Hadtudományi Szemle
13. évfolyam (2020) 4. szám 161–180. • DOI: 10.32563/hsz.2020.4.13

Paráda István,[1] Fekete Károly,[2] Tóth András[3]

Hálózattérképezés portszkennelési technikák segítségével

Network Discovery Using Port Scanning Techniques

Absztrakt

Jelen cikk a penetrációs tesztek szakaszán belül a felderítés és analízis szintjeinek bemutatásával foglalkozik. A szerzők a hálózatszkennelés és azon belül a portszkennelés technikai kivitelezése egyes lehetőségeinek elemzését mutatják be jelen művükben. A portszkennelési technikák segítenek a támadónak a megcélzott szerver vagy hoszt nyitott portjainak azonosításában. Bemutatják a felderítés során alkalmazható különböző technikai megoldásokat és az általuk kapott lehetséges támadási vektorokat. A szerzők ezen sérülékenységek kihasználhatóságára, valamint veszélyeire szeretnék felhívni a figyelmet, ennek megfelelően készítették el elemző-értékelő művüket.

Kulcsszavak: *hálózatfelderítés, hálózattérképezés, portszkennelés, Nmap, TCP/UDP szkennelés*

Abstract

This article discusses the levels of detection and analysis within the penetration testing phase. In the present work, the authors present an analysis of some possibilities of the technical implementation of network scanning, including port scanning. Port scanning techniques help an attacker identify open ports on a targeted server or host.

[1] Nemzeti Közszolgálati Egyetem, Katonai Műszaki Doktori Iskola, doktorandusz – University of Public Service, Doctoral School of Military Engineering, PhD student, e-mail: paradaistvan@gmail.com, ORCID: http://orcid.org/0000-0002-3083-6015

[2] Nemzeti Közszolgálati Egyetem, Hadtudományi és Honvédtisztképző Kar, egyetemi docens – University of Public Service, Faculty of Military Sciences and Officer Training, Associate Professor, e-mail: fekete.karoly@uni-nke.hu, ORCID: http://orcid.org/0000-0003-4483-5002

[3] Nemzeti Közszolgálati Egyetem, Hadtudományi és Honvédtisztképző Kar, adjunktus – University of Public Service, Faculty of Military Sciences and Officer Training, Assistant Professor, e-mail: toth.hir.andras@uni-nke.hu, ORCID: http://orcid.org/0000-0001-6098-3262

The different technical solutions that can be used during the reconnaissance and the possible attack vectors they provide are presented. The authors would like to draw attention to the exploitability and dangers of these vulnerabilities, and have prepared their analytical-evaluation work accordingly.

Keywords: *Network scanning, network mapping, portscan, Nmap, TCP/UDP scan*

1. Bevezetés

A hálózatfeltérképezés a penetrációsteszt-végrehajtás lépésének második szakasza, amely „életben lévő" és reagáló rendszerekről gyűjt információt a hálózaton. A portszkennelési technikák segítenek a támadónak a megcélzott szerver vagy hoszt nyitott portjainak azonosításában. A rendszergazdák gyakran portszkennelési technikákat használnak a hálózatok biztonsági politikájának ellenőrzésére, míg a támadók ezeket használják a futó szolgáltatások azonosítására egy hoszton, de ők a rendszergazdákkal ellentétben már a hálózat veszélyeztetése céljából.[4]

„A számítógép-hálózati felderítés a hálózatok struktúrájának feltérképezését, az adatbázisokhoz való illetéktelen hozzáférést és a támadható pontok meghatározását jelenti. Megvalósulhat a szemben álló fél számítógépes rendszereibe történő szoftveres vagy hardveres úton való behatolással. Célja az adatbázisokban tárolt adatokhoz, információkhoz való hozzáférés és azok felderítési céllal való hasznosítása, illetve a későbbi károkozással járó támadás kivitelezéséhez a hálózat támadható pontjainak és a támadás leghatékonyabb formáinak meghatározása. A felderítés az elszenvedő hálózat részéről általában nem észlelhető formában valósul meg, így a hálózat üzemeltetője és felhasználója számára a felderítés ténye többnyire nem ismert."[5]

A cikk egy lehetséges penetrációsteszt-módszertan felderítés és analízis második szintjének bemutatásával foglalkozik, ahol a technikai megvalósításokon van a hangsúly, és azok elméleti megértésén.

Felderítés- és analízisszintek:

- információgyűjtés:
  - a közzétett adatok elemzése:
    - információgyűjtés keresőmotorok segítségével,
    - információgyűjtés webszolgáltatásokon keresztül,
    - weboldalinformáció-gyűjtés,
    - e-mail-információgyűjtés.
  - alapvető hálózati információk lekérdezése:
    - WhoIs,[6]
    - DNS-információk kibontása;
- hálózat-feltérképezés:
  - célpontfelfedés,

[4] Paráda István: Basic of cybersecurity penetration test. *Hadmérnök,* 13. (2018), 3. 435–442.

[5] Haig Zsolt: *Információs műveletek a kibertérben.* Budapest, Dialóg Campus, 2018. 247.

[6] A WhoIs egy domainnév-adatbázis, ami olyan információkat tartalmaz, mint például a regisztráló, az adminisztrátor és a műszaki ügyekben illetékes személyek regisztrációkor megadott kapcsolatfelvételi adatai; a szponzoráló regisztrátor; a létrehozás, frissítés és a lejárat dátuma; valamint a névszerverek és domainállapotok.

  - portszkennelés,
  - OS-ujjlenyomat,
  - hálózatiforgalom-elkapás, lehallgatás;
- sérülékenység elemzése és értékelése.

Ahhoz, hogy a cikkben kontextusba tudjuk helyezni a bemutatni kívánt információgyűjtési metódusokat, gyakorlatokat és felkínált végrehajtási lehetőségeket, definiálni kell magát a tevékenység hatókörét, ami nem más, mint a kibertér. „Kibertér: felhasználók, eszközök, szoftverek, folyamatok, tárolt vagy átvitel alatt lévő információk, szolgáltatások és rendszerek gyűjtőfogalma, amelyek közvetlenül vagy közvetett módon számítógép-hálózathoz vannak kapcsolva."[7] Ebből kiderül, hogy mivel sokféle elemet érintő, dinamikusan változó tartományról beszélünk, a benne végrehajtható információgyűjtés is több síkon értelmezhető. Például a nyilvános adatok gyűjtése lehet emberközpontú, emberi kapcsolatok nézőpontjából, vagy szervezeti, illetve intézményi szempontok alapján, valamint ezek technikai oldalú megközelítésén keresztül vizsgált.

2. Hálózatfelderítési technikák

A hálózati szkennelés célja a hálózaton fellelhető esetleges hibák felderítése és azonosítása. A továbbiakban az alkalmazható eljárásokat és módszereket mutatjuk be.

2.1. A portszkennelés

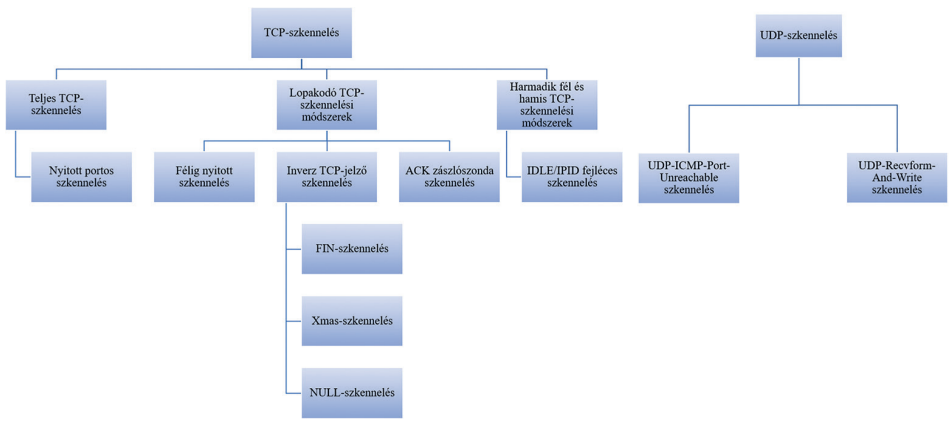
Minél több információ áll rendelkezésre a célszervezetről, annál nagyobb az esélye a hálózat biztonsági profiljának megismerésére és ennek következtében a jogosulatlan hozzáférésre.

Az alábbiakban bemutatunk néhány célt a hálózat szkennelésére:

- Hálózati számítógépek, IP-címek és az élő portok felfedezése. Nyitott portok használatával a támadó meghatározza a rendszerbe való belépés legjobb módját.
- Cél operációs rendszerek és rendszerarchitektúra felfedezése. Ezt ujjlenyomatnak is nevezik. A támadó az operációs rendszer sebezhetőségén alapuló támadási stratégiát fogalmazhat meg.
- Célrendszeren futó/hallgató szolgáltatások felfedezése. Ez a támadónak jelzi a (szolgáltatáson alapuló) biztonsági rések kihasználását a célrendszerhez való hozzáféréshez.
- Egy adott szolgáltatás alkalmazásának vagy verziójának meghatározása. Ezzel meghatározható, hogy a futtatott szolgáltatás tartalmaz-e ismert hibákat, amelyeket a támadó ki tud használni.
- A hálózati rendszerek sebezhetőségének meghatározása. Ez segít a támadónak a célrendszer vagy a hálózat veszélyeztetésére különféle kihasználások révén.

[7] Kovács László: *A kibertér védelme*. Budapest, Dialóg Campus, 2018. 18.

A szkennelés olyan folyamat, amely aktív, működő és reagáló rendszerekről gyűjt információt a hálózaton. A hálózatok szkennelésének első lépése az élő rendszerek ellenőrzése. Ez a szakasz bemutatja, hogyan lehet ellenőrizni az élő rendszereket az ICMP-szkennelés[8] segítségével, hogyan lehet pingelni egy rendszert, továbbá bemutat különféle ping-sweep[9] eszközöket. Amikor a támadók észlelik a célhálózat rendszereit, nyitott portokat keresnek a felfedezett élő rendszerekben. A hálózati szkennelési folyamat következő lépése az aktív rendszerek nyitott portjainak ellenőrzése. Időnként a felhasználók nyitva tartanak számukra szükségtelen portokat az alkalmazott rendszereiken. A támadó kihasználja az ilyen nyílt portok előnyeit a támadások indításához.[10]



1. ábra: *A szekennelés csoportosítása*

Forrás: a szerzők szerkesztése

A szkennelési technikákat az 1. ábra szerint továbboszthatjuk kettő kategóriába a hálózat szállítási rétegén a kommunikációhoz alkalmazott protokollok típusa alapján az alábbiak szerint:

- TCP-hálózati[11] szolgáltatások:
  - nyitott TCP-szkennelési módszerek:
    - TCP Connect I teljes nyitott vizsgálat;
  - lopakodó TCP-szkennelési módszerek:
    - félig nyitott szkennelés,
    - inverz TCP-jelző szkennelés:
      - Xmas-szkennelés,

[8] Az ICMP a hálózati rétegben futó néhány hasznos üzenetküldő parancs összessége. Ha egy ICMP-üzenetet nem lehet továbbítani, akkor az üzenet elvész, másolat nem keletkezik. Ennek célja, hogy elkerüljék, hogy az ICMP-üzenetek a hálózatot elárasszák. Leírását az RFC 792 tartalmazza.

[9] A ping-sweep egy egyszerű módszer a hálózaton élő IP-címek listájának megszerzésére, amely csak a bekapcsolt és aktív eszközök eredményeit adja vissza.

[10] Sean-Philip Oriyano: *Certified Etichal Hacker.* Sybex, 2016.

[11] Átvitelvezérlő protokoll: az egyik szállítási protokoll, amely egy megbízható összeköttetés-alapú protokoll, feladata a hibamentes átvitel biztosítása bármely két gép között az interneten.

        - FIN-szkennelés,
        - NULL-szkennelés,
      - ACK zászlószonda szkennelése;
    - harmadik fél és hamis TCP-szkennelési módszerek:
      - IDLE/IPID fejléc[12] szkennelése.
- UDP-hálózati[13] szolgáltatások szkennelése:
  - UDP-szkennelés.

2.2. A TCP-szkennelés

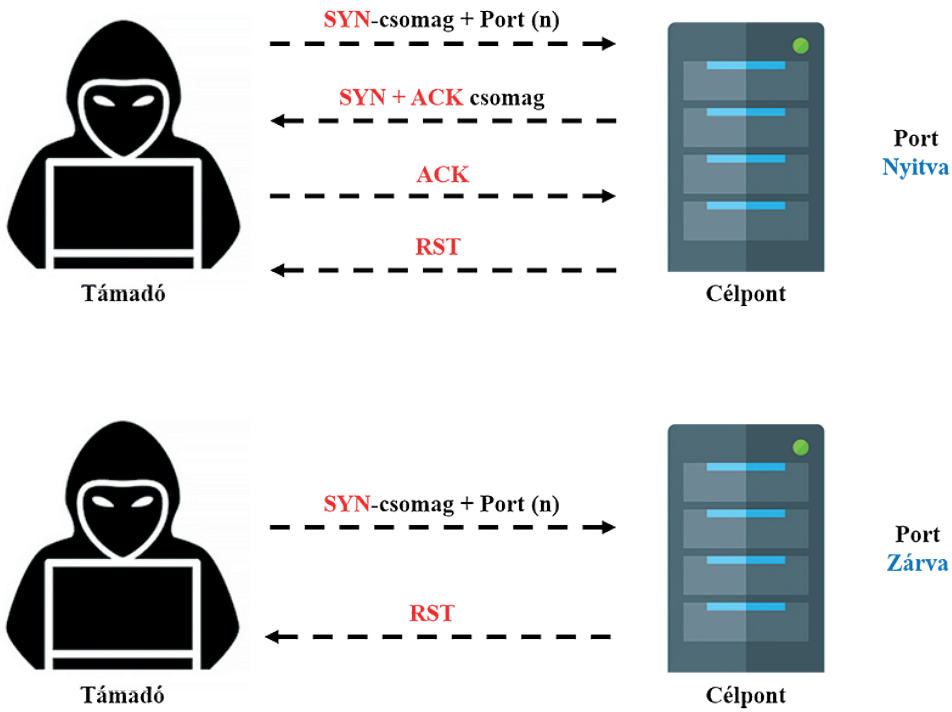
A portszkennelés végrehajtásához szükséges lépések egyik alapvető formája a TCP-alapú portvizsgálatok, amelyeket a következőkben mutatunk be.

2.2.1. Teljes vagy nyitott portos vizsgálat

A TCP-csatlakozás vagy a teljes nyitott vizsgálat csak egy újabb módszer annak megállapítására, hogy háromutas kézfogás megvalósul-e a célrendszer portjain azért, hogy meghatározzák, hogy melyik port nyitott és melyik zárt. Ha a port hallgat, akkor a csatlakozási meghívás sikeres kapcsolatot hoz létre az adott port számítógépével; egyébként hibaüzenetet küld, amely kijelenti, hogy a port nem érhető el. A teljesen nyitott szkennelés előnye, hogy a vizsgálat során azonnali pozitív visszajelzés kapható, hogy egy port nyitva vagy zárva van-e. Ennek a szkennelésnek azonban hátránya, hogy visszavezet a háromirányú kézfogás használatához. Köztudott, hogy a háromutas kézfogás célja annak megerősítése, hogy mindkét fél kommunikálni fog, viszont ha mindkét fél megerősíti jelenlétét és részvételét a kapcsolatban, akkor mindenki tudja, hogy mindkét fél ott van, és kik ők. Valamint a célrendszer naplófájljai felfedik az összekapcsolódást. A 2. ábra bemutatja, hogyan működik ez a folyamat egy nyitott és zárt port észlelése során.

A TCP háromutas kézfogásban a támadó SYN-csomagot[14] küld, amelyet a címzett egy SYN + ACK[15] csomaggal nyugtáz. A támadó nyugtázza a SYN + ACK csomagot egy ACK-csomaggal a kapcsolat befejezéséhez. A kézfogás befejezése után a szkennelő RST-csomagot[16] küld a kapcsolat megszakításához.

[12] IP Internet Identification: ez a mező a fragmensek összeszerelése során használt IP-csomagok azonosítását szolgálja.
[13] Összeköttetésmentes protokoll: az UDP információját egy IP-csomagba helyezi, ellenőrző összeget számol hozzá és feladja, ezáltal a kézbesítést nem garantálja, de a hibás kézbesítést észlelhetővé teszi.
[14] Nyitó csatlakozás.
[15] Nyugtázó csomag.
[16] Záró csatlakozás.



2. ábra: *Teljes vagy nyitott portos vizsgálat*

Forrás: a szerzők szerkesztése

Nyitott port esetén a válasz olyan, mint egy normál háromutas kézfogás esetén; azonban egy zárt port esetén csak RST-csomagot kap a támadó. A válaszmintázat ismeretével meghatározható, hogy a port véglegesen nyitva van-e, vagy sem. Az Nmap[17] teljes nyitott vizsgálatának futtatása parancssorba:

nmap –sT – v <cél IP-cím>

2.2.2. Lopakodó TCP-szkennelési módszerek

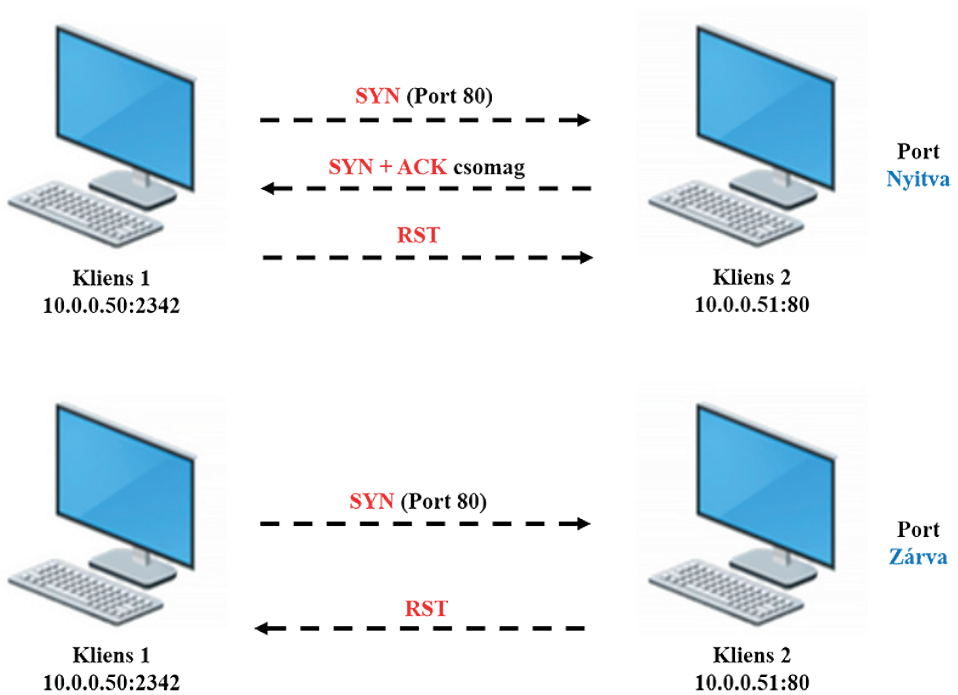
A továbbiakban a lopakodó TCP-szkennelési módszereket mutatjuk be különböző lehetséges eljárási módokon keresztül.

[17] Az Nmap („Network Mapper") egy hálózatfelderítésre és biztonsági ellenőrzésre használható, nyílt forráskódú megoldás.

2.2.2.1. Stealth-szkennelés (félig nyitott vizsgálat)

Az ilyen típusú szkennelésnél a folyamat nagyban hasonlít a teljes nyílt szkennelésre, néhány különbséggel, ami lopakodóbbá teszi. A fő különbség az ilyen típusú szkennelésnél az előző vizsgálati típushoz képest az utolsó lépésben található. A Stealth-vizsgálat magában foglalja a kliens és a szerver közötti TCP-kapcsolat hirtelen alaphelyzetbe állítását, mielőtt a háromirányú kézfogás jelei befejeződnének, tehát a kapcsolat félig nyitott lesz. A lopakodó vizsgálat egyetlen keretet küld a TCP-portra bármilyen TCP-kézfogás vagy további csomagátvitel nélkül. Az ilyen típusú szkennelés egyetlen keretet küld, egyetlen válaszra várva. A lopakodó vizsgálatot SYN-vizsgálatnak is nevezik, mivel csak a SYN-csomagot küldi el. Ez megakadályozza, hogy a szolgáltatás jelezze a bejövő kapcsolatot. A lopakodásos szkennelés folyamata a következő:

- az ügyfél egyetlen SYN-csomagot küld a kiszolgálóra a megfelelő porton;
- ha a port nyitva van, utána a szerver egy SYN-/ACK-csomagra válaszol;
- ha a szerver egy RST-csomaggal válaszol, akkor a távoli port „zárt" állapotban van;
- az ügyfél elküldi az RST-csomagot, hogy bezárja a kezdeményezést, mielőtt a kapcsolat létrejöhet.



3. ábra: *A félig nyitott vagy lopakodó szkennelés*

Forrás: a szerzők szerkesztése

A félig nyitott vagy lopakodó szkennelés előnye, hogy kevésbé valószínű, hogy észlelési mechanizmusokat indít. A hátránya, hogy egy kicsit kevésbé megbízható, mint a teljes nyitott vizsgálat, mivel a folyamat során nem érkezik megerősítés. Ennek az a hátránya, hogy bizonyos esetekben kissé lassú. A támadók lopakodó szkennelési technikákat alkalmaznak a tűzfalszabályok, a naplózási mechanizmusok megkerülésére és a hálózati forgalom elrejtésére.

A félig nyitott szkennelés végrehajtásának futtatása parancssorban:

nmap –sS –v <cél IP-cím>

2.2.2.2. Fordított TCP-jelző szkennelés

Ebben az esetben a támadók TCP-próbacsomagokat küldenek beállított TCP-jelzőkkel, -zászlókkal (FIN,[18] URG,[19] PSH[20]) vagy zászlók nélkül. Amikor a portok nyitva vannak, a támadó nem kap semmilyen választ a számítógéptől, de abban az esetben, ha a portok zárva vannak, a támadó gép egy RST-csomagot kap a célgépektől.

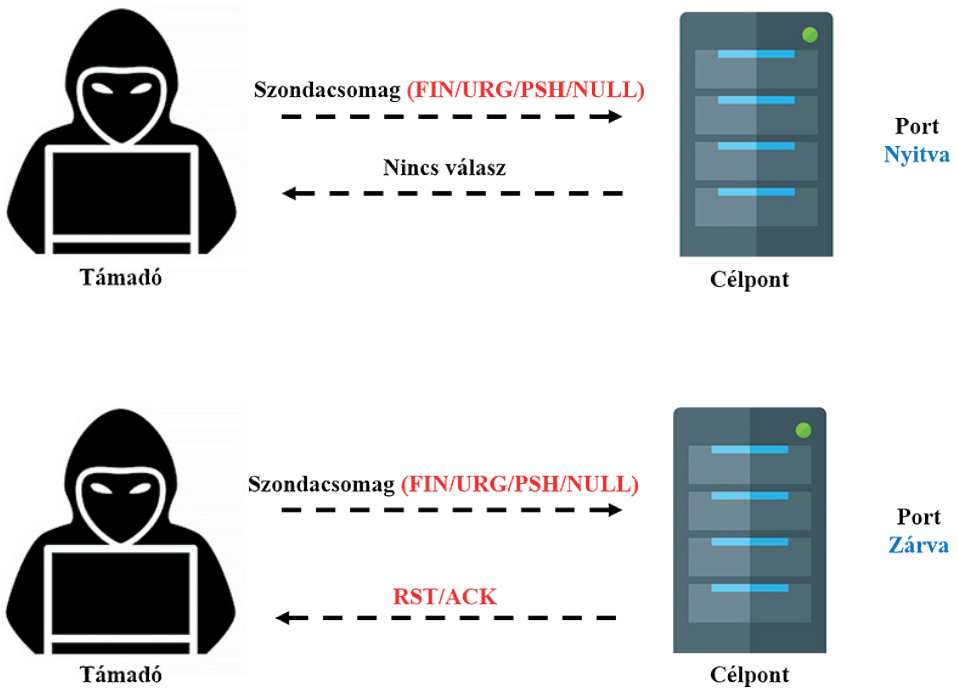
A biztonsági mechanizmusok, mint például a tűzfalak és az IDS,[21] felismerik a megcélzott állomások érzékeny portjaihoz küldött SYN-csomagokat. A félig nyitott SYN-zászlóval végzett vizsgálatok naplózására olyan programok érhetők el, mint például a Synlogger és a Courtney. Időnként a TCP-zászlókkal engedélyezett próbacsomagok észlelés nélkül átjuthatnak a szűrőkön, a telepített biztonsági mechanizmusoktól függően.

A fordított technika azt jelenti, hogy a cél felmérése félig nyitott SYN-zászló használatával történik, mivel a zárt portok csak a választ küldik vissza. Az RFC 793[22] szerint a kapcsolathoz küldött RST-/ACK-csomag alaphelyzetbe áll, amikor a számítógép bezár egy portot. A támadók kihasználják ezt a funkciót, hogy TCP-próbacsomagokat küldjenek a célállomás minden egyes portjára, különféle TCP-jelzőkkel beállítva.

A szondacsomaghoz használt általános zászlókonfigurációk a következők:

- FIN-szonda a beállított FIN TCP-jelzővel;
- Xmas-szonda a FIN, URG és PSH TCP-zászlókkal;
- NULL-szonda, amely esetében nincs beállítva TCP-jelző;
- SYN-/ACK-szonda.

[18] Kapcsolat bontása.
[19] Sürgősségi jelző.
[20] Áttöltési funkció, amely alkalmazása során a fogadó oldalon a TCP-réteg rögtön átadja a beérkező csomagokat az alkalmazásrétegnek.
[21] Illetéktelen hálózati behatolást jelző rendszer.
[22] Az átvitelvezérlő protokoll.



4. ábra: *Inverz TCP-jelző szkennelés*

Forrás: a szerzők szerkesztése

2.2.3. A FIN-szkennelés

A FIN-vizsgálat akkor fordul elő, amikor a támadó TCP-szegmenseket küld az áldozatoknak a beállított FIN-jelzővel. Ez a módszer a kapcsolat bezárását kéri, mivel további információk nem kerülnek elküldésre. Ennek a műveletnek az eredménye, hogy a célzott rendszer nem ad választ, ha a port bezáródik, de ha a port nyitva van, akkor egy RST-t küld vissza, hasonlóan az Xmas tree vizsgálatához.

A FIN-vizsgálat Nmap-ben a következő paranccsal végezhető el:

Nmap –sF <target IP address>

2.2.4. Az Xmas-szkennelés

Az ilyen típusú szkennelés során több jelző van beállítva, ami azt jelenti, hogy egy csomagot egy ügyfélnek küldünk a SYN, PSH, URG és FIN segítségével, ezeket egyszerre beállítva ugyanazon a csomagon. Ha a célpont megnyitotta a portot, akkor a távoli rendszer nem fog válaszolni. Ha a cél lezárta a portot, akkor távoli rendszerváltást fog kapni egy RST-vel. Az összes zászló beállítása után néhány rendszer lefagy; így a leggyakrabban beállított zászlók az URG-, a PSH- és a FIN-minta. A támadók a TCP Xmas-szkenneléssel ellenőrzik, hogy az RST-csomag segítségével a portok zárva vannak-e a célgépen. Ha a célrendszer elfogadja a csomagot, és nem válaszol, akkor az azt jelenti, hogy a port nyitva van. Ha a célrendszer RST-jelzőt küld, akkor az valószínűsíti a port zártságát.

Előnye, hogy kerülhető a TCP háromutas kézfogás. Hátránya, hogy csak UNIX-platformon működik.

Az Xmas-szkennelés Nmap-ben a következő paranccsal végezhető el:

nmap –sX –v <target IP address>

2.2.5. A NULL-szkennelés

A NULL-vizsgálat egy másik érdekes vizsgálat, amelyet végre lehet hajtani, és amely bizonyos módon ellentétes a Xmas vizsgálatával. A NULL-vizsgálat elvégzéséhez egy csomagot küldünk, amelyen egyáltalán nem kerülnek zászlók beállításra, és az eredmények megmutatják, ha a port nyitva vagy zárva van-e. A nyitott port nem válaszol, a zárt pedig egy RST-t ad vissza.

A NULL-szkennelés Nmap-ben a következő paranccsal végezhető el:

nmap –sN <target IP address>

2.2.6. Az ACK zászlószonda szkennelése

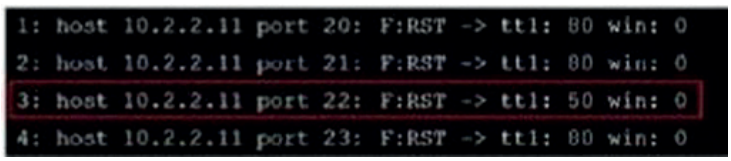
Ennél a módszernél a támadók TCP-próbacsomagokat küldenek egy távoli eszközre beállított ACK-jelzéssel, majd elemezik a kapott RST-csomagok fejlécinformációit (TTL-[23] és Window-mező) annak megállapítására, hogy a port nyitva vagy zárva van-e.

2.2.7. A TTL-alapú ACK zászlószonda szkennelése

Ebben a szkennelési technikában először el kell küldeni az ACK-szondacsomagokat a különböző TCP-portokhoz, majd elemezni kell a kapott RST-csomagok TTL-mezőértékét. Ha egy adott porton az RST-csomag TTL-értéke alacsonyabb, mint

[23] Time to Live: célja a számítógépen vagy a hálózaton keresztül áramló adatcsomagok élethosszának a meghatározása.

a 64 bites határérték, akkor a port nyitva van, amely látható az 5. ábra pirossal bekeretezett részén.

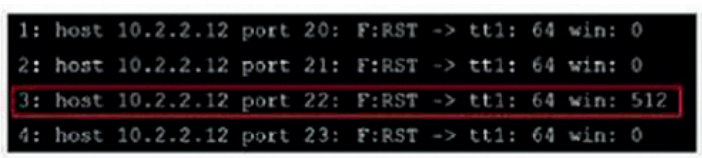


5. ábra: *TTL-alapú ACK zászlószonda szkennelése*

Forrás: a szerzők szerkesztése

2.2.8. A WINDOW-alapú ACK zászlószonda szkennelése

Ebben a szkennelési technikában először el kell küldeni az ACK-szondacsomagokat a különböző TCP-portokhoz, majd elemezni kell a kapott RST-csomagok Window-mező értékét. A támadó akkor használhatja ezt a szkennelési technikát, ha az összes port ugyanazt a TTL-értéket adja vissza. Ha az RST-csomag Window-értéke egy adott porton nem nulla, akkor a port nyitva van, amely látható a 6. ábra pirossal bekeretezett részén.



6. ábra: *Window-alapú ACK zászlószonda szkennelése*

Forrás: a szerzők szerkesztése

2.2.9. A célhálózatok szűrőrendszereinek ellenőrzése

Az ilyen típusú szkennelés során a támadók az ACK-jelzőkészlettel ellátott csomagot juttatják el a célponthoz. Az ACK-kérés, amely az áldozatnak kerül elküldésre, és amely nem ad vissza választ, azt jelzi, hogy tűzfal van jelen, és szűrést hajt végre, míg az áldozattól érkezett RST-jelzés azt jelzi, hogy nem történik szűrés.

2.2.10. Harmadik fél és hamis TCP-szkennelési módszerek

2.2.10.1. IDLE-/IPID-fejléces vizsgálat

Az IDLE-/IPID-fejléces vizsgálat egy TCP-portkeresési módszer, amelynek segítségével hamis forráscím küldhető egy számítógépre, annak kiderítésére, hogy azon milyen szolgáltatások futnak, milyen biztonsági funkciók állnak rendelkezésre.

A legtöbb hálózati szerver a TCP-portokon hallgat, például a 80-as porton lévő webszerverek és a 25-ös porton lévő levelezőkiszolgálók. A port akkor tekinthető „nyitottnak", ha egy alkalmazás figyeli a portot. Mint az feljebb már leírtuk, az egyik módszer annak megállapítására, hogy egy port nyitva van-e, egy „SYN"-csomag küldése a portra. A célgép egy „SYN ACK"-csomagot küld vissza, ha a port nyitva van, és egy „RST"-csomagot, ha a port zárva van. Az a gép, amely kéretlen SYN-/ACK-csomagot fogad, RST-vel válaszol. A kéretlen RST-t figyelmen kívül hagyjuk.

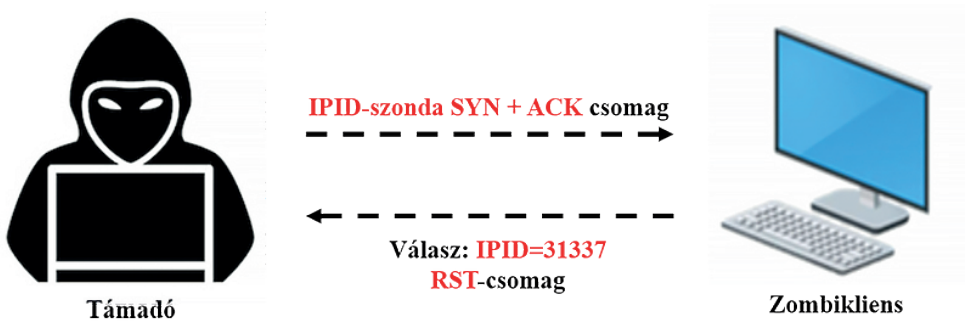
Az interneten minden IP-csomagnak van egy „fragment azonosító" száma (IPID). Az operációs rendszer növeli az egyes elküldött csomagok IPID-jét, így az IPID lekérdezése megadja a támadónak az utolsó szonda óta elküldött csomagok számát.

A támadó ezt a szkennelést hamisítás révén hajtja végre egy másik számítógép megszemélyesítésével. A támadó nem küld csomagot saját IP-címéről, ehelyett egy másik számítógépet használ a távoli számítógép átvizsgálásához, és az esetleges nyitott portok azonosításához, amelyet gyakran „zombi"-nak is neveznek. Ebben a támadásban a támadó ellopja a zombiszámítógép azonosítóját, és ha a távoli számítógép ellenőrzi a szkennelő fél IP-jét, akkor megjelenik a zombigép IP-je. A feltérképezése egyes lépései a következők:

1. lépés

Az IDLE-szkennelés végrehajtásának első lépése a megfelelő zombiállomás megtalálása. Az a zombi, amely globális alapon növekményesen osztja ki az IPID-csomagokat, egy megfelelő vagy tétlen zombi az IDLE-szkennelés elvégzéséhez. Minél rövidebb a kérés/válasz időszaka a támadó–zombi és a zombi–célpont között, annál gyorsabb a vizsgálat.

Az első lépésben elküldésre kerül a SYN + ACK csomag a zombi gépnek, hogy ellenőrizze annak IPID-számát. Itt a SYN + ACK csomagok küldésének oka az IPID-szám kipróbálására, de a TCP-kapcsolat létrehozásának megakadályozására (háromutas kézfogás).



7. ábra: *A megfelelő zombi megtalálása*

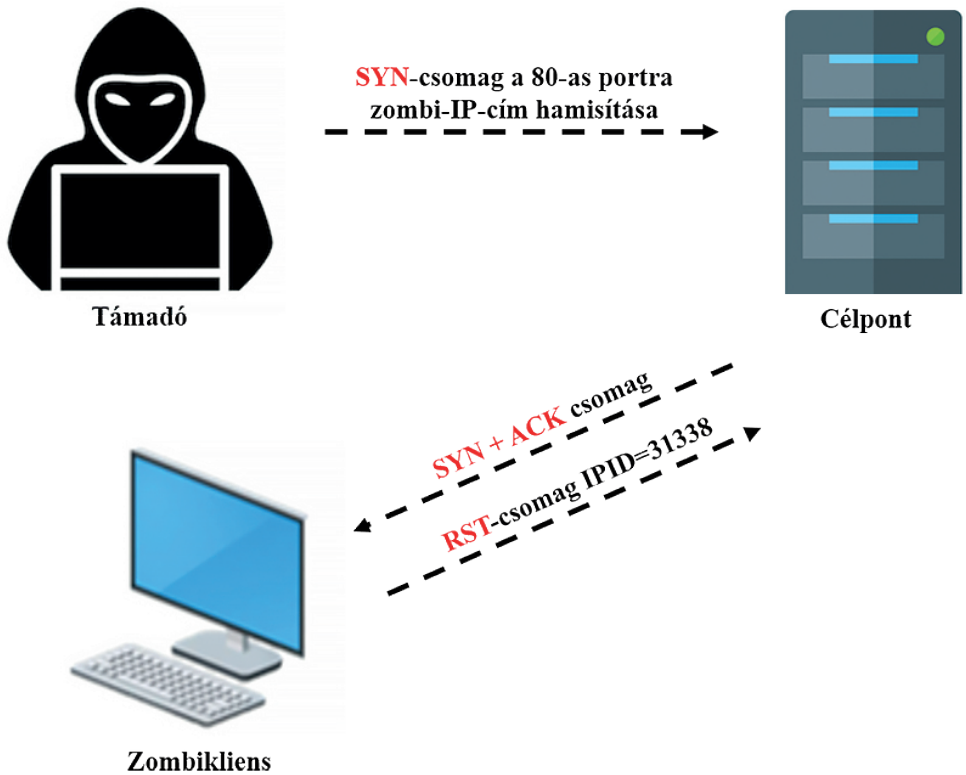
Forrás: a szerzők szerkesztése

Mivel a zombi nem vár SYN + ACK csomagot, RST-csomag visszaadásával megtagadja a kapcsolatot. Elemzésre kerül a zombi gép által küldött RST-csomag az IPID kibontásához. A fenti ábrán látható diagramon tegyük fel, hogy a zombi IPID = 31337 értékkel reagál. Tegyük fel, hogy ez az IPID X.

2. lépés

A támadó SYN-csomagot küld a célgépre a 80-as porton, hamisítja a zombigép IP-címét.

Ha a port nyitva van, a cél elküldi a SYN + ACK csomagot a zombi számára (mivel az IP-cím hamis volt), hogy folytassa a háromutas kézfogást. Mivel a zombi nem várt SYN + ACK csomagot a célgéptől, RST-csomaggal válaszol.

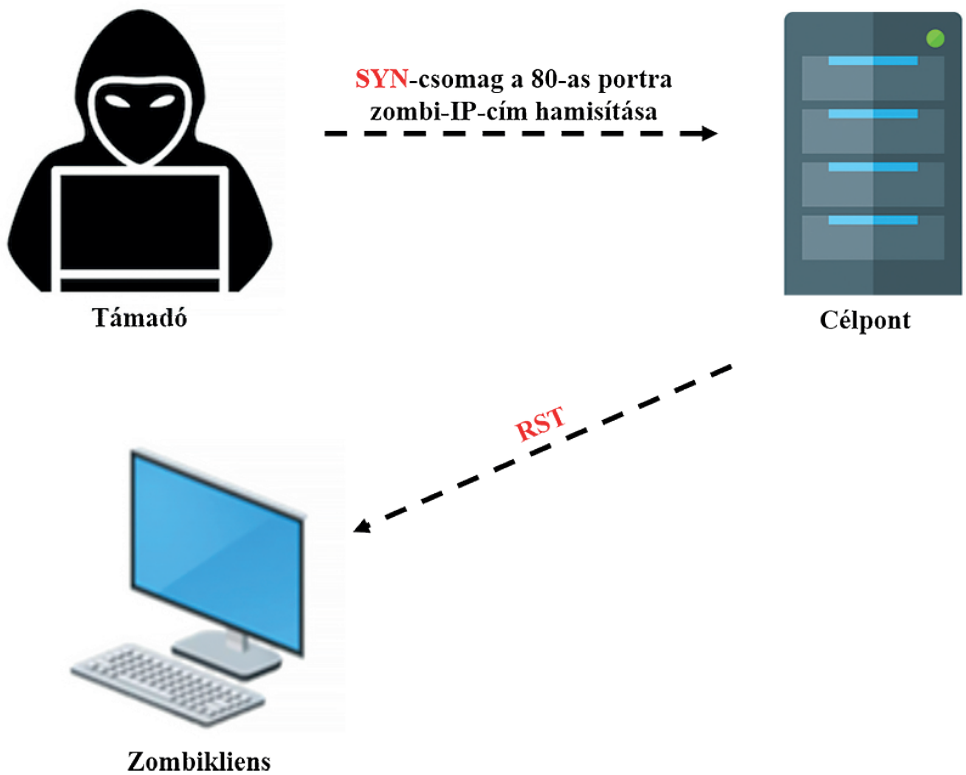


8. ábra: *SYN-csomaggal végrehajtott zombi-IP-cím-hamisítás*

Forrás: a szerzők szerkesztése

Mivel minden IP-csomagnak van egy „fragment" azonosító száma, amely minden egyes csomagátvitelnél eggyel növekszik, ezúttal a zombi a következő elérhető IPID-jét fogja használni, azaz 31338 (X + 1).

Tegyük fel, hogy a cél portja zárva van. Ezt követően, amikor megkapja a SYN-csomagot a támadótól a célpont egy RST-vel válaszol, és a zombi tétlen marad további lépések nélkül.



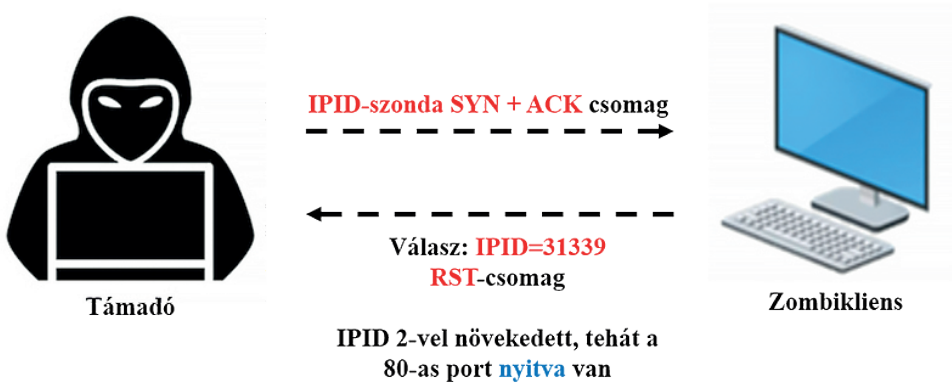
9. ábra: *A célpont válasza RST-vel*

Forrás: a szerzők szerkesztése

3. lépés

Egy SYN + ACK csomagot küldenek a zombinak, amely egy IPID-et tartalmazó RID-csomaggal válaszol. Feltételezve, hogy a cél portja nyitva volt, és a zombi már küldött egy RST-csomagot a célhoz, abban az esetben az IPID száma 1-gyel növekedett. Ezúttal a zombi RST-csomaggal válaszol a támadóra a következő IPID-jének, azaz 31339 (X + 2) használatával. Következésképpen az IPID megnövekedett kettővel, ami azt jelenti, hogy a célgép portja nyitva volt. Így egy tétlen szkenneléssel a támadó megtudhatja a célgépek nyitott portjait és szolgáltatásait azáltal, hogy az IP-címét egy zombi IP-címével hamisítja meg.[24]

[24] Nmap Network Scanning. TCP Idle Scan. Nmap.



10. ábra: *Nyitott port azonosítása az IPID-változóból*

Forrás: a szerzők szerkesztése

2.2.11. Az UDP-szkennelés

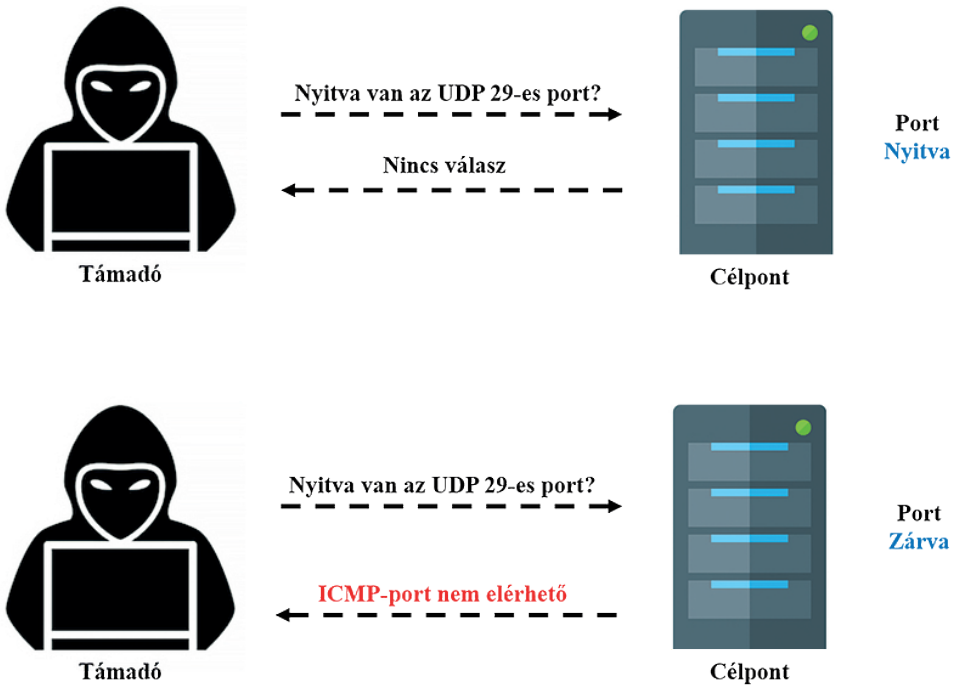
2.2.11.1. Az UDP-ICMP-Port-Unreachable szkennelés

Az UDP egyik fő jellemzője, hogy más protokollokkal ellentétben, nem igazolja vissza a fogadást, csak a lezárt port küld vissza egy üzenetet, hogy a port nem elérhető. A szkennelésnek ez a módja nagyon hosszadalmas, és a pontossága nagymértékben a szkennelt számítógép kihasználtságától, illetve rendszererőforrásaitól függ. Ráadásul csak Linux alatt működik, és azok a felhasználók, akik rootként vannak bejelentkezve, megkapják az ICMP-Port-Unreach üzeneteket. Ezeket a szkenneléseket csak a rendszergazda tudja elvégezni, hiszen system admin-jogok kellenek hozzá.

2.2.11.2. Az UDP-Recvfrom-And-Write szkennelés

Ellentétben az UDP-ICMP-Port-Unreach szkenneléssel, amelynél csak azok a felhasználók kapnak pozitív visszajelzést, akik rootként vannak bejelentkezve, a UDP-Recvfrom-And-Write szkennelés egy normál módon bejelentkezett felhasználónak is lehetővé teszi, hogy „hasznos" jelzéseket kapjon. A háttérben ismét egy hiba áll: ha megpróbálunk egy portra írni, amelyik az UDP-ICMP-Port-Unreach szkennelésre ICMP-Port-Unreach választ adott (amiről normál felhasználóként nem értesülünk), akkor többnyire ezt az üzenetet kapjuk: „Error 13 - Try Again", a normál „Error 111 - Connection refused" üzenet helyett. Ennél az eljárásnál tehát minden portot kétszer szkennelnek, egyszer, hogy a számítógép adjon egy választ, amit sajnos, nem látunk, és másodszor, hogy mégis kapjunk informatív visszajelzést. Ebben az esetben a 13-as

hiba azt jelenti, hogy a port le van zárva. Természetesen ez az eljárás is nagyon idő-igényes, és gyakran megbízhatatlan.[25]



11. ábra: *UDP-Recvfrom-And-Write szkennelés*

Forrás: a szerzők szerkesztése

2.2.11.3. Az UDP előnyei

Az UDP kevésbé ismeri fel a nyitott portot, mivel nem tartalmazza a TCP-kézfogást. Ha azonban az ICMP válaszol az egyes nem elérhető portokra, akkor az összes keret száma meghaladhatja a TCP-lekérdezésnél megadott keretek számát. Microsoft-alapú operációs rendszerek általában nem valósítanak meg ICMP-sebességkorlátozást, tehát ez a vizsgálat nagyon hatékonyan működik Windows-alapú eszközökön.

[25] Thomas Vosseberg: *Hacker kézikönyv.* Budapest, Computer Panoráma, 2002. 155.

2.2.11.4. Az UDP hátrányai

Az UDP-vizsgálat csak portinformációt szolgáltat. Ha szükségesek az információ további részletei is, akkor a vizsgálatot ki kell egészíteni egy versiondetektálással (-sV) vagy az operációs rendszer ujjlenyomat-bekapcsolásával (-0).

A legtöbb hálózat hatalmas mennyiségű TCP-forgalommal rendelkezik, és ennek eredményeként az UDP-szkennelés hatékonysága elveszik. Az UDP-vizsgálat megkeresi a nyitott portokat, és értékes információkat nyújt a biztonsági vezető számára a spyware-alkalmazások, trójai programok és más rosszindulatú szoftverek által okozott sikeres támadások azonosításához a nyílt UDP-portokon.

3. Szkennelő eszközök

A szkennelő eszközök az élő hosztokat, a nyitott portokat, a célhálózaton futó szolgáltatásokat, a helyinformációkat, a NetBIOS-információkat[26] és az összes TCP/IP-t, valamint UDP-nyitott portot érintő információkat keresik és azonosítják. Az ezekből a toolokból nyert információ segít a célszervezet profiljának létrehozásában és a hálózatra csatlakoztatott eszközök nyitott portjainak kutatásában.

3.1. Az Nmap

Az Nmap egy nagyon átfogó, funkciógazdag és széles körben használt portszkenner az IT-közösség egészének számára. Rugalmasságának köszönhetően ez egy kötelező eszköz a penetrációs tesztelők számára. Amellett, hogy portszkennerként használják, az Nmap számos más funkcióval is rendelkezik, az alábbiak szerint:

- Hosztfelfedezés: Az Nmap felhasználható élő hosztok keresésére a célrendszereken. Alapértelmezés szerint az Nmap ICMP-visszhangkérést, TCP SYN-csomagot küld a 443 porthoz, TCP ACK-csomagot a 80 porthoz és egy ICMP időbélyegző kérést küld a hoszt felfedezéséhez.
- Szolgáltatás/verzió észlelése: Miután az Nmap felfedezte a portokat, tovább ellenőrizheti a szervizprotokollt, az alkalmazás nevét és a célgépen használt verziószámot.
- Operációs rendszer észlelése: Az Nmap csomagokat küld a távoli hosztnak, és megvizsgálja a válaszokat. Ezután összehasonlítja ezeket a válaszokat az operációs rendszer ujjlenyomat-adatbázisával és kinyomtatja a részleteket, ha van egyezés. Ha nem tudja meghatározni az operációs rendszert, az Nmap URL-t biztosít, amelyen az ujjlenyomatot elküldheti az operációs rendszer ujjlenyomat-adatbázisának frissítéséhez.
- Hálózati nyomvonal: annak a portnak és a protokollnak a meghatározására szolgál, amely valószínűleg eléri a célrendszert. Az Nmap nyomvonala a Time

[26] A NetBIOS a Network Basic Input/Output System kifejezésből készült betűszó. A NetBIOS API lehetővé teszi az egyes számítógépek számára a helyi hálózaton keresztül történő kommunikációt.

to Live (TTL) magas értékével kezdődik, és csökkenti addig, amíg a TTL-érték el nem éri a nullát.
- Nmap Scripting Engine: Ezzel a funkcióval az Nmap kiterjeszthető. Ha olyan csekket szeretne hozzáadni, amely nem szerepel az alapértelmezett Nmap-ben, akkor ezt megteheti az Nmap-parancsfájl-készítő motorral történő írásával. Ellenőrzi a hálózati szolgáltatások sebezhetőségét és a célrendszer erőforrásainak felsorolását.[27]

3.2. A Hping2/Hping3

A Hping2/Hping3 egy parancssori orientált hálózati szkennelési és csomagmegmunkáló eszköz a TCP-/IP-protokollhoz, amely elküldi az ICMP-visszhangkéréseket, és támogatja a TCP-, UDP-, ICMP- és rawIP-protokollokat. A Hping2/Hping3 rendelkezik Traceroute-móddal, amely lehetővé teszi a fájlok rejtett csatornák közötti küldését. Emellett támogatja az alapjárat nélküli szkennelést. Az IP-hamisítás és a hálózati/hosztszkennelés felhasználható a szolgáltatások névtelen szonda-előállítására. A támadó egy alapjárat nélküli hoszt viselkedését tanulmányozza a céllal kapcsolatos információk megszerzése érdekében, mint például a hoszt által kínált szolgáltatások, a szolgáltatásokat támogató portok és a cél operációs rendszere.[28]

4. Összegzés

A kiberműveletekben és informatikában a hálózattérképezési és szkennelési tevékenységek azonosítják a szervezetekhez kapcsolódó technikai információkat, úgymint az üzemelő számítógépek számát, típusát, működésének állapotát, operációs rendszerét stb. A hálózattérképezési technikákat használva számos lehetőség nyílik a célszervezet hálózatának illetéktelen hozzáférésére. Nincs egységes módszer az információgyűjtésre, hiszen az információk számos módon gyűjthetők, beszerezhetők. Viszont a lehető legtöbb információt be kell gyűjteni, így érdemes ezt a fázist szervezett módon végrehajtani. Jelen cikk összefoglalja ennek lehetőségeit, az alkalmazott technika alapján kategorizálja azokat, és konkrét megvalósítási példákat is felsorakoztat a megértés érdekében. A portszkennelési technikák és eljárások indítása és ennek eredményeinek vizsgálata olyan fontos elemek, amelyeknek részletes tanulmányozásával a penetrációs tesztek kezdő fázisa folytatható, és jelentős eredmények érhetők el egy ilyen teszt módszertanának végrehajtásával.

Következtetésként a hálózatfeltérképezési és szkennelési technikák bár sokrétűek és sokfélék, az egyszerűbb információgyűjtéstől haladva a nehezebb, már toolokat alkalmazó gyűjtés felé, minél több forrásból nyerjük ki az információkat, annál nagyobb hatásfokkal indítható el maga a penetrációs teszt módszertanának folyamata. Ezeket

[27] *Top 15 Nmap Commands to Scan Remote Hosts.* SeucurityTrails.
[28] Don Parker: *Hping: How to better understand how hackers attack.* TechTarget.

a megfelelő struktúrába rendezve és gyűjtve egy munkafolyamat fontos lépését lehet megtenni, amely hozzájárul az eredményes és effektív feladat-végrehajtáshoz.

A cikkben taglalt hálózattérképezési és szkennelési tevékenységek jelentős alapot szolgáltathatnak a Magyar Honvédség által alkalmazott informatikai biztonsági teszteléshez. A szakállomány megfelelő kiképzésével, valamint a szkennelés sokrétű használatával olyan portokat és számítógépeket sikerülhet felfedni, amelyek által gyűjtött információk közvetetten hozzájárulhatnak Magyarország kibervédelmi stratégiájának megvalósításához, valamint a Magyar Honvédség kibervédelmi tevékenységeihez, az esetleges támadások elkerüléséhez, megelőzéséhez. Folyamatos alkalmazásával alkalmas az állomány digitális kompetenciájának fejlesztésére, illetve a szakállomány oktatására, továbbképzésére.

Felhasznált irodalom

Haig Zsolt: *Információs műveletek a kibertérben.* Budapest, Dialóg Campus, 2018.

Kovács László: *A kibertér védelme.* Budapest, Dialóg Campus, 2018.

Nmap Network Scanning. TCP Idle Scan. Nmap. Elérhető: https://nmap.org/book/idlescan.html (A letöltés dátuma: 2020. 03. 11.)

Oriyano, Sean-Philip: *Certified Etichal Hacker.* Sybex, 2016.

Paráda István: Basic of cybersecurity penetration test. *Hadmérnök,* 13. (2018), 3. 435–442.

Parker, Don: *Hping: How to better understand how hackers attack.* TechTarget. Elérhető: https://searchsecurity.techtarget.com/feature/Hping-How-to-better-understand-how-hackers-attack (A letöltés dátuma: 2020. 04. 06.)

Top 15 Nmap Commands to Scan Remote Hosts. SeucurityTrails. Elérhető: https://securitytrails.com/blog/top-15-nmap-commands-to-scan-remote-hosts (A letöltés dátuma: 2020. 04. 06.)

Vosseberg, Thomas: *Hacker kézikönyv.* Budapest, Computer Panoráma, 2002. Elérhető: http://fortresscomm.hu/olvasoterem/szamitastechnika/Hacker%20kezikonyv/06.pdf (A letöltés dátuma: 2020. 04. 05.)