**Content**

[1. Termin 1 2](#_Toc30687485)

[1. Zakladne vlastnosti bezpecnosti 3](#_Toc30687486)

[2. Elektronicky podpis 4](#_Toc30687487)

[1.2.1. Digitalny podpis 4](#_Toc30687488)

[1.2.2. SSL 5](#_Toc30687489)

[3. Email – Elektronicka posta 5](#_Toc30687490)

1. Termin 1
   1. Zakladne vlastnosti bezpecnosti

* **Utajenost** (**Confidentiality**)

**Data** **confidentiality** : Assures that private or confidential information is not made available or disclosed to unauthorized individuals. **Privacy** : Assures that individuals control or influence what Information related to them may be collected and stored and by whom and to whom that information may be disclosed. Mechanizmus na zabránenie prístupu k informáciám neautorizovaným osobám, ktoré nemajú na to dostatocne pravá.

Nastroje pre zabezpecenia utajenia:

* + Sifrovanie
  + Kontrola Pristupu
  + Autentifikacia
  + Autorizacia
  + Fyzicka kontrola pristupu
* **Integrita** (**Integrity**)

**Data** **integrity** : Assures that information and programs are changed only in a specified and authorized manner. **System** **integrity** : Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system. Mechanizmus na to, aby v komunikácií nedošlo k nejakej modifikácií informácií, èi už nejakými útoèníkmi alebo nejakým iným spôsobom

Nastroje pre zabezpecenia integrity

* + Zalohovanie
  + Kontrolne Sumy
* **Dostupnost** (**Availability**)

Assures that systems work promptly and service is not denied to authorized users.

Nastroje:

* + Redundancie
  + Fyzicka ochrana
* **Istota**, **Dovernost** (**Accountability** ?!)

Manažment dôveryhodnosti (dôvernost medzi systémom a používatelom) je na takom stupni, že systém a používatel si dôverujú navzájom.

* **Autentickost (Authenticity)**

Stanovenie, že údaje, postupy a práva vydané osobami alebo systémom sú pravé

Under Integrity. Je to stanovenie, že **údaje**, postupy a práva **vydané** **osobami** **alebo** **systémom** **sú** **pravé**. Hlavným nástrojom na zaistenie autentickosti je **digitálny** **podpis**.

* **Anonymita**:

Je to vlastnosť, že určité **záznamy alebo transakcie nepripadnú ku žiadnemu jednotlivcovi.** Nástroje pre zabezpečenie anonymity sú: **Agregácia** – kombinácia dát od viacerých používateľov. **Mixácia** – agregovanie informácií z viacerých strán a spájanie ich do zložiek, ktoré sa nedajú rozložiť. **Proxy** – dôveryhodní agenti, ktorí nahrádzajú skutočnú identitu používateľa. **Pseudonym** – fiktívna identita používateľa, ktorý predstiera identitu.

* 1. Elektronicky podpis

An eSignature refers to the electronic signing process that creates a legally binding agreement, while a digital signature tends to mean the cryptographic signature produced by public/private key signing.

A public/private key digital signature could be one of the signing methods used for eSignature.

The reason a digital signature is not an eSignature is that applying it alone won't create a legally binding agreement, for example, you could apply a digital signature to secure some data for transmission through an untrusted network (in fact this was the original purpose of the public/private key).

An electronic signature is any author identification and verification mechanism used in an electronic system. This could be a scan of your real hand-written signature, or any kind of electronic authenticity stamp. It's a generic term that covers a lot of authenticity measures.

A digital signature is a type of electronic signature.

* + 1. Digitalny podpis

1. Hlavný nástroj na zaistenie autentičnosti
2. analogický ručnému podpisu, ktorý slúži ako dôkaz autorstva, resp. súhlasu s obsahom dokumentu
3. je to určitá dátová štruktúra, ktorá je závislá na dokumente, vzniká hashovaním tohto dokumentu a tento kód je zašifrovaný súkromným kľúčom, ktorý je jednoznačným vlastníctvom vlastníka dokumentu
4. digitalnu signaturu
5. na vytvorenie digitalnej signatury sa vyuzije asymetricke sifrovacie algoritmi s verejnym klucom (RSA, DSA) a bezpecne jednocestne kryptovacie algoritmy (kryptograficke hashovacie funkcie MD5)
6. nie je možné dosiahnuť 100% autentičnosť údajov

How it works:

1. Noone cares if somebody reads the document, we just want to be sure that it comes unmodified from given person.
2. Create hash of the document (SHA-256). Even the tiniest change in the document would result in completely different hash value.
3. Now encrypt the hash with the pirvate key and embedd this hash into the document we want to send.
4. Make the public key available (alternatively a website where ones can fetch)
5. To verify: decrypt the hash – if the public key decypts it it means it comes from the person who has the public key. If the SHA-256 hash of the docs is the same as the previously decrypted hash, than noone modified the document.

Certification:

1. Digital certicate issued by a certification authority guarantees sender’s identy.
2. The digital certificate contains a public key along with other information about the sender+ expiration date.
   * 1. Digitalna obalka

Digitálna obálka: chráni symetrický kľúč pomocou public key

1. Bob vytvorí náhodný symetrický kľúč na jedno použitie
2. Zašifruje správu symetrickým kľúčom
3. Zašifruje jednorazový kľúč Aliciným public key
4. Pripojí šifrovaný symetrický kľúč k šifrovanej správe a odošleju Alici
5. Iba Alica je schopná dešifrovať jednorazový kľúč a správu
   * 1. SSL

„SSL" is the name that is most often used to refer to this protocol, but SSL specifically refers to the proprietary protocol designed by Netscape in the mid 90's. "TLS" is an IETF standard that is based on SSL, so I will use TLS in my answer. These days, the odds are that nearly all of your secure connections on the web are really using TLS, not SSL.s

TLS has several capabilities:

* Encrypt your application layer data. (In your case, the application layer protocol is HTTP.)
* Authenticate the server to the client.

Note: I wrote my original answer very hastily, but since then, this has turned into a fairly popular question/answer, so I have expanded it a bit and made it more precise.

**TLS Capabilities**

"SSL" is the name that is most often used to refer to this protocol, but SSL specifically refers to the proprietary protocol designed by Netscape in the mid 90's. "TLS" is an IETF standard that is based on SSL, so I will use TLS in my answer. These days, the odds are that nearly all of your secure connections on the web are really using TLS, not SSL.

TLS has several capabilities:

* Encrypt your application layer data. (In your case, the application layer protocol is HTTP.)
* Authenticate the server to the client.
* Authenticate the client to the server.

#1 and #2 are very common. #3 is less common. You seem to be focusing on #2, so I'll explain that part.

A server authenticates itself to a client using a certificate. A certificate is a blob of data[1] that contains information about a website:

* Domain name
* Public key
* The company that owns it
* When it was issued
* When it expires
* Who issued it
* Etc.

1. You can achieve confidentiality (#1 above) by using the public key included in the certificate to encrypt messages that can only be decrypted by the corresponding private key, which should be stored safely on that server.[2] Let's call this key pair KP1, so that we won't get confused later on. You can also verify that the domain name on the certificate matches the site you're visiting (#2 above).
2. But what if an adversary could modify packets sent to and from the server, and what if that adversary modified the certificate you were presented with and inserted their own public key or changed any other important details? If that happened, the adversary could intercept and modify any messages that you thought were securely encrypted.
3. To prevent this very attack, the certificate is cryptographically signed by somebody else's private key in such a way that the signature can be verified by anybody who has the corresponding public key. Let's call this key pair KP2, to make it clear that these are not the same keys that the server is using
4. Oversimplifying a bit, a certificate authority creates KP2, and they sell the service of using their private key to sign certificates for other organizations..
   1. Email – Elektronicka posta

Heathers;MUA;POP/IMAP;SMTPMTA;SPAM;DNS;MX-record

Every mta adds a line to the heather.

Spoofing – changing the heathers

Private (Outlook, Thunderbird) and Public (gmail) email system

1. pouzivatel A napise spravu pomocou programu MUA (mail user agent)
2. zvoli e-mailovu adresu pouzivatela B,
3. odosle a MUA pouzije SMTP (simple mail transfer protocol) na to, aby poslal spravu MTA (mail transfer agentovi), kt prevadzkuje ISP (internet service provider) pouzivatela A
4. MTA sa pozre na cielovu adresu, vyhlada nazov domeny v DNS aby zisti mail exchange servery pre danu domenu
5. DNS server odpovie MX zaznamom, kde uvedie mail exchange servre pre danu domenu, v tomto pripade ISP server pouzivatela B
6. MTA odosle spravu pouzitim SMTP protokolu, ktory ho doruci do webschranky pouzivatela B
7. pouzivatel B skontroluje novu postu v jeho MUA, ktory vyzdvihne postu pomocou protokolu POP/IMAP
8. Termin 2
   1. DNS and DNSSEC
      1. DNS

* domain name system
* DNS je system na spravu domenovych mien a ip adries
* protokol aplikacnej vrstvy, ktory preklada domenoveho mena na ip adresu
* uklada: Address (A), Mail Exchange (MX), Name Server (NS)
* prostrednik medzi potrebami cloveka a softveru
* poskytuje mechanizmus získania IP adresy pre každé meno stroja (lookup) a naopak (reverse)
* uvádza poštové servery (MX záznam) akceptujúce poštu pre danú doménu
  + 1. DNS SEC
* umožňujú zabezpečiť informácie poskytované DNS systémom v IP sieťach
* rozšírenie, ktoré klientom DNS umožňuje overenie pôvodu dát a ich integrity
* nezaisťuje však zašifrovanie prenášaných dát a nezaručuje ich dostupnosť
* používa asymetrické šifrovanie (1 kľúč pre zašifrovanie a 1 kľúč pre dešifrovanie)
  1. XSS
  2. Symetricke a Asymetricke Sifrovanie
     1. Symetricke Sifrovanie

- sifrovanie a desifrovanie sa deje len pomocou jedneho kluca

- sila algoritmu zavisi od dlzky kluca a schopnosti uchovat kluc na oboch stranach vbezpeci

- vyhodou je nizka vypoctova narocnost

- priklad DES, AES

* + 1. Hashovacie funkcie

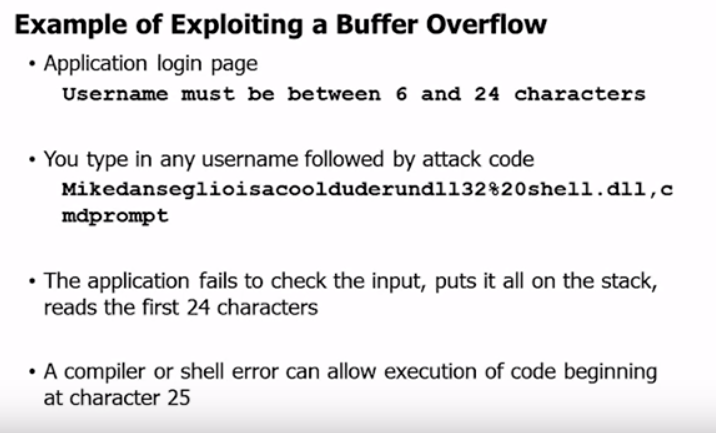
- hashovacia funkcia sluzi na kontrolu integrity dat

- napriklad ci zasifrovana sprava a odsifrovana sprava su totzne

- priklad MD5

- hashovacia funkcia vracia sekvenciu znakov ktore sa navzajom porovnavaju

* 1. Pretecenie Zasobnika – Buffer Overflow

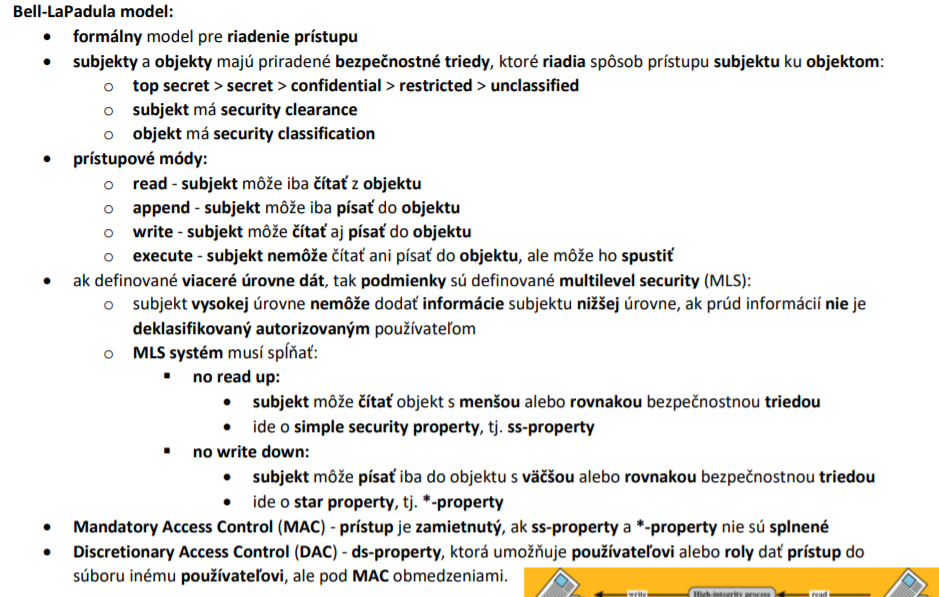


A situation where we're using some, probably low-level C function or something to write a string or some other variable - into a piece of memory that is only a certain length. But we're trying to write something in that's longer than that and it then overwrites the later memory addresses, and that can cause all kinds of problems.

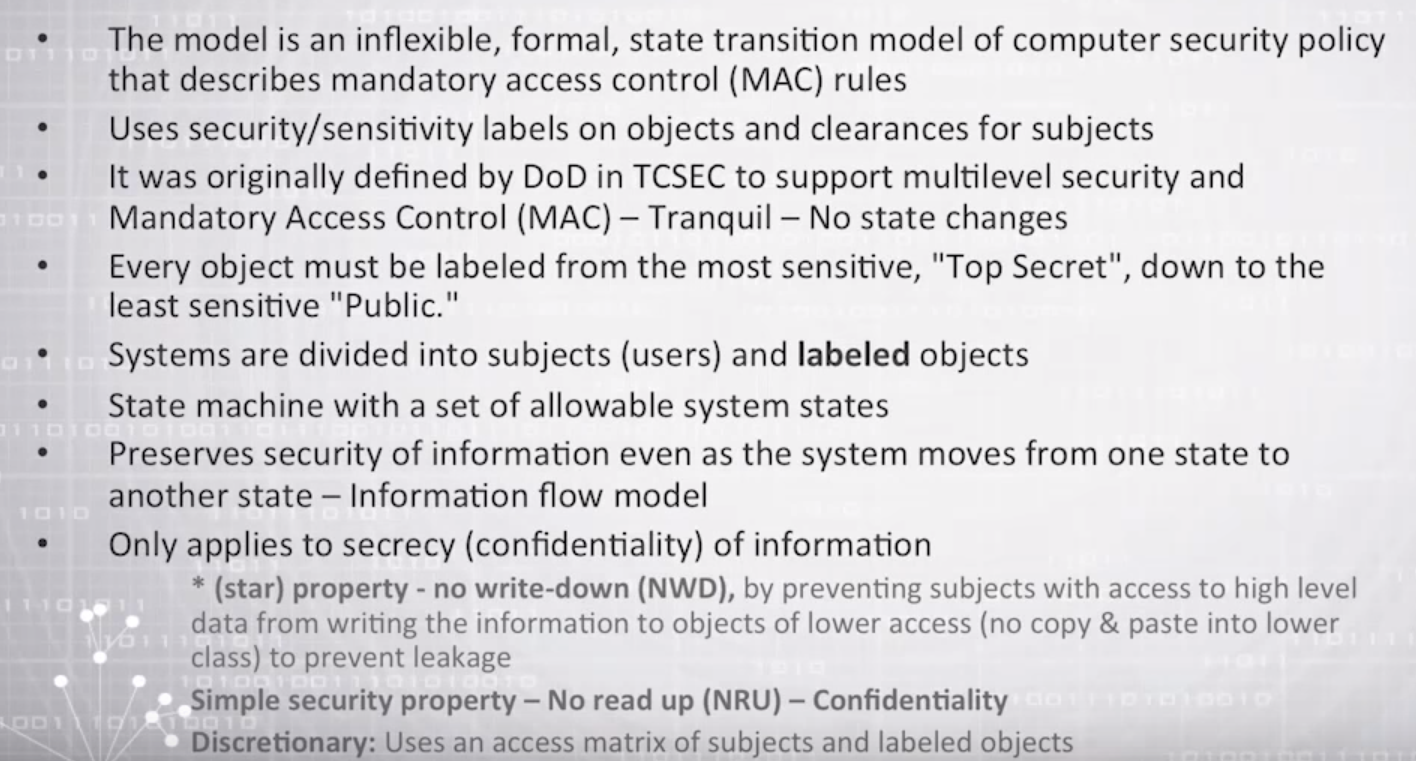
* 1. Security Modules
     1. DAC vs MAC

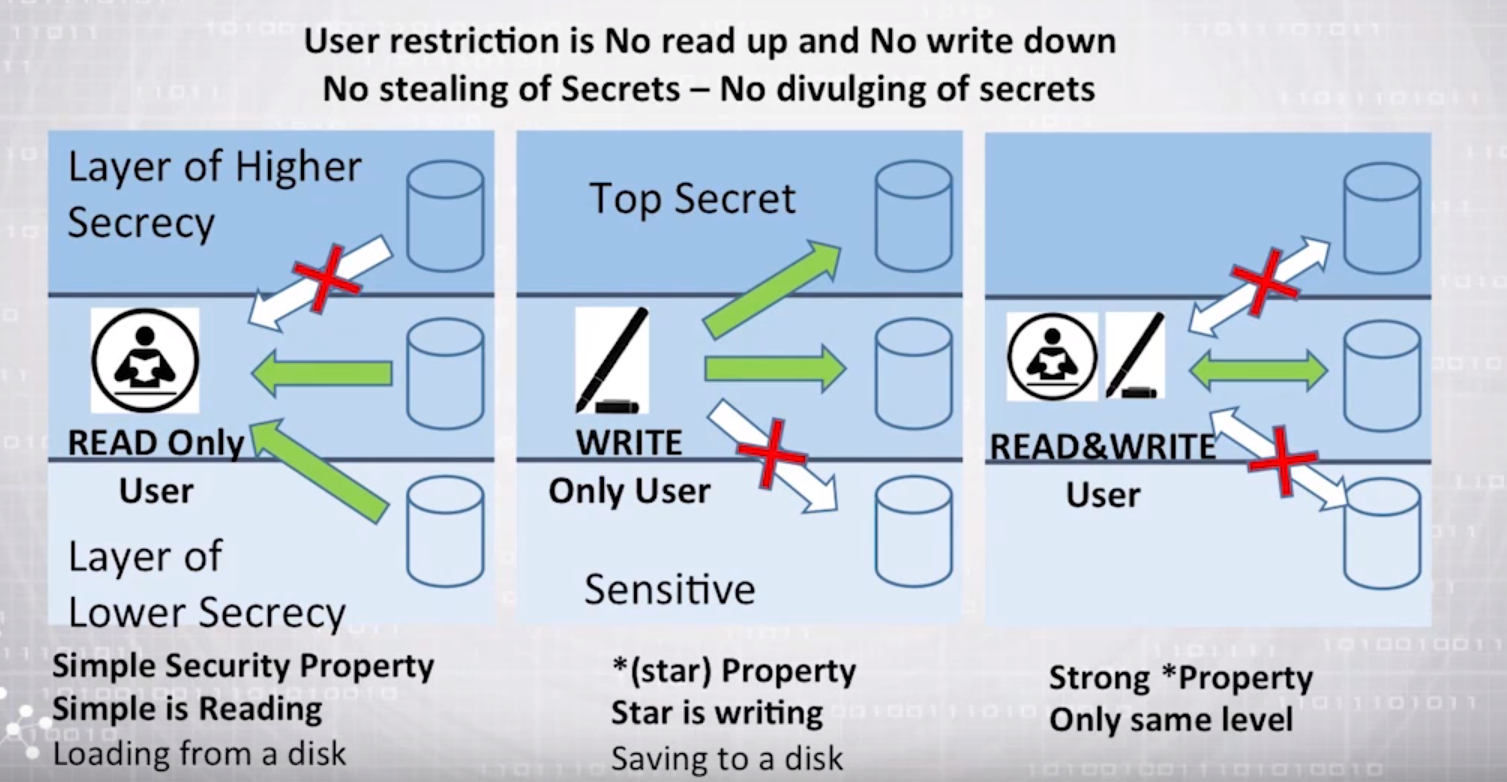
In this regard, Mandatory Access Control (MAC) and Discretionary Access Control (DAC) are two of the popular access control models in use. The main difference between them is in how they provide access to users. With MAC, admins creates a set of levels and each user is linked with a specific access level. He can access all the resources that are not greater than his access level. In contrast, each resource in DAC has a list of users who can access it. DAC provides access by identity of the user and not by permission level.

* + 1. Bella-La Padula

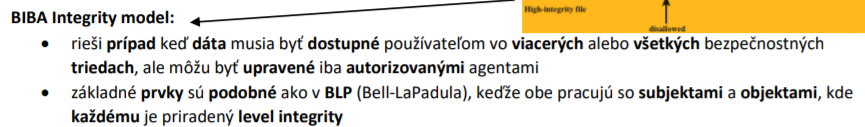


Confidentiality model, that’s concerning with keeping confidentiality data secure. Inflexible. Security Labels.

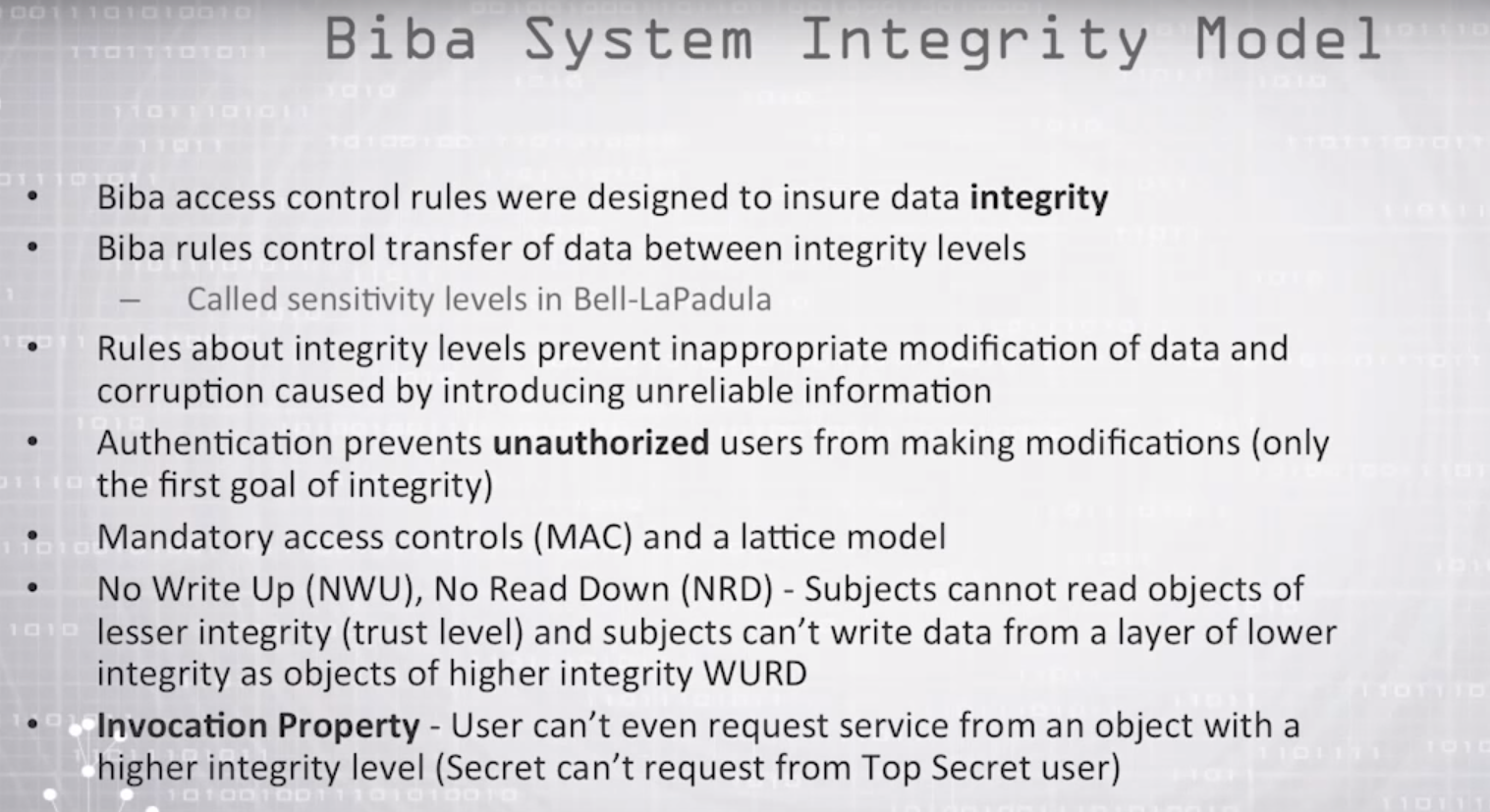




* + 1. Biba



* In general the model was developed to address integrity as the core principle, which is the direct inverse of the Bell–LaPadula model.
* In general, preservation of data integrity has three goals:
  + Prevent data modification by unauthorized parties
  + Prevent unauthorized data modification by authorized parties
  + Maintain internal and external consistency (i.e. data reflects the real world)
* This security model is directed toward data integrity (rather than confidentiality) and is characterized by the phrase: "read up, write down". This is in contrast to the Bell-LaPadula model which is characterized by the phrase "read down, write up".
* The Biba model defines a set of security rules, the first two of which are similar to the Bell–LaPadula model. These first two rules are the reverse of the Bell–LaPadula rules:
  + The Simple Integrity Property states that a subject at a given level of integrity must not read data at a lower integrity level (no read down).
  + The \* (star) Integrity Property states that a subject at a given level of integrity must not write to data at a higher level of integrity (no write up)[2].
  + Invocation Property states that a process from below cannot request higher access; only with subjects at an equal or lower level.

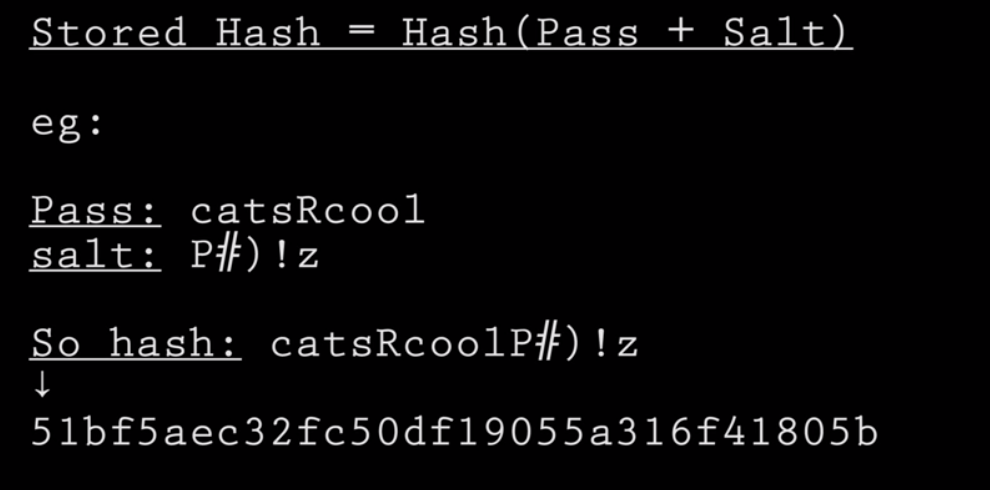


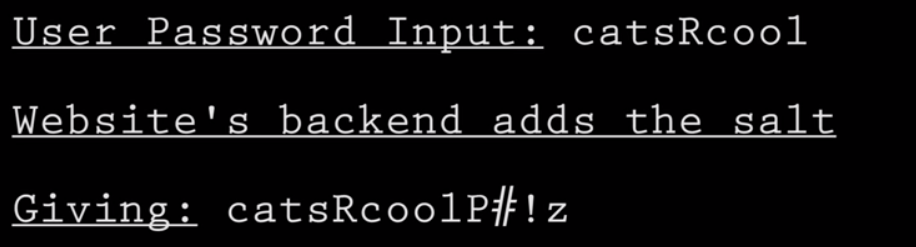
1. Bonus
   1. URL shortener (goo.gl, tinyurl...)
   2. Salting

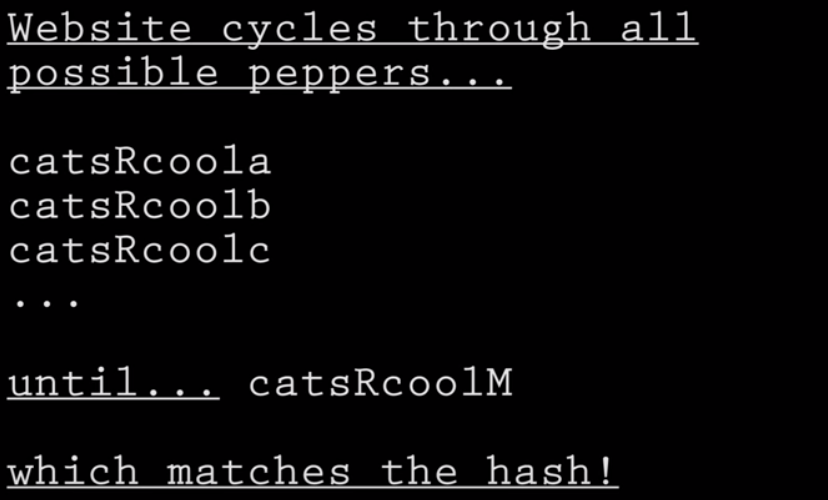
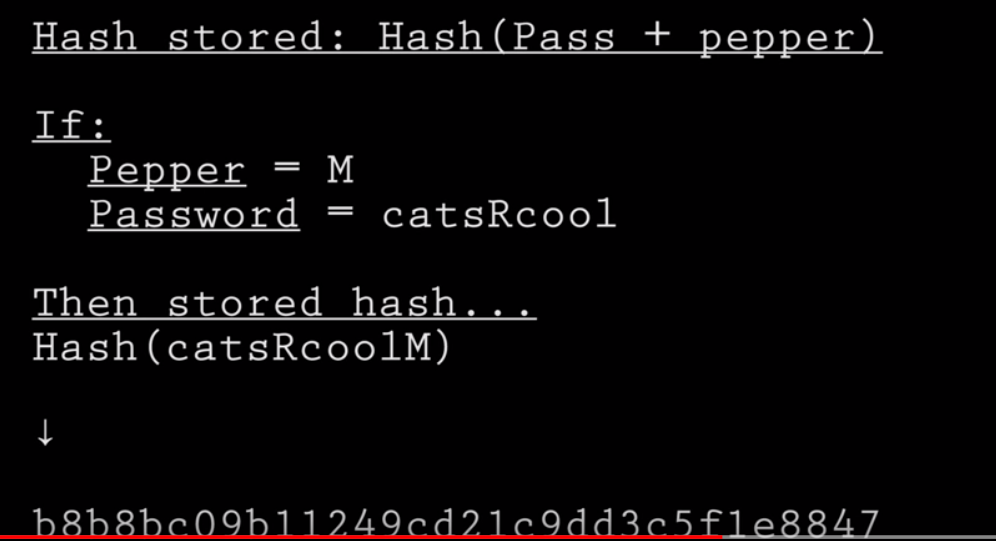
Hashing algo : transforms input data into a simingly random fixed length string.

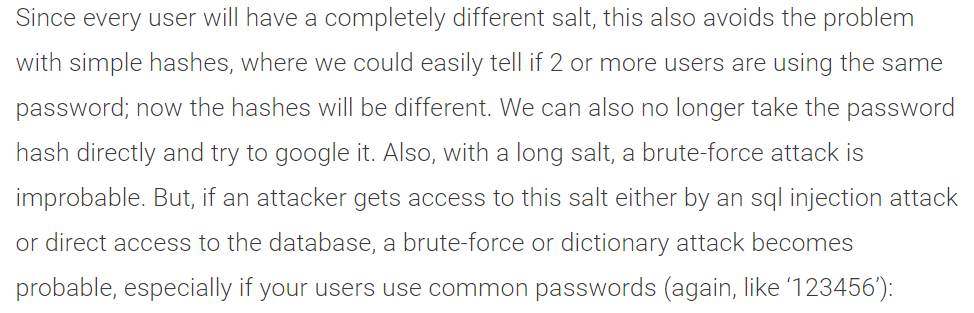
Rainblow table (Dictionary Attack) – precalculated hashes of most common passwords

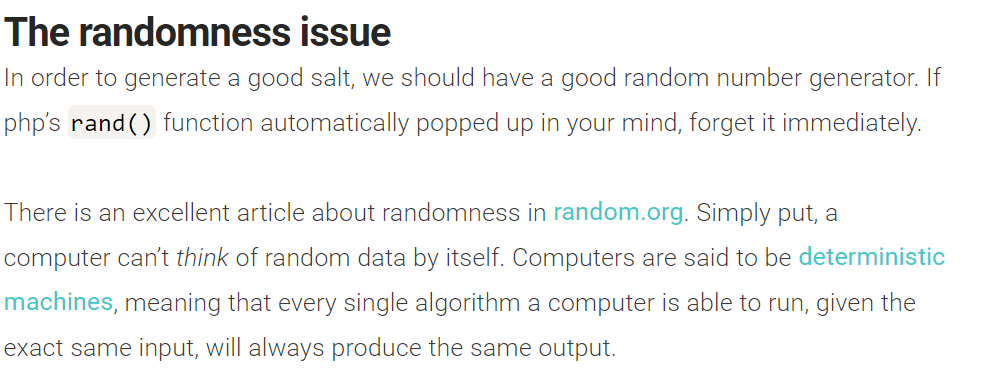
Bruteforce attack – trying every single combination. The comlexity increases exponentially by increasing the length.











Pass – the standard unix password manager.

* 1. Phishing

