Security on The Web

Take care of yourself!

Case studies from the real world

"

Case study: brute force password cracking

- A service is accessed by user/password
- In someway, service users are gathered:
  - online address books, social engineering, a leak on service itself
- A botnet of hosts try to guess weak passwords by using a dictionary of weak passwords:
  - hosts may be thousands and spread world-wide.
  - behaviour may be aggressive (thousand of guess a day for host) or stealth (5-6 tentatives/days for host).
- Some example of weak (used!) passwords:
  - 12345678, taylorswift, ronaldo7.

Case study: Heartbleed

- On 2014, It was discovered a vulnerability leak on the implementation of some version of TSL/SSL
- Thanks to this vulnerability, an attacker could dump the memory of the server, including the private key used by the server to grant itself on the client and encrypt connections.
- The vulnerability could be mitigated by some settings and it has been resolved in few weeks by updates.

https://it.wikipedia.org/wiki/Heartbleed

Case study: XcodeGhost

- XcodeGhost (and variant XcodeGhost S) are modified versions of Apple's Xcode development environment that used to produce modified version of apps for IOS containing a maliciouse library.
- Apps compiled with XcodeGhost can be remotely controlled.

https://en.wikipedia.org/wiki/XcodeGhost

Case study: Sony SQL Injection

- In 2011, the database of user of PlayStation Network has tumpered by using a tecnique called SQL Injection.
- By send a crafted user and password credential attacker has gained the access to the whole database containing the profile of all users

https://www.youtube.com/watch?v=cMMzC2WlJA0

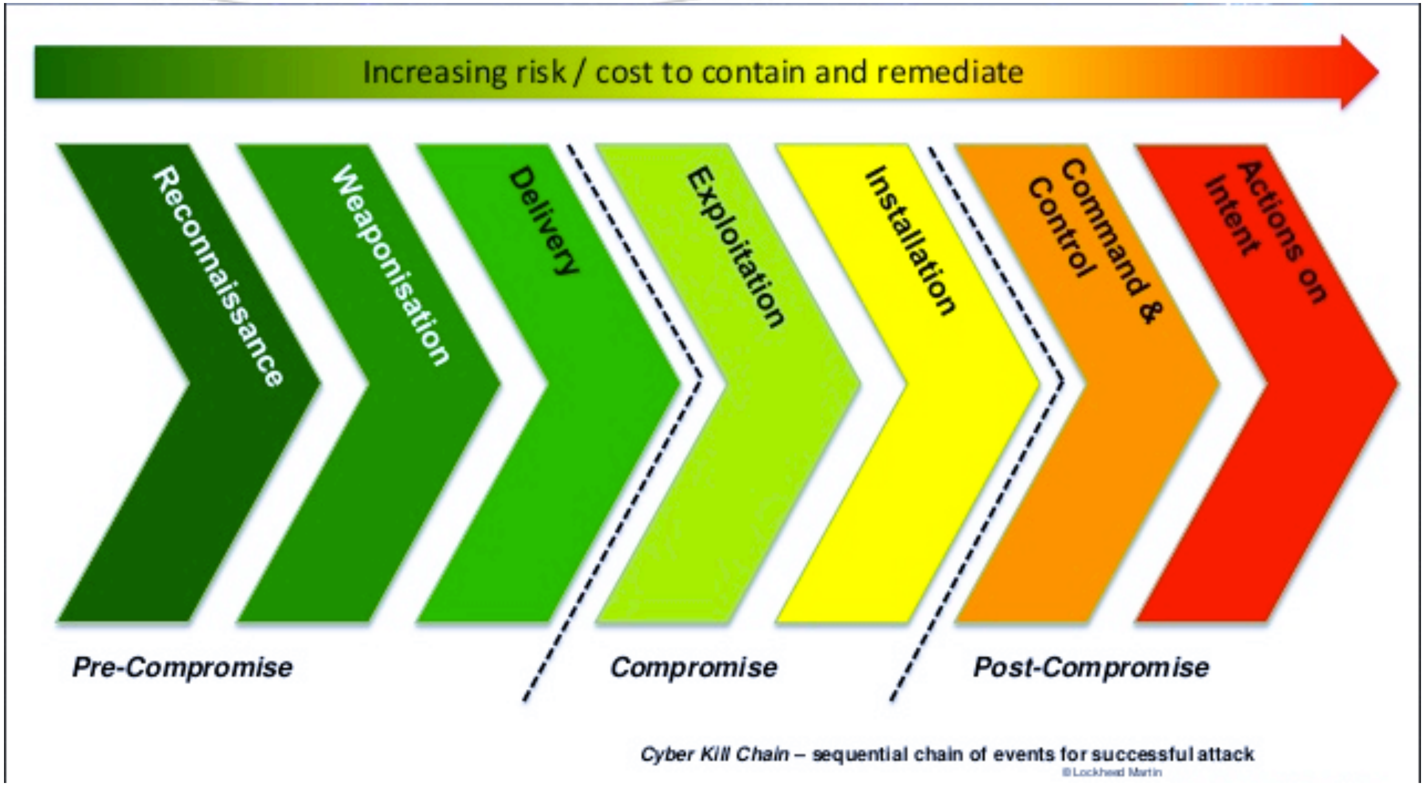
AOL search data leak

- In August 2006, AOL release for research a set of anonymized search log made by users.
- After few days, someone discovered that by make a cross-correlation on data, It was possible to extract the real identity of some users.

- https://en.wikipedia.org/wiki/AOL_search_data_leak

A schematic view

The kill chain



Increasing risk / cost to contain and remediate

Reconnaissance

Weaponisation

Delivery

Exploitation

Installation

Command & Control

Actions on Intent

Pre-Compromise

Compromise

Post-Compromise

Cyber Kill Chain – sequential chain of events for successful attack

©Lockheed Martin

https://www.slideshare.net/Tier3Huntsman/intelligencebased-computer-network-defence-understanding-the-cyber-kill-chain-royal-holloway-university-conference

Case study: the kill chain in action

- Gather all students email by fake "pizza" discounts
- Prepare a docx with a malicious macro that download a maliciouse program (first stage) and encapsulate it in a crafted email: "Mandatory steps for thesis" from a trustable sender, ex. "admin@unimi.it" (mail itself is not a trustable chanel).
- Delivery the malicious email to the targets.
- The maliciouse program download from a maliciouse site a (set of) programs in order to exploit a knowk or 0-day vulnerability for the attacked host (second stage) and install a trojan program
- The trojan program act the maliciouse actions:
  - connect to a botnet for further orders
  - Crypt all and ask for ransom

Targets

- Steal Informations
  - Prototypes, new undiscovered products, unregistered patents
  - Internal sensitive news
  - Accounts and sensitive data.
- Ransom
  - to recover encrypted data
  - to avoid discosure of exfiltered sensitive data
  - to stop a Denial of Service
- Logistic reasons
  - Compromise host in order to build a botnet

Vectors

- Social engineering
  - email/phone phishing in order to steal credentials or data
  - email/chat links to malware
- Credential cracking
  - Brute force cracking
  - Connection tapping/tampering
- Vulnerabity
  - Bug on application or some included plugin
  - Database tampering through backdoors
- Multi-stage attacks:
  - Combination of all above:

Defenses

- Protect sensitive information
  - Phisical protection
  - Audit and strong policies on administrator operations
  - Code audit
- Protect profiles by promote:
  - strong password
  - strong securities protocols
- Protect your application:
  - Updates from trustable sites
  - Write (or pretend from writer) good and clear code!
- Mantain a secure and updated backup of everything.