  - Nízke - základné možnosti používateľa,
  - Vysoké - sú potrebné oprávnenia na úrovni administrátora.

- **User interaction (Interakcia používateľa)** – definuje, či je na úspešné zneužitie zraniteľnosti potrebný používateľ (Žiadny alebo Požadovaný), ak nie je potrebný žiadny používateľ, vplyv na skóre CVSS je najvyšší.
- **Scope (Rozsah)** - ktoré zraniteľnosti môžu byť zneužité a následne použité na útok na iné časti systému alebo siete.
- **Confidentiality (Dôvernosť)** - možnosť neoprávneného prístupu k citlivým informáciám (vysoká, nízka, žiadna).
- **Integrity (Integrita)** - potenciál neoprávnenej modifikácie, narušenia alebo vymazania citlivých informácií.
- **Availability (Dostupnosť)** - možnosť odmietnutia prístupu pre oprávnených používateľov.

3 Implementačné riešenia



Obr. 1: Vyvojovy diagram.

Po spustení programu sa najprv overí pripojenie k databáze. Ak je pripojenie úspešné, používateľovi sa zobrazí hlavná ponuka webového prehliadača, v ktorej si môže vybrať, či chce IP adresu na skenovanie zadať ručne, alebo si ju nechá vygenerovať automaticky. Ak si používateľ vyberie druhú možnosť, spustí sa súbor automatic-ip.py s funkciou get-subnet().

Ak je operačný systém používateľa Windows, potom sa funkcia get-windows-if-list() modulu importovaného zo súboru scapy.arch pokúsi získať zoznam všetkých sieťových adaptérov v systéme, pričom filtruje napríklad lik-local adresy. V prípade operačného systému Linux alebo macOS program najprv získa názov prvého sieťového adaptéra pre Ethernet (funkcia get-first-eth-adapter()), ako aj pre Wifi (funkcia get-first-wifi-adapter()). Ak takýto adaptér existuje, získa z neho zoznam adries IP a masiek podsiete, ktoré použije na ďalšie skenovanie.

V prípade, že používateľ zvolí prvú možnosť, že chce IP adresu zadať sám, funkcia execute-scan() overí, či používateľ zadal iba IP adresu určitého hostiteľa alebo adresu siete s určitou maskou.

Súbor scan.py spracováva samotné skenovanie adries IP. Ak bola zadaná iba adresa jedného hostiteľa, spustí sa funkcia scan-host(), ktorá pomocou funkcie nmap prehľadá všetkých 65535 portov TCP a uloží ich do premennej výsledok vo formáte json.

Ak je zadaná adresa IP vrátane masky, spustí sa funkcia complete-network-scan(), ktorá pomocou funkcie nmap zabezpečí skenovanie celej siete s cieľom získať všetky adresy IP hostiteľov, ktorí sa v tejto sieti nachádzajú. Tieto adresy IP sa potom uložia do súboru ip-range.txt. Tieto adresy IP sa pomocou nástroja nmap prehľadajú na účely zistenia otvorených portov, operačného systému a verzií služieb. Zistené informácie sa pridajú do zoznamu výsledkov zariadenia. Pre každý zistený otvorený port sa do zoznamu výsledkov OpenPorts pridá záznam. Na vykonanie kontroly nmap sa zavolá funkcia run-zgrab2-scan() na kontrolu IP adries hostiteľov uložených v súbore ip-range.txt pomocou nástroja zgrab2. Nakoniec sa zavolá funkcia process-output() na uloženie výsledkov skenovania do výsledkov.

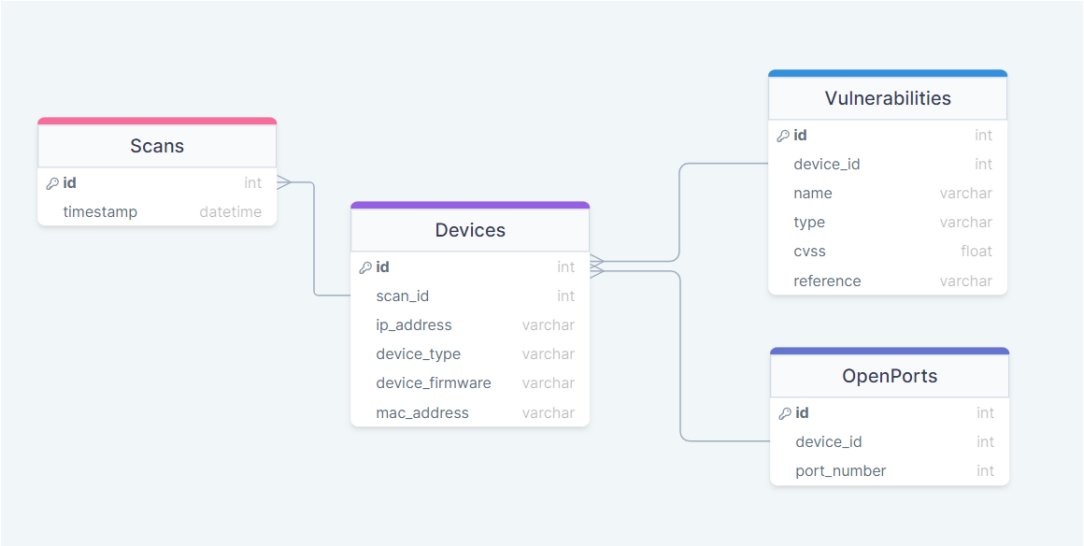
Potom nasleduje súbor cve-data-retrieve.py na zistenie informácií o zraniteľnostiach nastavenia. Najprv sa spustí funkcia CVE-val(), ktorá z výsledkov získa firmvér zariadenia. Ak sa tam nenájde, pridá do zoznamu zraniteľností výsledkov prázdny rad. Ak sa tu nájde firmvér, skontroluje sa, či je pole verzia prázdne. Ak je to tak, zavolá sa funkcia find-all-versions(), aby sa zabezpečilo, že sú prítomné všetky dostupné verzie firmvéru.

Nasleduje funkcia get-cve-details(), ktorá pomocou vzorovej adresy URL vykoná požiadavku HTTP GET na získanie zraniteľností zariadenia podľa firmvéru a verzie. Potrebné informácie, ako sú CVE ID, CVSS, verzia CVSS a odkaz, sa získajú z príslušnej webovej stránky. Referencie sa vytvárajú pomocou funkcie reference-finder() v súbore find¬-reference.py, ktorá vytvorí odkaz na databázu CVE s požadovanou CVE. Všetky novozískané hodnoty sa nakoniec uložia do zoznamu výsledkov Vulnerabilities. Celý obsah premennej results sa ukladá do databázy MySQL. Po uložení sa používateľovi zobrazia všetky hodnoty nájdené v tabuľke. Výpočet hodnôt sa vykonáva zo zoznamu výsledkov. Nakoniec sa všetky údaje v zozname výsledkov uložia do databázy MySQL.

3.1 Databáza

Databáza obsahuje štyri tabuľky: **Scan, Device, OpenPorts, Vulnerabilities**. Medzi Scans – Devices, Devices – OpenPorts a Vulnerabilities existuje vzťah jeden ku mnohým.

Momentálne aplikácia zobrazuje výsledky skenovania v Json formáte. Výsledok je možné vidieť aj v databáze.



Obr. 2: Aktuálny návrh databáze.

Scans:

– scan_id - automaticky generovaná celočíselná hodnota od 1 do n, ktorá predstavuje číslo skenovania.
– Timestamp - pre každé skenovanie sa uloží časová značka, ktorá predstavuje čas vykonania skenovania.

Devices:

- scan_id
- ip_address - táto premenná sa používa na uloženie ip adresy aktuálneho skenovaného zariadenia.
- device_id - automaticky generovaná celočíselná hodnota od 1 do n, ktorá predstavuje číslo zariadenia.
- device_type - premenná na sledovanie, či je zariadenie počítač alebo mobilné zariadenie.
- device_firmware - aký druh softvéru je spustený na aktuálnom zariadení.

Vulnerabilities:

- device_id
- name - názov zraniteľnosti.
- type - typ zraniteľnosti.
- cvss - desatinné číslo z rozsahu od 0 do 10 hodnotiace závažnosť zraniteľnosti (CVSS skóre).
- reference - referencia na zraniteľnosť.

OpenPorts:

- device_id
- port_number - celočíselná hodnota uchovávajúca číslo otvoreného sieťového portu.

3.2 Získávanie subnetu siete

automatic_ip.py

get_subnet

Táto funkcia služi na automatické získanie subnetu zariadenia. Postup funkcie:

1. Kontrola operačného systému. Na základe tejto kontroly sa volí postup.
2. Pri operačnom systéme Windows:
   (a) Prechádza sa zoznamom sieťových adaptérov.
   (b) Kontrola či adresa nepatrí lokálnej linke alebo či sa nejedná o adresu loopback.
   (c) Kontrola či sa jedná o adaptér Ethernet alebo Wifi.
   (d) Na koniec sa zisťuje subnet maska.
3. Pri operačnom systéme Linux:
   (a) Získavajú sa informácie o Ethernet adaptéry, ak je prítomný tak sa získa IP adresa a subnet maska.
   (b) Získavajú sa informácie o WiFi adaptéry, ak je prítomný tak sa získa IP adresa a subnet maska.

3.3 Získávanie informácií z databázy zraniteľností

cve_data_retrieve.py

get_cve_details

Táto funkcia získava podrobnosti CVE (spoločné zraniteľnosti a vystavenia) pre danú verziu firmvéru zariadenia pomocou vstupných parametrov.

Postup funkcie:

1. Konštruuje URL adresu pre podrobnosti CVE na základe poskytnutého firmvéru a verzie, ak nie je poskytnutá vlastná URL adresa.
2. Posiela GET požiadavku na konštruovanú URL adresu.
3. Parsuje HTML odpoveď pomocou BeautifulSoup.
4. Extrahuje relevantné informácie, ako sú identifikátor CVE, skóre CVSS, typ zariadenia a odkazy pre každú uvedenú zraniteľnosť.
5. Konštruuje zoznam slovníkov, pričom každý reprezentuje podrobnosť CVE, a vráti ho.

find_all_versions

Táto funkcia je použitá na nájdenie dostupných verzií firmvéru pre zariadenie

Postup funkcie:

1. Konštruuje URL adresu pre vyhľadávanie verzie na základe poskytnutého firmvéru.
2. Posiela GET požiadavku na konštruovanú URL adresu.
3. Parsuje HTML odpoveď pomocou BeautifulSoup.
4. Extrahuje všetky verzie uvedené v odpovedi.
5. Vráti zoznam verzií.

fetch_cves_for_versions

Získava podrobnosti CVE pre všetky verzie firmvéru.

Postup funkcie:

1. Volá funkciu find_all_versions() pre získanie zoznamu verzií.
2. Ak sú nájdené verzie, prechádza každou verziou.
3. Volá funkciu get_cve_details() pre každú verziu na získanie podrobností CVE.
4. Rozširuje zoznam results.data["Vulnerabilities"] získanými podrobnosťami CVE.

CVE_val

Vstupný bod pre získanie podrobností CVE.

Postup funkcie:

1. Spracováva prípady, keď je firmvér prázdny alebo verzia nie je poskytnutá.
2. Konštruuje vzor na zhodu s identifikátormi CVE.
3. Ak sa vzor zhoduje, pridáva k názvu firmvéru "Firmware".
4. Volá buď funkciu fetch_cves_for_versions() alebo get_cve_details() podľa toho, či je poskytnutá verzia alebo nie.

find_reference.py

reference_finder
Získava odkazy pre daný identifikátor CVE.

Postup funkcie:

1. Konštruuje URL adresu pre CVE.
2. Posiela GET požiadavku na konštruovanú URL adresu.
3. Parsuje HTML odpoveď pomocou BeautifulSoup.
4. Nájde konkrétnu položku obsahujúcu odkazy.
5. Extrahuje odkazy a vráti ich ako zoznam.

3.4 Skenovanie

Pre skenovanie sa používa python-nmap a zgrab2. Nmap sa používa na zistenie aktívnych zariadení v sieti a následne na skenovanie otvorených portov na jednotlivých zariadeniach. Zgrab2 sa používa na skenovanie priemyselných zariadení od firmy Siemens. Zgrab2 je schopný zistiť podrobné informácie o zariadení, ako napríklad sériové číslo, firmware a typ zariadenia.

Postup skenovania:

– Nmap zistí všetky aktívne zariadenia v sieti.
– Nmap po jednom zapíše IP adresu zariadenia do textového súboru a následne skenuje zariadenie, či sú na ňom otvorené porty.
– Získané informácie sa uložia do premennej data v súbore results.py.
– Zgrab2 skenuje všetky IP adresy, ktoré Nmap našiel.
– Dáta, ktoré Zgrab2 zistil, sa pridajú do premennej data v súbore results.py.

3.5 Použité externé knižnice a ich verzie

– Python 3.12.0,
– Python-nmap 0.7.1,

- Flash 3.0.3,
- Jinja 3.1.3,
- Mysql-connector-pyzhion 2.2.9,
- Netifacces 0.11.0,
- Psutil 5.9.8,
- Requests 2.31.0,
- Bs4 0.0.2,
- Zgrab2 2.0.0.

4 Popis spustenia programu

Na spustenie programu je potrebné mať nainštalované všetky externé knižnice, ktoré nainštalujeme pomocou príkazu:

$ pip install -r python-nmap Flask Jinja2 mysql-connector-python psutil netifaces requests beautifulsoup4

V prípade Zgrab2 musíte mať v operačnom systéme nainštalovanú verziu Go 1.17 alebo novšiu. Taktiež je nutné mať nainštalovaný nástroj git a nástroj nmap.

$ apt-get update
$ apt-get install -y git golang-go nmap

Vytvorený priečinok Zgrab2 umiestnite do priečinka projektu. Postup inštalácie nástroja:

$ git clone https://github.com/zmap/zgrab2.git
$ cd zgrab2
$ go build
$ make
$ ./zgrab2

Pred spustením programu je nutné vytvoriť databázu. Databázu je možné vytvoriť a rozbehnúť pomocou nástroja Docker. Dané príkazy je nutné vykonávať zo zložky **docker-databaza**:

$ docker build -t industrial_cve_db .
$ docker run -p 3307:3306 industrial_cve_db

Po úspešnej inštalácii nástroja môžete spustiť hlavný program pomocou nasledujúceho príkazu:

$ sudo python3 main.py

Po spustení programu stačí otvoriť webový prehliadač a do adresy URL zadať adresu Loopback (http://127.0.0.1:5000). Ak chcete skenovať konkrétne zariadenie alebo sieť, zadajte do poľa IP adresa adresu požadovaného zariadenia. Po dokončení skenovania sa výsledky zobrazia priamo vo webovom prehliadači.

5 Záver

Projekt skúma a analyzuje problematiku bezpečnostných zraniteľností v priemyselných zariadeniach s cieľom navrhnúť určité stratégie a nástroje na identifikáciu a riešenie týchto hrozieb. Ako jeden z výstupov bola vypracovaná a odovzdaná kontrolná štúdia, ktorá obsahuje úvod do problematiky projektu a predstavu o samotnom riešení webovej aplikácie. Súčasťou projektu je vytvorenie webovej aplikácie, ktorá slúži ako praktický nástroj na vyhľadávanie zraniteľností na vybranom zariadení alebo na celej sieti. K tejto webovej aplikácii bola vytvorená aj rozsiahla dokumentácia.

Hlavným výstupom projektu je vytvorenie webovej aplikácie, ktorá slúži ako praktický nástroj na vyhľadávanie zraniteľností na zariadeniach. Používateľ má dve možnosti: zadať IP adresu ručne alebo nechať aplikáciu vygenerovať adresu na základe jeho sieťových adaptérov. Je možné zvoliť skenovanie len jedného hosťa alebo celej siete.

Samotné skenovanie sa vykonáva pomocou dvoch nástrojov. Nmap sa v aplikácii používa najmä na získanie informácií o prítomnosti otvorených portov na zariadení. Druhým nástrojom je zgrab2, ktorý sa väčšinou používa na získanie podrobnejších informácií o zariadení, ako je firmvér a jeho verzia.

Po vykonaní skenovania sa informácie zo skenovania použijú na nájdenie zraniteľností. Pri tomto procese sa využíva externá databáza, v ktorej sú uložené informácie o zraniteľnostiach (CVE). V tejto databáze sa výsledky skenovania porovnávajú s dostupnými záznamami CVE, čo umožňuje identifikovať príslušné zraniteľnosti.

Nakoniec sa všetky identifikované informácie uložia do lokálnej databázy a zobrazia sa používateľovi v tabuľke vo webovej aplikácii. Používateľ je tak informovaný o prítomných zraniteľnostiach a môže prijať príslušné opatrenia na zaistenie bezpečnosti zariadení.

Literatúra

[1] *CVE Details [online].* MITRE Corporation, 2022 [cit. 2022-02-08]. Dostupné z: <https://www.cvedetails.com>.

[2] TEAM82 VULNERABILITY DASHBOARD. *Claroty [online].* 2022 [cit. 2022-02-08]. Dostupné z: <https://claroty.com/vulnerability-table/>.

[3] *The Exploit Database [online].* Offensive Security, 2022 [cit. 2022-02-08]. Dostupné z: <https://www.exploit-db.com/>.

[4] GUANGKAI Zhou, JUN Bai, BAILING Wang, SONG Jia. *A Method of Scanning Industrial Control System Equipment.* School of Computer Science and Technology, Harbin Institute of Technology, 2017 [cit. 2024-03-17].

[5] ANUMAK & COMPANY. *Industrial Communication Systems [online].* Medium. 2022-01-21 [cit. 2024-03-17]. Dostupné z: <https://anumakandcompany.medium.com/industrial-communication-systems-ff7dc5cf8fb0.>.

[6] *Python Nmap Module Fully Explained with 8 Programs [online].* [cit. 2024-3-15]. Dostupné z: <https://www.pythonpool.com/python-nmap/>

[7] *What is a CVE? [online].* [cit. 2024-3-15]. Dostupné z: <https://www.redhat.com/en/topics/security/what-is-cve>

[8] RISTO Jonathan. *What is Common Vulnerability Scoring System (CVSS) [online].* [cit. 2024-3-15]. Dostupné z: <https://www.sans.org/blog/what-is-cvss/>