**ZOSS 2**

**ALEKSANDAR**

**Slajd 4**

Kriptografija se koristi od davnih dana kao sredstvo pisane tajnosti. Još od svima poznatih hijeroglifa, preko Cezarovog šifrovanja, pa sve do Enigme. Ona ima četiri glavna zadatka gde se prva tri objašnjavaju preko CIA trijade (Confidentiality, Integrity, Authentication), a četvrti jeste Nonrepudation, tj. mogućnost da primalac bude siguran da je ta poruka poslata baš u tom obliku od strane određenog pošiljaoca koji ne može da negira slanje te poruke. U današnje vreme se koriste veoma kompleksni algoritmi koji obezbeđuju visok nivo tajnosti, kao što su AES-256 i Triple-Des.

Veoma često ljudi misle da je kriptografija mnoštvo komplikovane matematike, što je donekle i tačno, ali dobra strana je što nije potrebno poznavati tu matematiku ili detalje kriptografskih algoritama. Dovoljno je samo poznavati neka ključna svojstva algoritma kako bi ih uspešno koristili.

Jedni od glavnih pojmova u vezi sa kriptografijom su: simetrična i asimetrična kriptografija, kao i hešovanje.

Bitno je napomenuti da nije potrebno koristiti enkripciju radi korišćenja, nego da znamo zašto je koristimo. Potrebno je poznavati sistem i samim tim i njegove slabosti, nivoe pretnje u sistemu, osetljive podatke i gde se oni čuvaju i slično. Ali isto tako treba koristiti odgovarajuće mehanizme, standardne jake algoritme, redovno menjati i čuvati ključeve...

**Slajd 5**

Primena kriptografije

1. **Tajnost i integritet u skladištu**

Uz enkripciju podataka u prenosu, potrebno je zaštititi podatke i prilikom mirovanja. To podrazumeva korišćenje uređaja koji automatski kodiraju sve podatke čuvane na disku računara i prilikom njegovog pokretanja je potrebno da korisnik unese ključ kako bi podaci bili dekodirani. Ovo sprečava pristup podacima na disku ukoliko dođe do njegove krađe. Takođe, moguće je koristiti razna šifrovanja nad zasebnim fajlovima i folderima.

Ovo za sobom povlači potrebu redovnog menjanja ključeve i/ili načina kodiranja kako bi se obezbedila sigurnost podataka. Međutim, ukoliko korisnik zaboravi ključ, podaci nemaju smisla i postaju beskorisni. Zbog ovoga se savetuje da postoji *backup* kriptovanih podataka u obliku običnog teksta. Na ovaj način su podaci kodirani samo dok se nalaze u skladištu, ali ne i dok se koriste.

Integritet u skladištu je ostvaren uz pomoć sistema za kontrolu pristupa, sa bravama i ključevima i ostalim vidovima zaštite da se onemogući neautorizovani pristup skladištenim podacima. Međutim, u savremenom dobu je potrebno zaštititi podatke i od napada preko mreže. Korišćenjem kriptografskog *checksum*-a se utvrđuje validnost podataka u skladištu, koji su dosta ranjiviji od podataka u prenosu, zbog dužeg izlaganja (*exposure*) i veće količine informacija koje sadrže.

1. **Tajnost i integritet u prenosu**

Kriptografiju je moguće koristiti kako bi se osiguralo da podaci ne budu menjani tokom njihovog prenosa, tj. da se održi integritet. Na ovaj način se onemogućava slučajna ili namerna izmena podataka koja bi dovela do nepoželjnih dešavanja. Takođe, potrebno je osigurati i uspostavljanje same konekcije prilikom razmene podataka i većina sistema koristi kriptosistem sa privatnim ključem, tj. simetričnu kriptografiju. Alternativni pristup je korišćenje master ključa koji se koristi za uspostavu sigurne i bezbedne konekcije (ključ sesije), a mnogo sigurnije jeste korišćenje kriptografije sa javnim ključevima, tj. asimetrična kriptografija.

Jedan primer primene jeste korišćenje prethodno pomenutog *checksum*-a.

Slajd 6

1. **Autentifikacija**

Autentifikacija je proces dokazivanja identiteta korisnika sistema. Ovo se lako postiže korišćenjem lozinki, koje se u savremenim sistemima čuvaju u šifrovanom obliku. Na ovaj način, iako dođe do "curenja" podataka, lozinke ostaju tajne. Naravno, sve veći broj sistema koristi takozvanu višefaktorsku autentifikaciju. Lozinke su analogne ključevima u kriptosistemima, jer omogućavaju pristup podacima onima kojima je poznaju.

1. **Sistemi akreditiranja**

Akreditacija je dokaz o kvalifikaciji ili sposobnosti koji se vezuje za osobu da bi se pokazalo da je pogodan za nešto (npr. vozačka dozvola ili pasoš). Elektronska akreditacija omogućava elektronsku potvrdu verodostojnosti neke tvrdnje (*claim*). Ovakav sistem nije samostalan već se koristi zajedno sa drugim uređajima kao što su pametne kartice, koje vrše kriptografske funkcije i koriste se za čuvanje tajnih informacija.

Slajd 7

1. **Digitalni potpisi**

Digitalni potpis je mehanizam pomoću kog se poruka autentifikuje, tj. dokazuje da poruka dolazi od nekog određenog korisnika. Da bi bio efikasan kao potpis na papiru, potrebno je da bude težak za falsifikovanje. Uglavnom se koristi kada strane koje učestvuju u nekoj transakciji nisu fizički blizu. Digitalni potpisi se mogu kreirati uz pomoć korišćenja sistema javnih ključeva i procesa hešovanja.

1. **Elektronski novac**

Elektronski transfer sredstava (EFT – Electronic Funds Transfer) predstavlja elektronsku razmenu novca između dva naloga preko sistema baziranog na računarima. Ovo uključuje plaćanja preko mreže, plaćanje debitnim karticama, transakcije korišćenjem bankomata i slično. Očigledno da bilo kakav napad na jedan ovakav sistem može da dovede do ekonomske propasti na nacionalnom nivou u veoma kratkom periodu i zato je značaj sigurnosti i bezbednosti u ovakvom sistemu neopisivo visok.

Jedno od važnih svojstava keša (novac u fizičkom obliku) jeste anonimnost. Korišćenjem keša, prilikom preuzimanja novca iz banke, ne zna se na šta će korisnik trošiti te pare, zatim, prodavac ne zna ko smo mi i ne traži nikakvu potvrdu ko je uplatilac. S druge strane, korišćenjem platnih kartica, prodavac zna ko plaća, ali isto tako i banka zna od koga mi nešto kupujemo. Samim tim se ne održava anonimnost i ne štiti se privatnost korisnika. Zbog ovoga je bilo potrebno uvesti metode enkripcije kako ove i slične informacije ne bi dospele u "pogrešne ruke".

Baš iz tih razloga je Dejvid Čaum formulisao "zaslepljeni" potpis. On predstavlja poseban oblik kriptografskog potpisa koji omogućava potpisivanje virtuelnog novčića, bez da potpisnik vidi taj novčić i time je omogućeno korišćenje digialnog novca koji pruža anonimnost i nemogućnost praćenja. Ovaj oblik valute je poznat pod nazivom digitalna gotovina (Digital Cash).

**DEMO**

**SLAJD 9 I DALJE:**

**SLAJD 9: UVOD, PROCITAS NASLOV**

**Napadi, posledice i najbolje preporuke**

**SLAJD 10:**

**Kao što je prikazano u prethodnom demo primeru, a i inače kada su kriptografski napadi u pitanju glavni cilj jeste krađa osetljivih podataka. Pored slabih kljuceva, slabih algoritama, slabih tehnika hesovanja lozinki, jedan od najcescih razloga dolaska do ovakvih jeste jednostavno nekripotvanje osetljivih podataka. Iako to ne deleuje kao slučaj, realnost jeste da se podaci i dalje ne kriptuju pri prenosu ili u skladistu. Pored toga najvece greske prilikom rukovanja osetljivim podacima jesu korisćenje lose kriptografije, tj. Slaba, provaljenja ili rizicna enkripcija.**

Kao u prethodnom demo primjeru, a i inače kada su kriptografski napadi u pitanju glavni cilj je krađa osjetljivih podataka. Pored slabih kljuceva, slabih algoritama, slabih tehnika hesovanja lozinki, jedan od najcescih razloga dolaska do ovog napada jeste jednostavno nekriptovanje osetljivih podataka. Iako ne izgleda da je slučaj, ipak i dalje se dešava da se podaci ne kriptuju pri prenosu ili u skladištu. Pored toga, najveće greške prilikom rukovanja osjetljivim podacima jesu korišćenje loše kriptografije, tj. slaba, provaljena ili rizična enkripcija.

**SLAJD 11:**

**Brute force and rainbow attack**

**Prema sporevednim istraživanjima 80% hakerskih napada u poslednjoj godini izvršeno je upotrebom prethodno ukradenih lozinki. Jedna od čestih taktika koja se koristi za otkrivanje i krađu lozinki jeste Brute force attack, gde se mnoštvom nasumičnih pogađanja dolazi do lozinke.**

**Pored brute force napada još jedan vid napada jeste Rainbow table attack. Rainbow tabele su tabele koje u sebi sadrže unapred izračunate heš vrednosti. Pomoću njih napadači mogu da od hash vrednosti dođu do lozinke u plain tekstu upotrebom jednostavnih operacija poređenja.**

**Slajd 12:**

Glavne posledice loše kriptografije mogu se kategorisati u 4 grupe:

Narušavanje poverljivosti, gde dolazi do pristupa informacija ili komunikaciji od strane neautorizovane osobe.

Narušavanje integriteta podataka gde dolazi do neautorizovane izmene podataka u skladištu ili tranzitu između pošiljaoca i nameravanog primaoca, bez da to bude primjećeno od obe strane.

Narušavanje autentifikacije gde dolazi do krađe identiteta pa se pristigli tj. preuzeti podaci ne mogu smatrati validnim obzirom da ne dolaze od pouzdanog izvora.

Narušavanje neporecivosti, pa se omogućuje da kreator nekih podataka ili aktivnosti kasnije to i porekne.

**Slajd 13:**

Osim narušavanja prethodno opisanih koncepata, posledice lose kriptografije mogu se razvrstati i u sledece kategorije.

Kradja inforimacija (neke poslovne tajne, sistemski podaci..),

Kradja source koda,

Narusavanje reputacije (neke organizacije, firme..)

I kradja intelektualne svojine.

Medjutim, najcesca posledica i nejacesci cilj napadaca jeste pristup privatnim informacijama individualaca.

Pristup podacima moze biti od strane osobe koja nije eksplicitno autorizovana ili kojoj nije implicitno dato odobrenje od strane osobe ciji su podaci.

Privatne licne podatke je neophodno zastititi bilo da je osoba korisnik proizvoda ili da su njegovi podaci deo podataka obradjivanog od strane proizvoda.

Izlozenost nekih podataka i dalje je zabranjena zakonom.

Tipovi privatnih informacija:

* + Identifikatori
  + Kontaktne informacije, kao sto su adrese i brojevi telefona
  + Geografska lokacija (gde je trenutno korisnik ili koja je mesta posecivao)
  + Istorija zaposlenja
  + Finansijski podaci – broj kreditne kartice, plata, bankovni nalozi...
  + Slike, audio i video snimci
  + Sabloni ponasanja – istorija pretrazivaca, kada se obavljaju odredjene aktivnosti...
  + Komunikacije – e-mail adrese, privatne poruke, sms-ovi, chat
  + Zdravstvene informacije – zdravstveno stanje, status osiguranja, izdavani recepti..
  + Sifre naloga i ostali kredencijali

Neke od ovih informacija mogu biti kategorizovane kao PII (Personally Identifiable Information), PHI (Protected Health Information). Kategorije nekih privatnih informacija mogu varirati u odnosu na nameru i nacin zloupotrebe istih, ili u odnosu na politike i praksu odredjene industrije.

**SLAJD 14:**

Salting:

Salt je jedinstven nasumično generisan string koji se dodaje na lozinku u procesu hesovanja. Salt je jedinstven za svakog korisnika, što u mnogome ograničava napadača. Na primer ukoliko dva korisnika koriste istu lozinku, hash vrednosti lozinki nece biti jednake, jer hash nastaje primenom hash funkcije na lozinku i dodati salt, a ne samo na lozinku. Na taj način, napadač je sprečen da istovremeno provali više lozinki, već samo jednu po jednu. Tako se vreme potrebno za razbijanje lozinki povećava u direktnoj proporciji sa brojem heševa. Kada je u pitanju salt, preporuka je da minimalna dužina bude 16 karaktera.

**Slajd 15:**

Peppering:

Kada je u pitanju zastita podataka, posotji jos jedan dodatni sloj, peppering. Pepper je sličan salt-u, ali ima dve ključne razlike:

* Salt je jedinstven za svaku lozinku, dok je pepper zajednički za sve uskladištene lozinke
* Za razliku od salt-a pepper se ne čuva u bazi podataka, vec u nekim konfiguracionim fajlovima

PRATITI SLIKU SA ANIMACIJOM I PRICOM:

Ukoliko imamo neku lozinku koju je potrebno sacuvati u bazi podataka, na nju cemo prvo konkatenirati salt, a zatim i pepper, te dobijamo rezulat koji predstavlja konkatenaciju ta 3 stringa. Zatim se na taj rez primeni jednosmerna hash funkcija, ciji se rezultat upisuje u bazu. Potrebno je primetiti da se pored hesirane lozinke, u bazi cuva i salt, dok se pepper cuva u nekom konf fajlu koji je svakako potrebno adekvatno zastititi.

Pepper se koristi kako bi sprečio napadača da dođe do lozinki ukoliko na neki način pristupi samo bazi podataka(na primjer ukoliko uspješno izvrši SQL Injection ili pristupi backup-u baze podataka).

Tradicionalno pepper se pre samog procesa hashovanja konkatenirao sa lozinkom, pa se tek nakon toga vrsilo hesovanje. Nesto alternativniji pristup bi bio da se prvo hesuje lozinka kao i inace, a zatim da se hash enkriptuje upotrebno simetricnog algoritma enkripcije, gdje bi kljuc bio prethodno definisani pepper, pa tek nakon toga da se podaci uskladište u bazu.

**SLAJD 16:**

Prethodno je pomenuto generisanje random vrednosti pri upotrebi salta i peppera, pa cemo se sada kratko osvrnuti i na pseudo random number generator kao i cryptographiacly secure random generator. PRNG obezbedjuje nizak nivo slucajnosti prilikom generisanja vrednosti, te su kao takvi ovi generatori brzi ali ih ne bi trebalo korisititi u sitacijama koje zahtevaju veliku sigurnost obzirom da napadac moze da predvidi koja ce vrednost biti generisanja.

CSPRNG su dizajnirani za situacije u kojima je potrebna velika sigurnost i veci kvalitet slucajnosnih vrednosti, tj vecu entropiju. Oni su takodje sporiji, i vise opterecuju sam CPU, sto ponekad moze da rezultuje blokadi procesora ukoliko je potrebno izracunati veliku kolicinu podataka. Ne bi trebalo koristiti ovaj generator ukoliko je potrebno izracunati veliku kolicinu random vrednosti koja nije povezana direktno sa sigurnoscu.

TABELA procitas

**Slajd 17:**

Takođe kod algoritama heširanja važno je odabrati odgovarajući work factor, odnosno pronaći balans između nivoa sigurnosti i performansi koje se obezbjeđuju. Veći work factor će otežati napadaču da provali hash, ali sa druge strane će biti potrebno više vremena da se sam hash izračuna.

Visok work factor dovodi do degradacije performansi aplikacije i ostavlja prostora da potencijalni napadač prouzorkuje denial od service.

Stoga, Ne postoji zlatno pravilo kada je u pitanju izbor work factora. Izbor će svakako zavisiti od performansi servera i broja korisnika aplikacije. Svakako, potrebno je izvršiti odgovarajuća testiranja. Ono što je neko generalno pravilo i preporuka jeste da računanje hasha treba da traje manje od sekunda, osim ako su u pitanju sajtovi sa velikim saobraćajem, tada taj vremenski interval treba da bude znatno kraći.

**SLAJD 18:**

Prethodno su izlozeni neki od koncepata hesiranja. Ono sto bi bila preporuka kada je hesiranje u pitanju jeste da se uvek kada god je to moguce koristi moderan hes algoritam. Neki od pomenutih jesu Argno2, PBKDF2, bcrypt, scrypt obzirom da oni u sebi vec implementiraju pomenute koncepte.

Ukoliko nije moguce korisititi neki od pomenutih alogoritama, pozeljno je pratiti sledece korake:

ANIMACIJA

1. Koristiti najjaci moguci dostupan algoritam
2. Za svaku lozinku koristiti salt koji je potrebno generisati pomocu CSPRNG
3. Koristiti pepper
4. Koristiti veliki broj iteracija algoritma

**SLAJD 19:**

I za kraj, neke generalne preporuke koje se ticu kriptografije generalno.

Ono sto je definitivno lose i sto ne bi trebalo raditi jeste koristiti eksperimentalnu kriptografiju, tj da sami izmisljamo svoj algoritam. Takodje, velika je greska pretpostaviti da ce kriptografski algoritam uvek biti dovooljno sifguran i jak.

Cinjenice su da se u svaki sistem moze provaliti ukoliko na raspolaganju ima dovoljno vremena. Racunari na trzistu konstantno postaju sve mocniji dok im cena opada. Uporedo sa tim, kriptografski napadi postaju sve ozbiljniji i takoreci pametniji.

Zbog pomenutog, potrebno je uvek voditi se tim tj raditi sa pretpostavkom da napadac ima sav nas izvnorni kod. Jedan od principa koji uvek treba imati na umu je Kirckohof koji glasi da sistem treba da bude siguran cak iako je sve osim kljuca, podelejeno sa javnoscu.