SIGURNOST RAČUNALNIH SUSTAVA, 2020/2021

ZADATCI ZA PRIPREMU ZA ZAVRŠNI ISPIT
14.06.2021.

1. **Je li jednostavnije skenirati TCP ili UDP pristupe? Objasnite zašto.**

Ako skeniram otvoreni TCP port, dobit ću nazad potvrdu da je otvoren (primjerice, na SYN ću dobiti nazad SYN+ACK)
Ako skeniram zatvoren TCP port, dobit ću nazad potvrdu da je zatvoren (dobit ću RST)
Ako negdje na putu stoji filter, neću dobiti nikakvu informaciju nazad.

Ako skeniram otvoreni UDP port, neću dobiti nikakvu informaciju nazad.
Ako skeniram zatvoren UDP port, dobit ću nazad "ICMP port unreachable".
Ako negdje na putu stoji filter, neću dobiti nikakvu informaciju nazad.

=> TCP sken je, ako mi promet nije pouzdan na putu do poslužitelja, onda ću dobiti informaciju o otvorenosti porta.

=> S druge strane, za UDP neću dobiti nikakvu informaciju ako je port otvoren, te tada ne znam je li port otvoren ili se promet filtrira putem. Također, UDP je nepouzdan, tj. ne mogu biti siguran da je moj paket došao do odredišta, stoga moram više puta slati paket da budem sigurniji. Dodatno neki OS-ovi ograničavaju brzinu slanja ICMP poruka, što dodatno otežava UDP skeniranje.

2. **Koja je razlika između skeniranja otvorenih pristupa metodama "TCP SYN" i "TCP connect"? Kako biste otkrili da ste žrtva skeniranja tipa "TCP connect".**

Ako skeniram otvoreni TCP port sa TCP SYN metodom, na poslužitelju se neće potpuno otvoriti TCP veza. Točnije, veza će završiti u polu-otvorenom stanju (bez dovršenog three-way handshakea), te takve veze neće završiti u logovima poslužitelja.
S druge strane, svaki TCP connect sken otvara novu TCP vezu s three-way handshake-om, te će se takve veze vidjeti u logovima. Možemo otkriti da smo žrtva TCP connect skena ako vidimo jako puno kratkotrajnih TCP veza na puno portova.

3. **Snimanjem prometa alatom tcpdump, na poslužitelju s adresom 10.0.1.10 primjećujete veliku količinu dolaznih ICMP paketa s raznih adresa. Poslužitelj pritom nije slao nikakve ICMP pakete koji bi prouzrokovali taj promet. O kojem se napadu radi? Koje je vrijednosti napadač trebao unijeti u polja izvorišne (source) i odredišne (destination) adrese u IP zaglavlju početnog ICMP paketa kako bi izveo napad? Sve podmreže u zadatku koriste prefiks duljine 24 bita.**

```
ispis naredbe "tcpdump -ni eth0"
10:09:49.062946 IP  10.0.0.1  > 10.0.1.10: ICMP echo reply  id 0, seq 0,  length 8
10:09:49.062970 IP  10.0.0.21 > 10.0.1.10: ICMP echo reply, id 0, seq 0,  length 8
10:09:49.062972 IP  10.0.0.22 > 10.0.1.10: ICMP echo reply, id 0, seq 0,  length 8
10:09:49.062994 IP  10.0.0.30 > 10.0.1.10: ICMP echo reply, id 0, seq 0,  length 8
...
10:09:49.062999 IP  10.0.0.29 > 10.0.1.10: ICMP echo reply, id 0, seq 0,  length 8
10:09:49.063001 IP  10.0.0.32 > 10.0.1.10: ICMP echo reply, id 0, seq 0,  length 8
```

```
10:09:49.063013 IP 10.0.0.35 > 10.0.1.10: ICMP echo reply, id 0, seq 0, length 8
10:09:49.063018 IP 10.0.0.20 > 10.0.1.10: ICMP echo reply, id 0, seq 0, length 8
...
```

=> Radi se o "smurf" napadu - napadač je poslao ICMP echo zahtjev sa odredišnom adresom 10.0.0.255/24 (broadcast za podmrežu 10.0.0.0/24) kako bi sva računala u podmreži primila paket, te sa (lažiranom) izvorišnom adresom poslužitelja 10.0.1.10 kako bi onda sva računala u podmreži "odgovorila" poslužitelju na primljeni echo zahtjev.

4. Što je IP zavaravanje (IP spoofing) i kako se može zloupotrijebiti?

IP spoofing je slanje IP paketa sa lažiranom izvorišnom adresom u svrhu zavaravanja primatelja. IP spoofing se zloupotrebljava u mnogim DDoS napadima (swarm, UDP amplification, UDP reflection, UDP storm, UDP hijacking). Primjerice, napadač bi lažiranjem svoje IP adrese mogao glumiti UDP server za koji neki korisnik misli da je siguran, te time od korisnika uzet povjerljive podatke (ovo je UDP hijacking).

5. Što je to DDoS? Navedite neki primjer i objasnite kako radi.

DDoS je "Distributed Denial of Service" - raspodijeljeni napad uskraćivanjem usluge. Najčešće se izvor DDoS napada zaražena računala (bot) pod kontrolom napadača (botnet) s kojih napadač onda šalje velike količine zahtjeva na poslužitelj.
Jedan primjer DDoS napada je HTTP slow request/response napad - napadač sa jednog ili više računala otvara HTTP veze prema poslužitelju, te šalje legitimne HTTP zahtjeve, ali jako sporo. Otvaranjem brojnih sporih veza na poslužitelju se troše resursi i zagušuje se poslužitelj, te ako je napad dovoljno jak, poslužitelj više neće moći odgovarati na ostale zahtjeve.

6. Objasnite pojam CA (Certificate Authority) u sklopu arhitekture PKI (Public Key Infrastructure). Koja sumu glavna zaduženja?

Certificate Authority je tijelo koje izdaje, skladišti i potpisuje digitalne certifikate.
Kada komuniciramo s nekim poslužiteljem, moramo biti sigurni u autentičnost poslužitelja. Kao dokaz svoje autentičnosti, poslužitelj nam šalje svoj certifikat kojeg je izdao (potpisao) neki CA. Mi tada provjeravamo taj potpis, i ako možemo potvrditi da je potpis došao od CA kojemu vjerujemo, onda vjerujemo i poslužitelju.
Ne vjerujemo direktno svim CA-ovima naravno, već i autentičnost manjih CA-ova provjeravamo tako da zatražimo njihov certifikat kojeg je potpisao neki (hijerarhijski) viši CA, i tako sve dok ne dođemo do nekog "korijenskog" CA čiji certifikat imamo spremljen u pregledniku.

7. Objasnite ranjivosti protokola ARP. Opišite moguće napade.

ARP protokol nema nikakvu zaštitu povjerljivosti, integriteta ili autentičnosti. Napadač lako može lažirati IP->MAC preslikavanja.
Primjerice, ako PC1 želi komunicirati s nekim poslužiteljem preko router1 u LAN, PC1 mora poslati ARP upit kako bi saznao MAC adresu router1. Ukoliko se napadač nalazi na putu između PC1 i router1, može na primljeni ARP upit odgovoriti sa svojom MAC adresom, čime će sada PC1 u svoj arp cache zapisati da IP adresa od router1 pripada MAC adresi od napadača, te će PC1 sve pakete namijenjene poslužitelju zapravo slati napadaču. Napadač će morati nastaviti slati odgovore na svaki ARP upite od PC1, te ovima praktički izvodi DoS napad.
Napadač može i samo proslijediti pakete dalje do router1 i poslužitelja, te može napraviti identičnu stvar s

ARP upitima od router1 i proslijediti njegove pakete. Ovima je napadač postao MITM, te ima potpunu kontrolu nad komunikacijom između PC1 i router1.

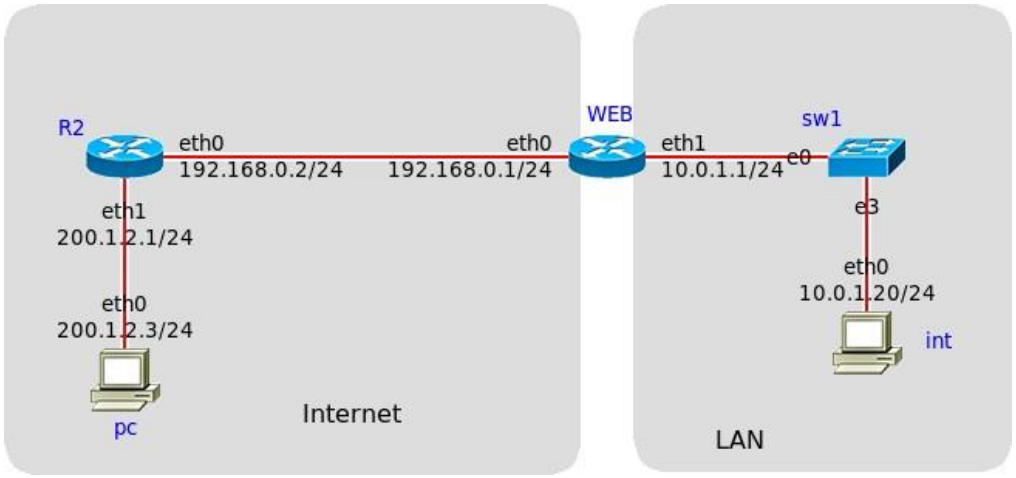
8. Objasnite pojam „demilitarizirane zone“.

DMZ je područje naše mreže u koje ćemo smjestiti poslužitelje i ostale uređaje za koje želimo da budu dostupni na vanjskoj mreži (Internet). Obično se DMZ izvodi ili kao veza da dodatno sučelje firewalla, ili kao mreža između dva firewalla. Kada imamo DMZ i LAN ovako razdvojene, onda lako možemo definirati različita firewall pravila za obje podmreže (primjerice, u LAN nećemo propuštati nikakve pakete ako oni nisu dio već neke established veze, dok u DMZ želimo propustiti nove pakete).

9. Koji osnovni sigurnosni zahtjevi se osiguravaju korištenjem protokola SSH? Objasnite kako.

SSH osigurava:
a) Autentičnost - od klijenta se očekuje da provjeri ispravnost prikazanog sažetka ključa poslužitelja, te se daljnje poruke potpisuju dogovorenim algoritmom - klijent se mora autentificirati lozinkom ili parom ključeva - daljnje poruke potpisuju dogovorenim algoritmom
b) Tajnost i integritet - klijent i poslužitelj na početku dogovaraju algoritme kojim će se osigurati tajnost i integritet, te razmijene potrebne ključeve (ako se ne može dogovoriti, onda se veza prekida)

10. Prikazana je konfiguracijska datoteka vatrozida instaliranog na računalu WEB koje se korisiti kao web poslužitelj. Računalo ima dva mrežna sučelja, eth0 koje je spojeno na Internet i eth1 koje je spojeno na lokalnu mrežu.



popis pristupa:
```
tcp/22 ssh
tcp/23 telnet
tcp/25 smtp
tcp/53 dns
udp/53 dns
tcp/80 http
tcp/443 https
```

```
#interface eth0 192.168.0.1/24 (outside)
#interface eth1 10.0.1.1/16 (inside)
#interface lo 127.0.0.1/8 (loopback)

*filter
:INPUT DROP [0:0]
:OUTPUT DROP [0:0]
:FORWARD DROP [0:0]

-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 !  -i lo -j DROP

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

a. **Napišite pravilo koje će svim računalima iz Interneta omogućiti pristup web poslužitelju protokolom https.**

-A INPUT -p tcp --dport 443 -i eth0 -j ACCEPT

b. **Napišite pravilo koje će omogućiti pristup vatrozidu protokolom SSH isključivo s računala "int" izlokalne mreže (LAN).**

-A INPUT -p tcp --dport 22 -i eth1 -s 10.0.1.20 -j ACCEPT

c. **Je li dozvoljen pristup s firewalla na mail.google.com korištenjem protokola http kroz SSL/TLS? Označite redak u konfiguraciji kojim se to dopušta/zabranjuje. Objasnite.**

Nije. Paketi za pristup s firewalla na mail.google.com idu na OUTPUT chain. OUTPUT chain ima samo jedno definirano pravila te ono nije primjenivo na nove TCP pakete, stoga odlazi na default policy ":OUTPUT DROP [0:0]", tj. ne prolazi.

d. **Napišite pravilo kojim ćete zabraniti ulaz spoofanim dolaznim paketima iz vanjske mreže s izvorišnom adresom jednakom adresama iz lokalne mreže (tj. iz inside mreže).**

-A FORWARD -i eth0 -s 10.0.1.1/24 -j DROP

e. **Kako će vatrozid odgovoriti na dolazne poruke koje su upućene s adrese 200.18.56.28 na vrata tcp/22. Objasnite zbog kojeg je pravila to tako? Prikazana je konfiguracijska datoteka vatrozida instaliranog na računalu WEB koje se korisiti kao web poslužitelj. Računalo ima dva mrežna sučelja,eth0 koje je spojeno na Internet i eth1 koje je spojeno na lokalnu mrežu.**

Dolazni paketi se neće poklopiti ni sa jednim pravilom, te se primjenjuje default policy ":INPUT DROP [0:0]".

11. Koje sigurnosne zahtjeve ostvarujemo ispravnim korištenjem protokola HTTPS na webu?

Ostvarujemo autentičnost, integritet i tajnost.

12. Što je tipičan cilj napada XSS (cross-site scripting)? Objasnite kako nas u tom smislu štiti politika istogizvorišta (same origin policy)

Tipičan cilj napada XSS (cross-site scripting) je umetanje zlonamjernog skriptnog koda na web stranicu koja će se izvršiti u pregledniku korisnika. To omogućuje napadaču da izvede različite vrste zlonamjernih radnji, uključujući krađu korisničkih podataka, preusmjeravanje korisnika na zlonamjerne web stranice, manipuliranje sadržajem stranice ili čak pokretanje napada na druge korisnike.

Politika istog izvorišta (same origin policy) je sigurnosni mehanizam koji se primjenjuje u preglednicima kako bi ograničio interakciju između različitih domena. Osnovni princip politike istog izvorišta je da JavaScript kod iz izvornog izvora ima pristup resursima (kao što su DOM objekti) samo ako je taj kod pokrenut na stranici koja ima isto izvorište kao i izvorni izvor.

To znači da XSS napad može biti uspješan samo ako napadač uspije umetnuti i izvršiti zlonamjerni skriptni kod na istom izvorištu kao i legitimni kod na web stranici. Ako napadač pokuša izvršiti XSS napad iz drugog izvora, politika istog izvorišta će spriječiti pristup resursima izvornog izvora i onemogućiti uspješno izvršenje napada.

Politika istog izvorišta osigurava da JavaScript kod s jednog izvora ne može pročitati ili mijenjati podatke na drugom izvoru. To pruža dodatnu zaštitu korisnicima i sprječava napadače da iskoriste ranjivosti na web stranicama kako bi izvršili zlonamjerne radnje. Međutim, važno je napomenuti da politika istog izvorišta nije jedini sigurnosni mehanizam protiv XSS napada i da razvijatelji web aplikacija također trebaju primjenjivati druge sigurnosne prakse, poput sanitizacije ulaznih podataka i odgovarajuće upotrebe mehanizama za escapiranje znakova, kako bi spriječili uspješnost ovih napada.

13. Na koji napad na webu je osjetljiv sljedeći kod? Što bi trebali napraviti kako bi navedeni kod bio sigurniji?

```
if( isset( $_REQUEST[ 'Submit' ] ) ) {

    $id = $_REQUEST[ 'id' ];
    $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';"

     $result = mysqli_query($GLOBALS["_mysqli_ston"], $query ) or die( '<pre>' .
((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) :
(($_mysqli_res = mysqli_connect_error()) ? $_mysqli_res : false)) . '</pre>' );

    while( $row = mysqli_fetch_assoc( $result ) ) {
        $first = $row["first_name"];
        $last  = $row["last_name"];

        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";

    }
    mysqli_close($GLOBALS["___mysqli_ston"]);
}
```

Kod je osjetljiv na SQL injection - za "id" napadač može unijeti niz znakova koji će izmijeniti SQL upit. Trebali bismo bolje sanirati input, primjerice za "id" dozvoliti samo brojeve, te ne bismo trebali vraćat korisniku previše informacija o njegovom upitu.

14. Objasnite zašto ste u 3. laboratorijskoj vježbi mogli doći do izvorne lozinke korisnika *pablo*!

Do lozinke korisnika "pablo" smo mogli doći jer input nije bio saniran, pa smo mogli izvršiti SQL injection napad. Uz to, lozinke na poslužitelju su bile hashirane slabim MD5 algoritmom.