1. **SQL Injection**

Napad podmetanjem SQL upita (eng. SQL injection) je metoda koja iskorištava sigurnosne ranjivosti kod pristupa web aplikacije bazi podataka. To je najrašireniji i najopasniji napad koji se odvija na aplikacijskom sloju TCP/IP stoga. Spomenute ranjivosti napadač iskorištava preoblikovanjem SQL upita koje web aplikacija šalje bazi podataka tako da otkriju osjetljive podatke ili izvedu nedozvoljene radnje nad njima. Pomoću umetnutih SQL naredbi napadač može preuzeti potpunu kontrolu nad povjerljivim podacima u bazi.

Primjer:

SELECT \* FROM logins   
WHERE username = 'Joe' AND password = 'anything' **OR 1 = 1**

Ovaj **OR 1=1** će omogućiti da je WHERE dio uvjeta uvijek istinit.   
Njega postavljamo da npr. unutar login forme upišemo sljedeće:

**' OR 1 = 1 –**

Ili npr:

**'; DROP TABLE logins --**

Zadnja naredba će izvršiti i SELECT ali i ovaj drugi DROP upit. Rezultat je zapravo:

SELECT \* FROM logins   
WHERE username = 'Joe' AND password = '**'; DROP TABLE logins --**

1. **XSS**

*Cross-site Scripting* (XSS) je vrsta napada na web aplikacije gdje napadač ubacuje JS kod u stranicu. Napadač je često u mogućnosti čitati ili promijeniti osjetljive podatke dostupne web pregledniku, odnosno ukrasti korisničke račune i *cookie* datoteke, zatim brojeve kreditnih kartica. Osim toga, može usmjeriti web preglednik na neke druge adrese ili proslijediti štetan sadržaj iz druge web aplikacije.

Npr. unutar neke forme gdje se unose komentari, napadač unese:

<script>window.location = 'www.hacker.com'</script>

Sljedeći put, kada klijent pristupi stranici gdje se nalazi zlonamjeran komentar, korisnike će se prisilno preusmjeriti na stranicu [www.hacker.com](http://www.hacker.com).

1. **CSRF**

*Cross-Site Request* Forgery (CSRF)  je vrsta napada na web stranice koji se izvodi preko web preglednika klijenta. Pomoću njega napadač može pristupiti ranjivoj stranici i izvesti određene akcije na njoj s ovlastima klijenta. Ako je napadnuto web sjedište banke, a u web pregledniku zapamćena prijava korisnika na to web odredište, moguće je primjerice pokretanje novčane transakcije u ime korisnika. Općenito zaštita od CSRF napada zasniva se na takvom oblikovanju web stranica koje će sigurnosnim provjerama osigurati poštivanje tzv. *same origin* principa. Riječ je o pravilu koje kaže da se web zahtjevi mogu izvoditi samo naspram web odredišta s kojeg su učitani.

1. **Hashiranje (SALT)**

*Hashiranje* je zapravo generiranje sažetka poruke. *Hash* je funkcija koja na ulazu prima niz znakova PROIZVOLJNE duljine a izlaz je niz znakova FIKSNE duljine. Neki od hash algoritama su MD5, SHA-1, SHA-2, SHA-3..).

*Salt* je random niz znakova koji se dodaje hash funkciji kako bi se napadaču onemogućilo jednostavno nalaženje hasheva iz već gotovih riječnika.

*Primjer:*

Ako imamo šifru:

pero 🡪 možemo realno jako brzo naći tu riječ u rječniku

Ako nadodamo salt na kraju:

pero/()afsj827182)=)%&(=) 🡪 tada je rječnik neupotrebljiv

1. **Slowloris za DOS**

Slowloris je DOS napad koji pokušava otvoriti što više konekcija na žrtvu koje drži otvorenima koliko god duže je moguće. Radi na principu da periodički šalje HTTP requestove koji nikad ne završavaju. Žrtva pokušava zadržati uspostavljene veze, popunjujući tako pool konekcijama. Nakon što se pool popuni, daljnje konekcije se ne uspostavljaju.

1. **DDOS**

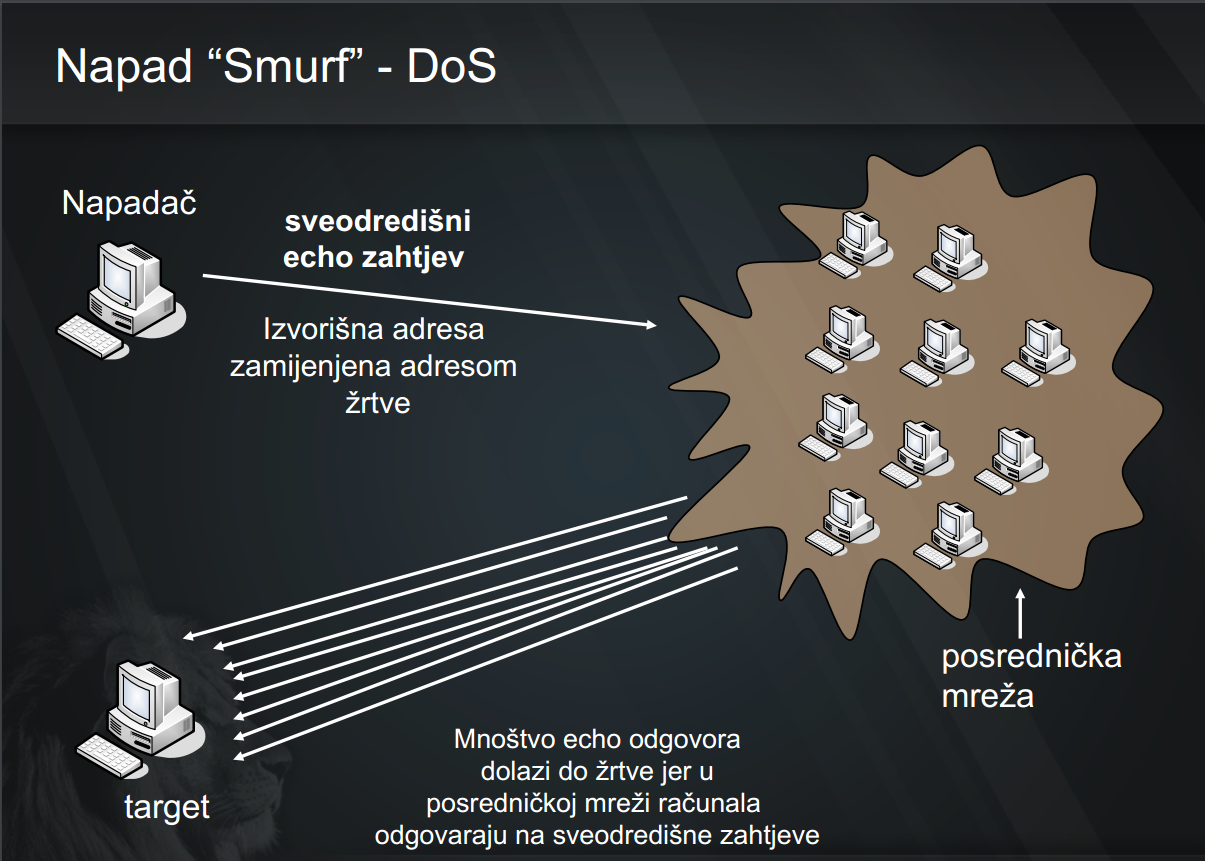
Distributed Denial-of-Service (DDOS) je napad kojim sprječava pristup legitimnim korisnicima pristup nekom serveru ili usluzi. Funkcionira na način da se serveru šalje više zahtjeva nego šta taj server može obraditi. Napadač napad izvodi pomoću botneta (mreža većinom inficiranih računala).

1. **DNS i DNSSEC**

DNSSEC temelji se na korištenju asimetrične kriptografije, javnih i privatnih ključeva. DNS podaci se potpisuju privatnim ključem. Javni ključ se objavljuje putem DNS-a i on se koristi za provjeru valjanosti potpisa odnosno DNS podatka. Na taj način osigurava se valjanost podataka između autoritativnog poslužitelja i resolvera.

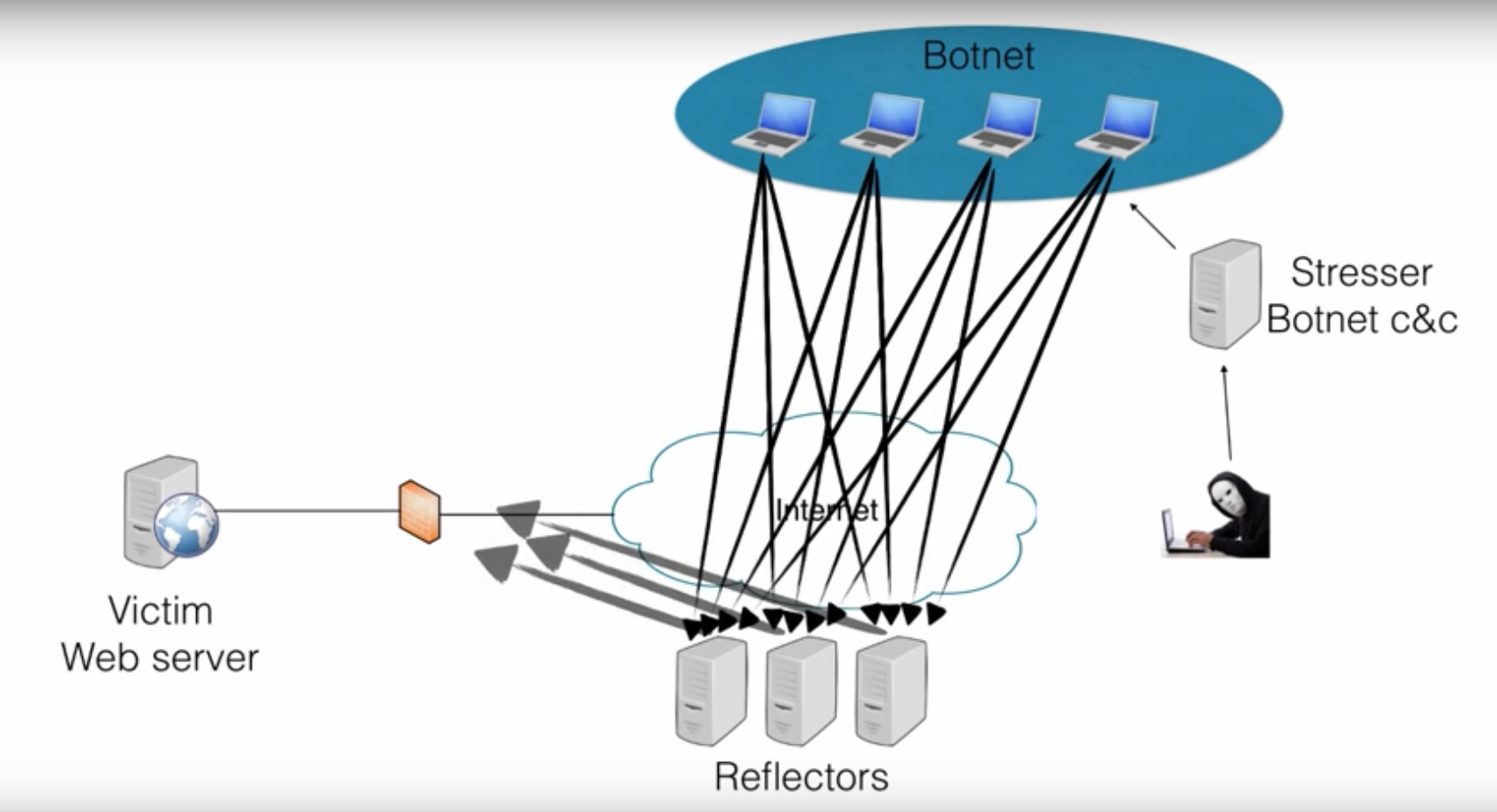
1. **Smurf (DOS napad, ICMP protokol)**

Napad započinje slanjem ECHO zahtjeva na broadcast adresu posredničke mreže s lažiranom izvorišnom adresom koja se postavi na adresu žrtve. Računala u posredničkoj mreži odgovaraju slanjem ECHO odziva 🡪 na adresu žrtve. Na taj način posrednička i ciljna mreža su zagušene prometom.



1. **UDP reflection i amplification**

Servisu koji koristi UDP pošalje se upit s lažiranom izvorišnom adresom a njegov odziv sadrži više podataka od upita. Napad se može izvesti preko botneta – svi šalju UDP upit s lažiranom izvorišnom adresom (žrtvina adresa) na npr. NTP (Network Time Protokol) servis. NTP servis šalje sve odgovore na žrtvu s količinom podataka do 556x većom nego što je bila veličina zahtjeva.



1. **Spoofing**

IP zavaravanje – lažno predstavljanje. Šalju se paketi kojima se promjeni izvorišna adresa 🡪 slanje poruka s tuđim identitetom.

1. **Ping i NMap**

*pojedinačni ping* – ICMP *echo request* / *echo reply*

*sveodredišni ping* – pokušaj otkrivanja aktivnih računala u mreži; broadcast *echo request* / puno *echo reply* odgovora

*nmap* – program za mrežno otkrivanje; koristi IP pakete kako bi otkrio koji su hostovi i servisi vidljivi u mreži

1. **One Time Pad**

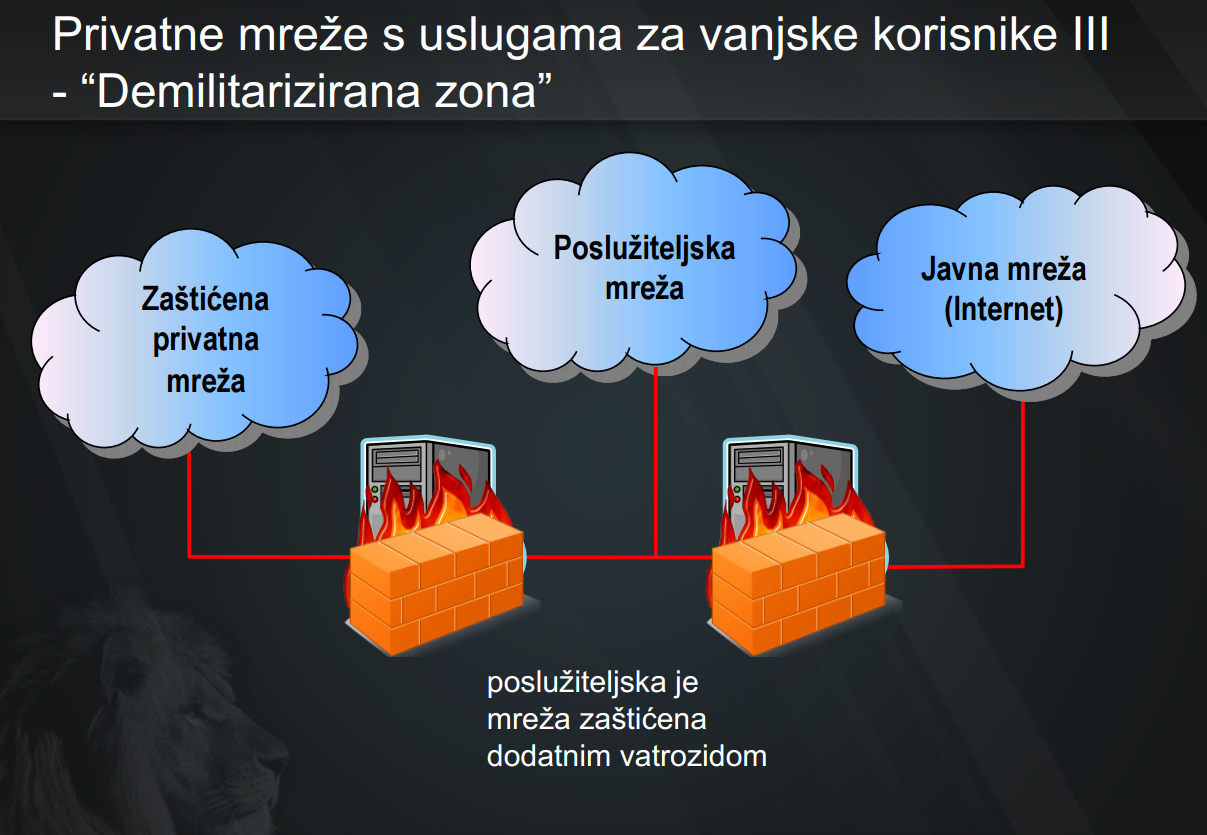
Razmjeni se slučajno generirani ključ (pad). Otvoreni tekst se šifrira preko tog ključa u modularnoj aritmetici (XOR). Šifrat se dešifrira s istim tim ključem u modularnoj aritmetici. Ključ mora biti jednako dug kao i otvoreni tekst.

1. **DMZ**

Demilitarizirana zone (DMZ) je područje mreže između 2 filtera paketa:

* vanjski filter – dopušta samo promet izvana
* interni filter – dopušta samo promet iznutra

Odvaja vanjsku i unutarnju mrežu.



1. **HTTPS (kako radi i na kojem sloju, za što služi, certifikati…)**

HTTPS omogućava kriptiranu komunikaciju i sigurnu identifikaciju. HTTP radi na aplikacijskom sloju ali sigurnosni protokol radi na nižim podrazinama kodirajući poruku prije prijenosa i dekriptirajući poruku pri dolasku. HTTPS se zapravo odnosi na korištenje HTTP preko kriptirane SSL/TLS veze.

Certifikat sadrži javni ključ koji je potreban da bi se započela sigurna sesija. Uz javni ključ, sadrži i ime na koga je izdan certifikat, serijski broj, datum isteka, javni ključ vlasnika i digitalni potpis izdavača certifikata.

1. **URL napadi (A4 i A7) i kako spriječiti**

A4 - Nesigurne reference na objekte  
kada imamo vidljiv ID u URLu pa onda mijenjanjem možemo doći na stranicu koju ne bi trebali moći vidjeti: /users/2 🡪 lako možemo pogoditi /users/3 (i slično…)

Rješenje – koristiti drugačiji format zapisa ID-a (npr. random niz znakova – UUID). Isto tako, treba provjeravati smije li korisnik stvarno pristupiti toj stranici (ako ne, redirectat ga negdje drugo).

A7 - nezaštićen pristup URLovima  
ako npr. prikazujemo u menu neke linkove a možemo do njih doći preko URL-a tada napadač može pristupiti funkcionalnostima i uslugama na koje nema pravo.

Rješenje – nije dovoljno samo sakriti link već potrebno dodati i autorizaciju na sve akcije u aplikaciji

1. **Same Origin Policy**

*Same Origin policy* (politika istog izvora) odnosi se na kod koji se izvršava u pregledniku klijenta. Skripte koje se izvode na ***istoj stranici*** smiju međusobno dijeliti pristup podacima (DOM) dok skripte drugih stranica ne smiju.

Primjer:

*Na nekoj stranici (npr.* **fer.hr***) podmetnut je kod:*

$.post('http://bank.com/transfer', { account: 'pero', ammount: '10000' })

*Tim kodom radi se POST zahtjev na aplikaciju na* **bank.com** *kojim se prebacuje novac na napadačev račun. Ukoliko ne bi bilo provjere izvorišta, podaci za autentikaciju bi se pokupili iz cookiea i zahtjev bi se izvršio prilikom loadanja stranice.*

Ovakva vrsta napada se naziva CSRF (Cross-Site Request Forgery).

Same Origin Policy zahtjeva da se web zahtjevi mogu pokretati samo nad stranicama iz kojih su učitani. Tako je web odredište s kojeg se učita programski kod jedino odredište prema kojem mogu biti upućeni zahtjevi za sadržajem koji je dio učitanog koda.

Zaštita – provjeravati Referer header u HTTP zaglavlju te provjeravati forgery token koji će se generirati u svakoj formi.

1. **Stuxnet**

Maliciozni crv usmjeren na industrijske računalne sustave i bio je odgovoran za izazivanje štete na iranskom nuklearnom programu. Usmjeren je na PLC-ove koji omogućavaju automatizaciju elektromehaničkih procesa kao što su linije na tvornici, centrifuge za odvajanje nuklearnog materijala i sl. Stuxnet je navodno ugrozio iranske PLC tako što je prikupljao informacije o industrijskom sustavu te ih iskoristio kako bi ubrzao centrifugu i time je doveo do raspada (čak petinu iranskih centrifuga za razdvajanje nukl. materijala).

Obično se na ciljani sustav prenosi preko usb-a, zatim se širi preko mreže i skenira Siemens Step7 program koji upravlja PLC-ovima. Ako ne uspije, crv se automatski uspava, inače upada u program, mijenja kodove i naredbe, dok korisniku vraća povratne vrijednosti koje on očekuje.

1. **Ransomware**

Korisniku uskraćuje pristup računalnim resursima i traži plaćanje otkupnine za uklanjanje ograničenja. Neki oblici kriptiraju datoteke, drugi cijeli sustav. Trojan, najčešće se prenosi kao prilog u el. pošti i oslanja se na društveni inženjering.

1. **SMTP/ extended SMTP**

SMTP   
- Specificira način prijenosa poruka između dva računala, ne ovisi o mrežnom protokolu i omogućuje prosljeđivanje poruka kroz raznovrsne mreže.

-Strogo definira sintaksu i redoslijed odvijanja transakcije – polazno računalo šalje SMTP naredbe na koje ciljno računalo odgovara kodovima (uspjeh ili pogreška – 250 Ok, 221 Bye itd.), svaka naredba MORA dobiti odgovor primatelja.

- čvorovi: MUA i MTA

-naredbe: HELO, MAIL. RCPT, DANA, RSET, VRFY, NOOP, QUIT

-sigurnosni problemi: nema sigurnosnih mehanizama, otvoren i u tekstualnom obliku, nema autentifikacije, podrazumijeva se povjerenje i suradnja

Extended SMTP  
-definira proširenja:

• 8BITMIME — 8-bitni prijenos podataka, RFC 1652

• ATRN — autentificirani TURN za On-Demand Mail Relay,

RFC 2645

• SMTP-AUTH — autentificirani SMTP, RFC 4954

• CHUNKING — cjepkanje velikih poruka, RFC 3030

• DSN — obavijest o dostavi, RFC 3461

• ETRN — udaljeni TURN, RFC 1985

• HELP — pomoć, RFC 821

• PIPELINING — slanje više naredbi odjednom, RFC 2920

• SIZE — deklaracija veličine poruke, RFC 1870

• STARTTLS — Transport layer security, RFC 3207

• UTF8SMTP — korištenje UTF-8 u zaglavljima, RFC 5336

1. **Secure element**

Sigurnosni hardware u koji se smještaju sigurnosno zahtjevne aplikacije i njihovi povjerljivi i kriptirani podaci. Postoje tri različite forme SE-a:

MicroSD – prilagođena SD kartica s dodatnom sigurnom memorijom i mogućnosti za NFC

UICC (Universal Integrated Circuit Card) – opća namjena SIM kartica + dodatna sigurna memorija

Ugrađeni SE – SE čip direktno zalemljen na matičnu ploču

1. **NAT**

Pretvorba privatnih IP adresa (iz privatne mreže) u globalno jedinstvene IP adrese – više računala iz privatne mreže pristupa Internetu preko jedne javne IP adrese. Na taj način ujedno se i skrivaju interna računala (lokalna 🡪 javna IP). U tablici (na switchu) se čuva podatak o preslikavanju socketa na unutarnjem i vanjskom sučelju.

NAT je u osnovi proxy – jedan host šalje zahtjeve u ime svih internih računala (implementiran na transportnom sloju)

**Načini rada:**

Statička translacija – preslikava područje javnih IP adresa u blok iste veličine s privatnim adresama, jednostavna statička translacija za svaku korištenu IP adresu   
Port forwarding – statička translacija gdje se prosljeđuje promet za samo određeni port, a ne za cijelu IP adresu

Dinamička translacija – veća grupa internih klijenata dijeli jednu IP adresu u svrhu skrivanja identiteta ili proširivanja prostora raspoloživih mrežnih adresa. Portovi na jednoj IP adresi mogu se proslijeđivati na specificirane privatne IP adrese

Balansiranje opterećenja – Firewall odabire kojem od poslužitelja iz poola treba proslijediti zahtjev

Mrežna redudancija – balansiranje opterećenja linkova prema ISPu i automatsko prebacivanje na drugi link.

1. **Virusi**

Računalni program, ne funkcionira kao samostalna aplikacija nego svoj kod dodaje postojećim datotekama (npr. notepad), pokreće se isključivo djelovanjem korisnika, jednom pokrenut traži druge datoteke na računalu koje može inficirati (self replication), širi se djelovanjem korisnika (mailovi – privitak + društveni inženjering, mediji za pohranu, mreža - download)

1. **OS fingerprinting**

TCP napad, otkrivanje operacijskog sustava računala, poznavanjem OS i sigurnosnih propusta koji u njemu postoje napasti računalo. Koriste se razne kombinacije i opcije u paketima koji se šalju – u pravilu nestandardne ili nedozvoljene kombinacije, različiti operacijski sustavi različito odgovaraju. Nmap se koristi za ovaj napad

1. **Kako rade antivirusi?**

Antivirus radi tako da analizira datoteke s izvršnim programskim kodom sa ciljem otkrivanja zlonamjernog koda. Oni zapravo raspoznaju samo prisutnost zlonamjernog koda u bilo kojoj datoteci na računalu.

Postoje 2 načina zaštite (i otkrivanja malwarea):

* **prepoznavanje koda** ( eng. *fingerprinting* ) → svaki virus ima određeni znakovni kod pa ga po tome može detektirati i nakon toga ili briše virus iz datoteke ili stavlja datoteku u karantenu ili briše inficiranu datoteku. Uspješnost ovog načina zaštite ovisi o prisutnosti znakovnog koda u bazi poznatih virusa, ali se pokazao veoma koristan zato što se najveći broj napada na sustave izvodi zapravo poznatim virusima.
* **heuristika** ( prepoznavanje ponašanja ) → nadzor ponašanja svih programa, problem je što se sumnjivim akcijama proglašavaju i one legitimne. Ova se metoda zapravo zasniva na prepoznavanju naredbi koje su označene kao potencijalno opasne, odnosno na traženju dovoljno dobrog rješenja koje ne mora davati dobar rezultat u svim slučajevima. Primjer potencijalno opasne naredbe je prepisivanje drugog izvršnog koda što može, ali ne mora biti virus.

1. **Malware kod mobitela**

Prvi malware Mosquito 2004. godine, trojanac skriven u igrici, slanje sms-a na premium brojeve  
U odnosu na računala pristup lokaciji i pristup pokretnoj mreži. Prijenos: e-mail, linkovi, instalacija naizgled korisnih aplikacija, bluetooth, jailbreaking i sl.  
Što radi: krađa lozinki i drugih povjerljivih podataka, phising, korištenje uređaja kao botneta, uništavanje uređaja.   
Rješenje: potpisivanje aplikacija i koncept Storea (app store, Play store)   
Problemi: download aplikacija s 3rd party storea   
Android najugroženiji zbog velikog broja korisinika

1. **SSL/TLS**

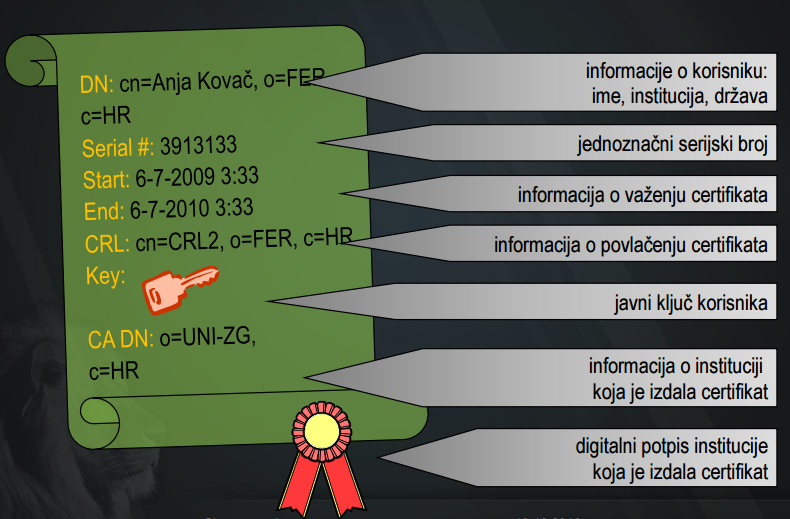
Sigurnosni protokol za uspostavljanje sigurnog i šifriranog komunikacijskog kanala.   
Osigurava autentifikaciju poslužitelja (i klijenta opcionalno) preko certifikata, privatnost podataka (šifriranje korištenjem tajnog dijeljenog simetričnog ključa) i cjelovitost podataka (MAC poruka)

Dva sloja protokola:   
handshake: početna autentifikacija i prijenos ključeva, autentifikacija poslužitelja klijentu, dogovoranje kriptografskh algoritama, generiranje dijeljene tajne, uspostavljanje šifrirane SSL konekcije  
record: fragmentacija, komprimiranje, integritet, autentifikacija i šifriranje

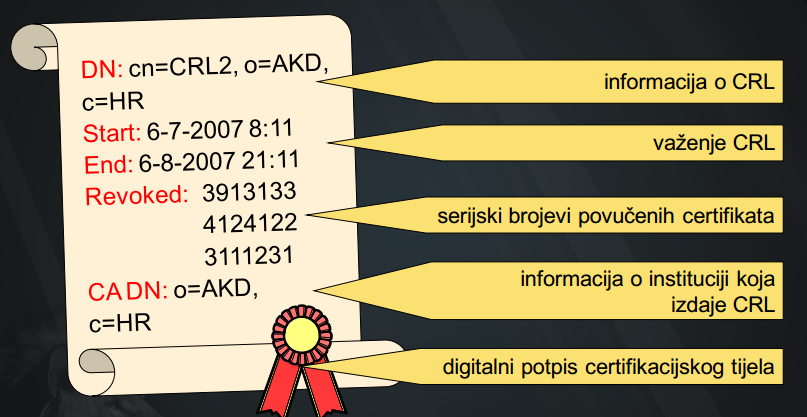
1. **PKI (Infrastruktura javnog ključa) i Digitalni certifikati**

Problem upravljanja ključevima: kreiranje, distribucija, korištenje, arhiviranje, automatizacija životnog ciklusa  
PKI:  
• Ima li privatni ključ samo odgovarajuća osoba? – **kriptografski uređaj**   
• Kako povezati javni ključ sa osobom? – **certifikat**  
• Vrijedi li nečiji javni ključ? Da li je opozvan? – **lista opozvanih certifikata - CRL**   
• Tko će izdati i jamčiti za certifikat? – **certifikacijsko tijelo – certification authority – CA**• Tko će jamčiti identifikaciju i autentifikaciju osobe? – **registracijsko tijelo – RA**• Kako dobiti / distribuirati javni ključ ? – **javni imenik**   
• U koju svrhu mogu koristiti ovaj certifikat? – **certificate policy – CP**

-certifikat sadrži javni ključ i ostale informacije o subjektu, izdavatelju i valjanosti te povezuje javni ključ s osobom i potvrđuje njeno ime  
-osoba kojoj pripada ime je **subjekt** certifikata (subjekt može biti fizička osoba, izdavatelj certifikata, organizacija, korisnički račun, uloga, poslužitelj, uređaj, aplikacija)  
-certifikat izdaje CA (certifikacijsko tijelo) i on ga potpisuje svojim potpisom

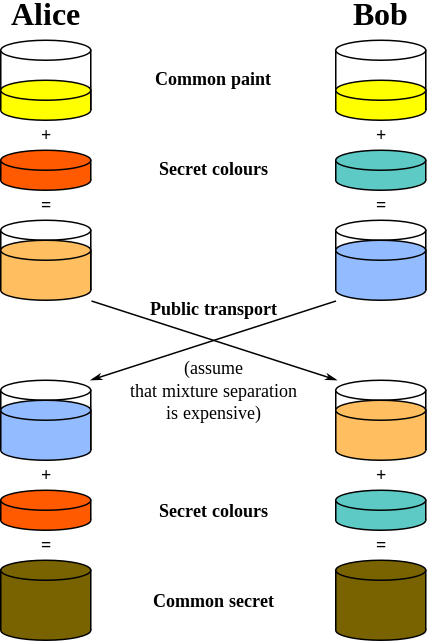


Važenje certifikata:   
• javni ključevi – mogu se koristiti kroz dugi vremenski period (desetljeća)   
• privatni ključevi – trebaju imati što kraće vrijeme važenja   
• opoziv ključa – ako je ključ kompromitiran treba ga opozvati – ako je ključ opozvan, potpisani dokument ne vrijedi (osim kada ima vremensku oznaku, timestamp) – ako je ključ opozvan, svi dokumenti koji su njime šifrirani su kompromitirani   
• provjera certifikata – obavezna! (pošiljatelj ili primatelj) – treba uključivati provjeru važenja certifikata i provjeru potpisa certifikata – ako je potrebno arhivirati potpisani dokument – time stamp  
  
CRL (Lista opozvanih certifikata):



1. **Diffie-Hellman algoritam**

Algoritam koji koristi asimetričnu kriptografiju za prijenos zajedničke tajne (simetričnog ključa).   
Primjena: dvije strane žele sigurno komunicirati i moraju razmijeniti tajni ključ (razmjena ključa preko nesigurnog kanala)



Matematički:

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo *p*, where *p* is prime, and *g* is a [primitive root](https://en.wikipedia.org/wiki/Primitive_root_modulo_n" \o "Primitive root modulo n) modulo *p*. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to *p*–1. Here is an example of the protocol, with non-secret values in blue, and secret values in **red**.

1. Alice and Bob agree to use a modulus *p* = 23 and base *g* = 5 (which is a primitive root modulo 23).
2. Alice chooses a secret integer ***a*** = **6**, then sends Bob *A* = *g****a*** mod *p*
   * *A* = 5**6** mod 23 = 8
3. Bob chooses a secret integer ***b*** = **15**, then sends Alice *B* = *g****b*** mod *p*
   * *B* = 5**15** mod 23 = 19
4. Alice computes ***s*** = *B****a*** mod *p*
   * ***s*** = 19**6** mod 23 = **2**
5. Bob computes ***s*** = *A****b*** mod *p*
   * ***s*** = 8**15** mod 23 = **2**
6. Alice and Bob now share a secret (the number **2**).
7. **Asimetrična kriptografija**

Kod asimetrične kriptografije koriste se dva ključa, jedan za šifriranje, a drugi za dešifriranje. Kod ove kriptografije uvode se privatni i javni ključ. Pošiljatelj može šifrirati svoju poruku javnim ključem primatelja, a primatelj dešifrira poruku svojim privatnim ključem. Iz javnog ključa nemoguće je generirati privatni ključ, obrnuti postupak je moguć.

Asimetrična kriptografija se koristi kod: prijenosa podataka, autentifikacije, digitalnih potpisa i razmjene tajnih ključeva (uspostavljanje sjednice).

1. **Šifriranje tokova**

Iz ključa K generira se pseudoslučajni niz bitova. S tim nizom se šifrira bit po bit u modularnoj aritmetici na temelju unutarnjeg stanja. Primatelj s istim nizom dešifrira bit po bit.

Postoje dvije vrste šifriranja tokova:

**sinkroni** – pseudoslučajni niz ne ovisi o otvorenom tekstu ili šifratu, nego se generira nezavisno, mora postojati sinkronizacija obje strane

**samosinkronizirajući** (asinkroni) – niz od N prethodnih bitova šifrata koristi se za izračun ključa, lakše se vraća sinkronizam u slučaju pogreške

Sigurnost:

- period ponavljanja mora biti velik

- ključ se ne smije otkriti na temelju niza bitova

- isti niz bitova se ne smije koristiti dva puta

- ne osigravaju integritet, nego samo privatnost

Koristie se kod: GSM – A5, TLS ili WEP – RC4

1. **CIA narušavanje kod maila**

povjerljivost(confidentiality) - čitamo nečije povjerljive podatke (npr. SMTP se prenosi u plaintextu)  
cjelovitost,integritet(integrity) - mjenjamo sadržaj nekih informacija, možemo promijeniti i sadržaj IP paketa (opet SMTP di bi mogli mijenjat sadržaj maila)  
raspoloživost(availability) - raspoloživo - DoS napad ili požar - više nije raspoloživo

1. **PGP/Web of Trust**

PGP je mehanizam koji osigurava povjerljivost komunikacije elektroničkom poštom s kraja na kraj. Omogućava 5 osnovnih usluga: autentifikaciju, šifriranje, sažimanje (kompresiju), kompatibilnost s infrastrukturom elektroničke pošte, segmentaciju i ponovno slaganje poruke. Povjerenje u ključeve temelji se na modelu web of trust.

Web of Trust je model mreže povjerenja čiji je smisao mreže decentralizirana razmjena javnih ključeva. Pojedinci jedni drugima potpisuju ključeve, nema središnjeg autoriteta za potpisivanje ključeva, PGP izračunava razinu povjerenja za svaki ključ na “privjesku” ( PGP Key Ring ). Korisnici sami interpretiraju razine povjerenja.

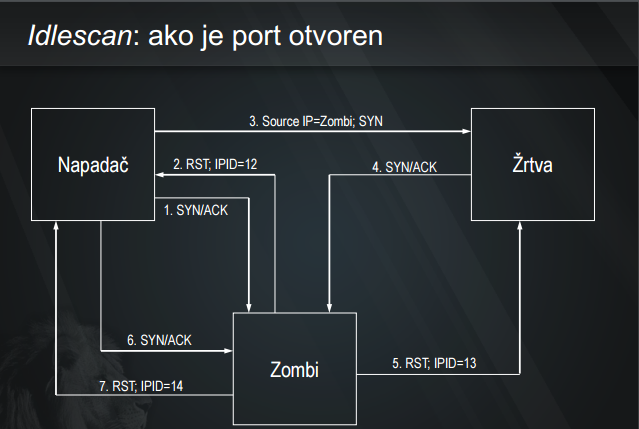
1. **Syn Flooding**

Šalju se SYN segmenti s izmijenjenom IP adresom pošiljatelja, izmijenjena IP adresa odgovara nepostojećem računalu te žrtva nikada neće dobiti odgovor na svoj SYN-ACK. DoS – ostaju poluotvorene konekcije (spremaju se u memoriji i kad se ona popuni nove konekcije se odbijaju)

1. **TCP skeniranje portova**

Cilj: prikupiti informacije o žrtvi i na temelju podataka naći slabosti koje će omogućiti napad

Metode:   
**TCP connect ()** – uspostavlja se potpuna veza s portom, lako se otkriva  
**TCP SYN** – poluotvoreno skeniranje, šalje se paket SYN, ako dobijemo kao odgovor RST port nije otvoren, ako dobijemo SYN/ACK port je otvoren i šaljemo RST za raskidanje veze  
**TCP FIN** – ako se poluotvoreno skeniranje bilježi na FW ovo se vjerojatno ne bilježi, ako na paket FIN sustav odgovori RST, port je zatvoren, a ignorira ga ako je otvoren (na nekim sustavima u oba slučaja odgovara RST pa se ne može zaključiti)  
Idlescan – metoda skeniranja u kojoj žrtva ne primi niti jedan paket s IP adresom napadača (zombi računalo). Ideja skeniranja: praćenje IP ID broja



1. **ARP napad**

-Moguće je poslati odgovor prije pravog računala te vratiti lažno preslikavanje adresa (IP/MAC)  
-Lažni ARP upiti mogu se koristiti za spremanje krivih ARP preslikavanja na računalu kome su upućeni   
-ARP poruke mogu se slati kontinuirano kako bi se lažni podaci zadržali u cacheu  
-Preusmjeravanje prometa koje bi trebalo ići na router   
-Može se koristiti za lažno preslikavanje gatewaya na nepostojeću MAC adresu (DOS)

1. **S/MIME**

Sigurnosno proširenje standarda MIME, nije ograničeno na e-postu, osigurava povjerljivost s kraja na kraj. Omogućuje:

* Koristenje svih znakova
* Definiranje strukture poruke i vrstu poruke
* Dodavnje binarnih, HTML ili visemedijskih datoteka
* Prema specifikacije RFC 821 i 822

Jedini zahtjev **klijent mora biti kompatibilan sa standardom**

Dvije glavne komponente

* Digitalni potpis (autentifikacija, cjelovitost i neporecivost)
* Šifriranje (privatnost, sigurnost podataka)

Funkcije

* *enveloped-data* – osigurava privatnost i sigurnost podataka

1. Stvara se jednokratni ključ 3DES ili RC2/40
2. Ključ se šifrira s javnim ključem primatelja
3. sve potrebne informacije (certifikat pošiljatelja, algoritam, šifrirani ključ) pohrane se u vrijednost *RecipientInfo*
4. Šifriranje sadržaja poruke s jednokratnim ključem

* *Signed-data* – osigurava autentičnost, cjelovitost i neporecivost. Primatelj mora podržavati S/MIME za čitanje i verifikaciju potpisa.
  1. Odabir algoritma SHA1 ili MD5
  2. Digitalno potpisivanje sažetka ( šifriranje našim privatnim, zatim primateljevim javnim ključem)
  3. podaci o potpisniku (certifikat, identifikator algoritma, šifrirani sažetak) pohranjuju se u vrijednost *SignerInfo*
  4. Sažetak i sadržaj kodiraju se prema *base64*
* *Clear-signed-data* – osigurava autentičnost, cjelovitost i neporecivost. Primatelj mora podržavati S/MIME za verifikaciju potpisa ali ne i za čitanje poruke.
  1. Odabir algoritma SHA1 ili MD5
  2. Digitalno potpisivanje sažetka ( šifriranje našim privatnim, zatim primateljevim javnim ključem)
  3. podaci o potpisniku (certifikat, identifikator algoritma, šifrirani sažetak) pohranjuju se u vrijednost *SignerInfo*
  4. Samo se sažetak kodira se prema *base64*

mogući napadi:

– prijava pod tuđim imenom (class 1 certifikati)

– korištenje jednog certifikata, potpisivanje drugog korisnika

– krivotvorenje zaglavlja poruke