Sicurezza informatica

- Il kernel linux è scritto in c e cpp, linguaggi non memory-safe, e in tal senso vulnerabili
- Trait model (modello di rischio) di un sistema informatico
- Malware: any software designed with malicious intent to disrupt normal function, gather sensitive information, access other connected systems when it is installed and run on a device. The user downloads and installs the malware. The typical way to protect devices from malware is with antivirus software (AV)
- Exploit (leggi expluá): script, worm or binary which exploits a vulnerability to gain system administrator's privileges (root). The user wasn’t trying to install any code.
- Worm: virus che di diffonde in maniera semiautonoma
- Exploitation. To be exploited
- Vulnerability: buffer overflow, race condition, heap overflow, double free, integer overflow, SQL injection, or any software defect which can open a breach in the security of the system
- SQL injection: the act of exploiting a SQL database by a malformed query. For example, typing a special strings in the search input field of a web site, a user could get (or change) information pertaining to other users, or produce an undesired content to be rendered in the public web page. SQL injection is a code injection technique that might destroy your database. SQL injection is one of the most common web hacking techniques. SQL injection is the placement of malicious code in SQL statements, via web page input. SQL injection is an instance of a more general class of vulnerabilities based on poor input validation and bad design that can occur whenever one programming or scripting language is embedded inside another.
- Race condition: a class of error in environments that are multi-threaded or otherwise multi-tasking, where an operation is falsely assumed to be atomic. That is, if two operations overlap instead of being done sequentially, there is some risk of the resulting computation not being correct. There are many cases where such a condition can be security critical
- DRM (Digital Right Management): the science of protecting copyright on applications and digital media contents. DRM must block pirates, users who obtained the application illegally
- Code Signing: a DRM mechanism to authenticate a binary, for being executed only on a specific device or any other purposes
- Apple DRM policy is based on several means: App Store review process, code signing, sandbox, SIP
- The app runs in a sandbox at a low privilege level to reduce damage they can cause

- Apple introduced the app Code Signing with the App Store. A section of the Mach-O binary is encrypted. Apple or developers can sign applications using their private key.
- Apple developers can sign an app using the FairPlay protocol and provisioning profiles
- A hash function is any function that maps data of arbitrary size (the pre image) to data of fixed size (image, hash, or digest).
- Una variazione di 1 bit del file di ingresso deve portare al cambiamento di almeno il 50% dei bit dell'hash
- cryptographic hash functions. Una funzione di hashing per uso crittografico deve verificare tre condizioni:
- A hash function is *collision-resistant* if an adversary can't find any collision, i.e two different inputs (pre-images) which result in the same output
- A hash function is *pre-image resistant* if, given an output (image), an adversary can't find any input (pre-image) which results in that output.
- A hash function is *second-pre-image resistant* if, given *one* pre-image, an adversary can't find any *other pre*-image which results in the same image.
- SHA-1 (Secure Hash Algorithm 1) è un esempio di hash function (160bit), introdotto nel 1995. A feb 2017 Google ha effettuato un collision attack, ottenendo una collisione per SHA-1. SHA-1 da deprecato passa a obsoleto. Adesso si preferisce usare SHA-256 (256bit) o SHA-3
- Git uses SHA-1
- Types of attack to hash: collision, pre-image, second pre-image
- Two types of digital signatures: hash-based and asymmetric-crypto-based signatures
- Digital signature: Data that proves that a document (or other piece of data) was not modified since being processed by a particular entity. Generally, what this really means is that anyone who has the right public key can demonstrated which private key was used to sign the data.
- Firma digitale: Il mittente e autore del documento calcola un hash su tutto il file, p.es con SHA-2, e poi, con una crittografia asimmetrica, p.es RSA, firmo l'hash usando la mia chiave privata. Il ricevente rimuove la sezione del file contenente la firma digitale, calcola l'hash SHA-2, poi usa la chiave pubblica per decifrare la firma, e infine confronta i due hash. Se sono uguali allora il documento è autentico
- Talvolta si usa applicare la funzione di hash ricorsivamente più volte su se stesso, in maniera da renderlo più resistente ad attacchi brute-force
- Crittografia + hash: avendo un messaggio in chiaro, diciamo un sw, cifrarlo e basta non é il metodo più sicuro. Infatti, si potrebbe trovare un modo di modificare il messaggio cifrato in maniera tale che, una volta decifrato, il risultato è ancora valido, nel nostro esempio il sw può girare, con magari delle funzioni attive normalmente disabilitate. Infatti per abilitare una funzione anche complessa di un sw basta cambiare un bit. La soluzione è aggiungere al messaggio l'hash del messaggio stesso prima che il messaggio venga cifrato, e poi fare un semplice controllo sull'hash
- Usare un hash che ammette collisioni è insicuro

- TLS e HTTPS sono protocolli che si basano sulla robustezza della firma digitale (delle cosiddette certificate authorities)
- Su sistemi Apple libdyld.dylib (dynamic loader) é la libreria di sistema che ha il compito di decriptare a runtime il programma firmato e caricarlo in una area di memoria controllata dove esso verrà eseguito
- La crittografia in questione è asimmetrica, basata sulla aquisizione e mantenimento del certificato Fairplay sul proprio dispositivo
- Su mac il sistema Gatekeeper filtra le app prima di eseguirle in base all'origine (AppStore, anche da developers, oppure tutte). Su iOS il sistema non è bypassabile
- La firma, se non applicata da Apple stessa, deve essere applicata dallo sviluppatore mediante il provisioning profile
- Self modifying code: tecnica usata da molti malware
- Shell code: codice binario eseguibile che viene caricato in Ram a runtime. É uno dei sistemi adottati dai malware
- ROP: Return Oriented Programming, exploit che permette ad un attaccante di eseguire codice anche in presenza di difese. Siccome il sistema impedisce l'esecuzione di codice che venga caricato a runtime in aree di memoria diverse da quelle pre assegnate (ad esempio tramite code injection da parte di qualche sorgente esterna), si cerca di far eseguire al sistema dei comandi che sono già stati caricati in RAM
- ASLR (Address Space Layout Randomization): misura di protezione contro buffer overrun e exploit che consiste nel rendere (parzialmente) casuale l'indirizzo delle funzioni di libreria e delle aree di memoria. Molto utile contro ROP
- Il code signing marca le pagine di memoria dove il codice puó girare.
- architetture iphone: armv6, armv7, armv7s
- UPnP: universal Plug and Play. Alcune app domestiche devono poter essere accessibili da internet. La porta x del router domestico può essere dirottata verso un ip interno.
- La distribuzione di app al livello di enterprise può essere fatta mediante sito web
- Ogni app ha un URL associato. Una app può aprirne un'altra usando il suo URL
- Un account sviluppatore può installare app su 100 device differenti. Nel suo provisioning profile egli registra i device e poi da Xcode genera le app in formato .ipa da scaricare nei device, tramite iTunes o allegato email.
- AES (Advanced Encryption Standard) is a fast general-purpose block cipher standardised by NIST (the National Institute of Standards and Technology)
- AES-128 and AES-256 are examples of symmetric cryptography
- RSA: Asymmetric algorithm for encryption and digital signatures invented by Ron Rivest, Adi Shamir and Leonard Adleman, based on prime number factorisation. There are two prime numbers p1, p2 (private key) which give the co-prime p1*p2 (public key). RSA-4096, where p1 and p2 are 2048 bit numbers, is considered very secure
- DSA (Digital Signature Algorithm): digital signing based on discrete

logarithm inverse

- ECDSA (Elliptic Curve DSA): digital signing based on elliptic curves. We start from a origin point on the curve defined in a p-field (p prime), multiply it by a scalar k (private key) to get another point on the curve (public key). It is able to provide the same security level as RSA with a smaller key and computationally cost. For this reason we use ECDSA for signing on mobile devices. ECDSA-256 is considered secure
- To create a new ssh-key you use the command *ssh-keygen -t <type> -b <size>*, where *<type>* is either of *dsa*, *rsa* and *ecdsa*, and *<size>* is the size in bits of the key
- Authentication: the process of verifying that someone or something is the actual entity that they claim to be. Authentication is what happens when you log into a system. It compares your credentials (often user name and password) with a previously established known value such that the system can know that you are who you say you are. For sensitive systems, there is a trend toward using two factor authentication (2FA) which essentially means that users must supply two different secrets, usually one is a password (something they know) and the other is a pin supplied via text (verifying something they have).
- Backdoor: malicious code inserted into a program for the purposes of providing the author covert access to machines running the program
- Certificate: a data object that binds information about a person or some other entity to a public key. The binding is generally done using a digital signature from a trusted third party (a certification authority)
- SSL (Secure Socket Layer): a popular protocol for establishing secure channels over a reliable transport. This protocol has evolved into the TLS protocol, but the term SSL is often used to generically refer to both
- TLS (Transport Layer Security) is the successor to the SSL protocol. In addition to encrypting data over the wire, TLS authenticates a server with a certificate to prevent spoofing
- Spoofing: it is the act of providing a false identity to a network client. It can be conducted at any one of the ISO/OSI layer
- ARP Spoofing: tecnica utilizzabile da un attaccante che prevede l'invio di messaggi ARP su una rete locale generalmente al fine di associare il proprio indirizzo MAC all'IP di un altro host (ad esempio il gateway predefinito) intercettando quindi tutto il traffico ad egli destinato.
- CFI (Control Flow Integrity): used to prevent ROP like attacks, where chain of gadgets (chunck) of code. Prendiamo il sorgente di tutto il programma e genero il grafo diretto in fase di compilazione e creo dei controlli in forward edge e in backward edge (ritorno). Source e target sono i punti di partenza e di arrivo. CFI controlla che il programma salti sa un source a uno dei target validi per quel source. Purtroppo non sempre si ha tutto il sorgente, ma ci sono librerie, framework, ecc prese da terze parti e non protette da CFI. Un altro sistema è basato su firme. Posso saltare a tutte le funz che hanno una certa firma. Posso applicare il controllo anche in backward. Le prossime generazioni ARM includeranno registri dedicati per CFI backward

edge, il return address verrà cifrato in forward e decifrato in backward. Purtroppo i compilatori JIT (come quello del Java) tipicamente non implementano CFI
- Data Only Attack: attraverso particllari dati si forza il programma
- Al livello di sicurezza, i browser sono il punto debole, in quanto devono erigere barriere contro vulnerabilità di vati media filesm, protocolli di rete, ecc

- SIP

- SIP (System Integrity Protection): also known as "rootless", it is a access control system where permissions change from user based to process based. SIP is intended to protect system resources integrity, also for the root user. It is a sandbox profile that any processes have to follow. The list of resources can be found in /System/Library/sandbox/ rootless.configuration.
- SIP was introduced with OS X El Capitan and prevents potentially malicious software from modifying protected files and folders on your Mac. It also restricts the root user account and limits the actions that the root user can perform on protected parts of the Mac operating system. SIP grants
- protection of contents and file-system permissions of system files and directories;
- protection of processes against code injection, runtime attachment (like debugging) and DTrace;
- protection against unsigned kernel extensions ("kexts").
- To see which resources are protected use ls -Ol /. For example you can see that /System is a "restricted" folder. You can't change that folder even though using root privileges
- Before SIP, exploiting a system by executing code in kernel space did not add any substantial advantage over taking the root privileges. After SIP it makes sense
- To disable SIP you need to reboot the Mac in Recovery Mode (by holding down the Command+R keys at starts up). Enter *csrutil disable* into the Terminal window and hit the *Return* key to disable SIP. Then restart. Use *csrutil enable* command in Terminal to re-enable System Integrity Protection. SIP can be also easily disabled from kernel space
- Modificare binari di sistema, installare daemon in certe locazioni di memoria, ecc sono tutte azioni particolari che il kernel consente o meno a un processo sulla base di due fatti: chi ha firmato il processo, gli entitlements dichiarati per quel processo al momento di firmarlo. P.es accedere ai nostro contatto, ottenere la lista dei processi attivi, ecc sono tutti entitlements
- TrustedBSD e MAC F.
- xss (Cross Site Scripting): temporaneo o persistente