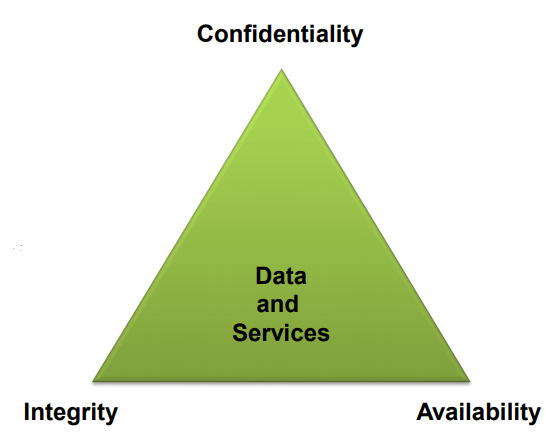
WEEK 2

**Security Triad**



**Confidentiality**

***“Information is not disclosed to any parties other than the intended recipients”***

**Security Mechanisms:**

* Cryptography
* Access Control

**Integrity**

**“*Protection against unauthorised modifications to the data or the system”***

**Security Mechanisms:**

* Cryptography (hash/MAC)
* Error detection, correction codes, checksum
* Establish procedures for system operation, maintenance and administration.

Availability

***“The information and system resources are available to the legitimate users”***

**Security Mechanisms:**

* Redundancy
* Fault tolerance
* Resilient system design
* Attacks against the availability of a system are known as Denial of Service attacks (**DoS**).
* High availability is very expensive.

Authentication

***“Provides the means to verify the identity of an entity”***

* An entity can be a **person**, **computer**, or any **object** **on** the **network**, **including data**.
  + What you know (*passwords)*
  + what you posses (*badges, smart cards)*
  + what you are (*biometrics)*

Non-Repudiation

***“An entity cannot deny of performing some action”***

* **Non-repudiation of origin**
  + A sender **cannot** **deny** that he **sent** a **message**.
* **Non-repudiation of delivery**
  + A receiver **cannot** **deny** that he **received** a **message**.

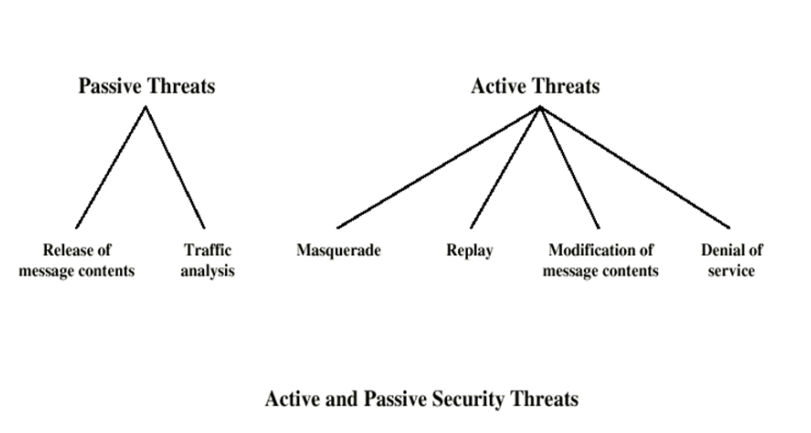
Secure Coding/Programming

* **Practice of developing computer software**
  + That guards against the security vulnerabilities.
* **Defects, bugs and logic flaws are**
  + The primary cause of commonly exploited software vulnerabilities.
* **Most vulnerabilities stem from**
  + A relatively small number of common software programming errors.
* **Identify the insecure coding practices**
  + Educating developers on secure alternatives.

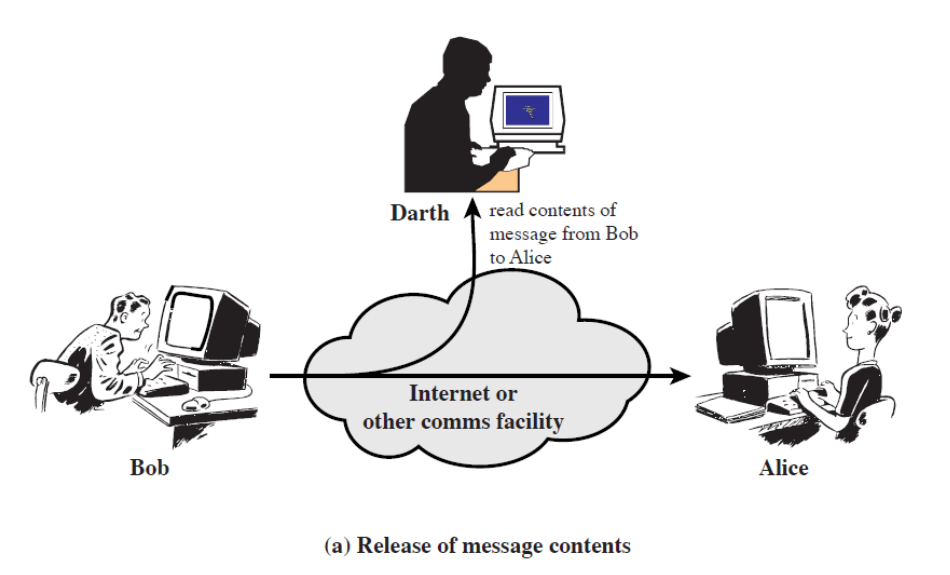
Threat, Vulnerability and Attack

* A **threat** is an **element** which can **potentially** **cause** **harm** or **loss**.
* *Threats can be human or non-human*
* A **vulnerability** is a **weakness** that makes it **possible** for a **threat** to **occur**.
* An **attack** is any **action** by a **threat** **exploiting** a **vulnerability** in order to cause **harm** or **loss**.

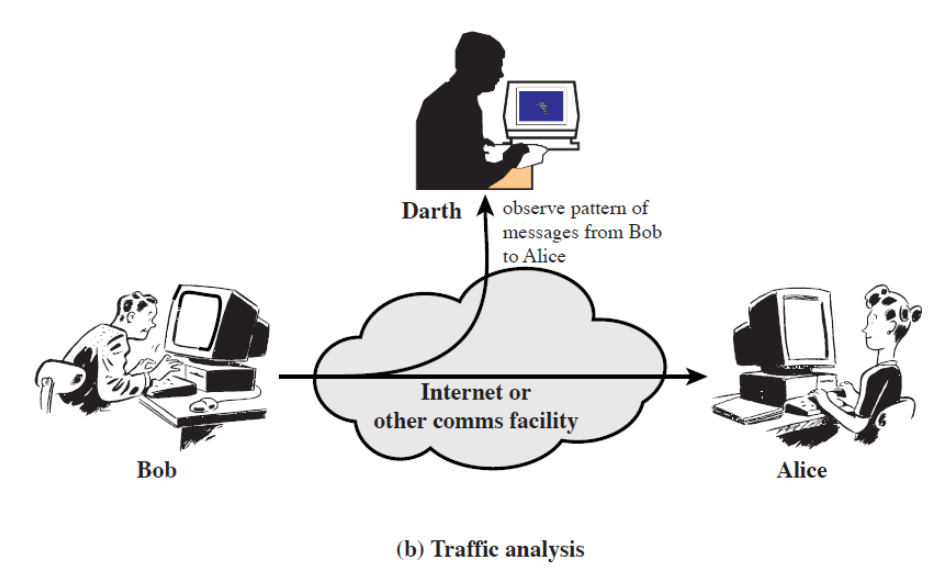
Active and Passive Threats



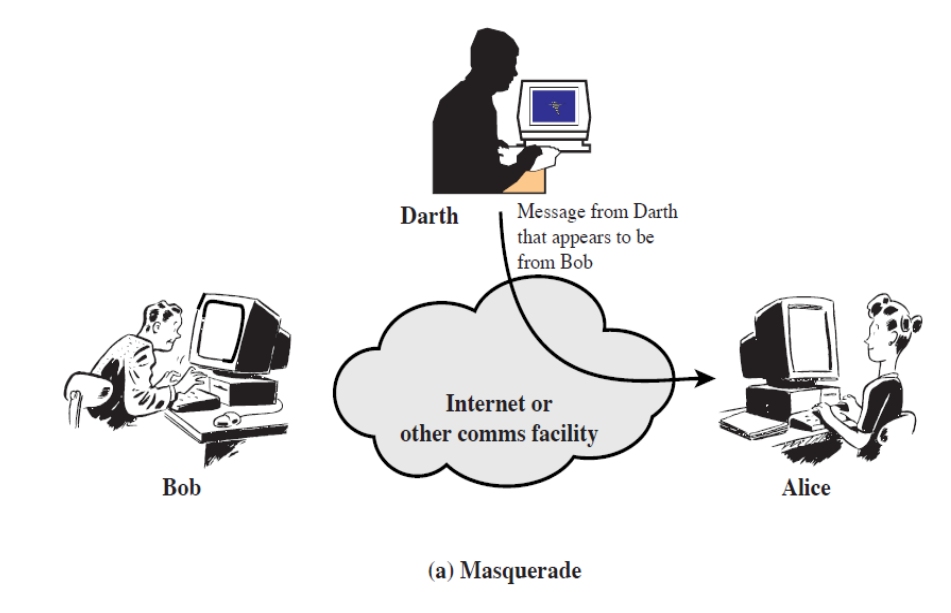
Passive Attack – Eavesdrop



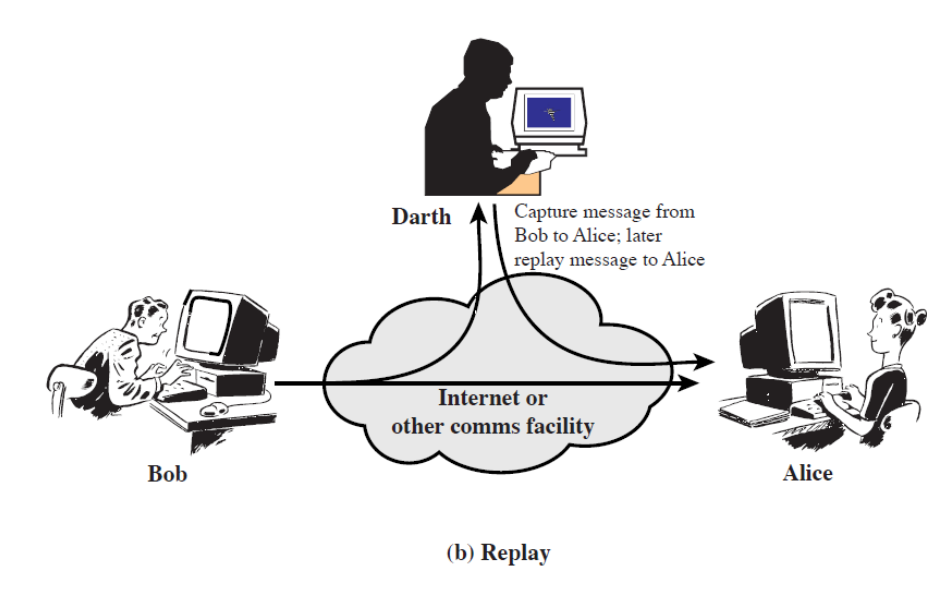
Passive Attack – Analysis



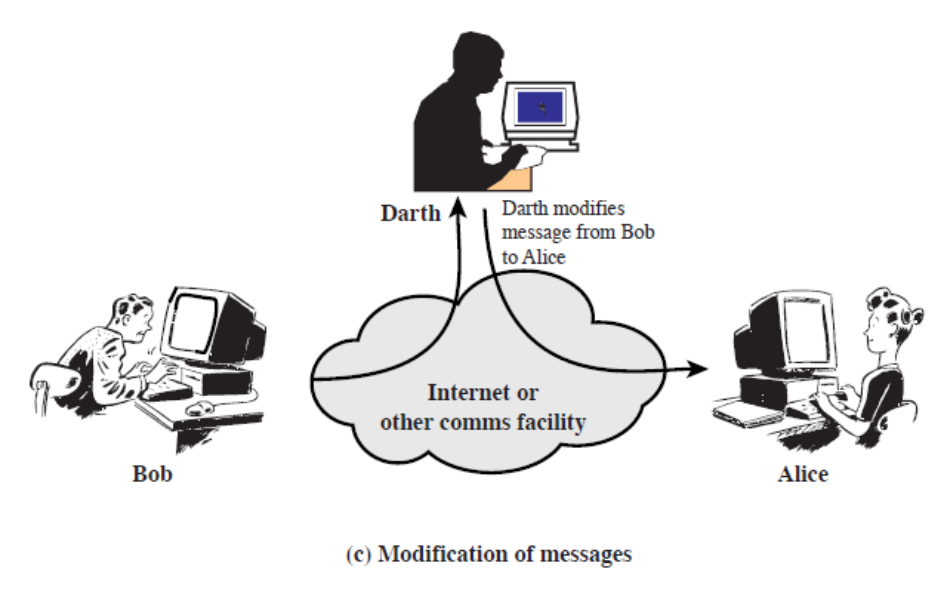
Passive Attack – Impersonation



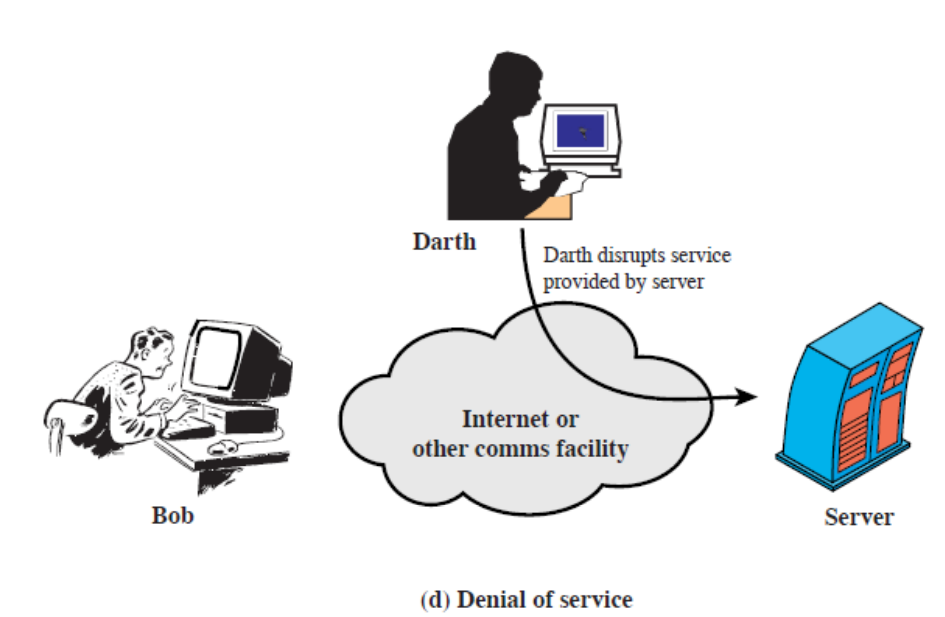
Passive Attack – Replay



Passive Attack – Intercept and Modify

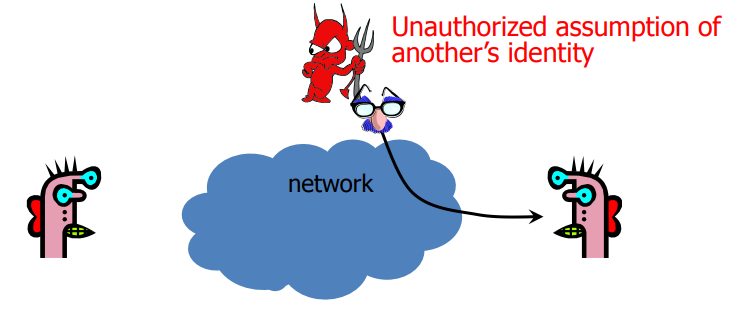


Active – DoS



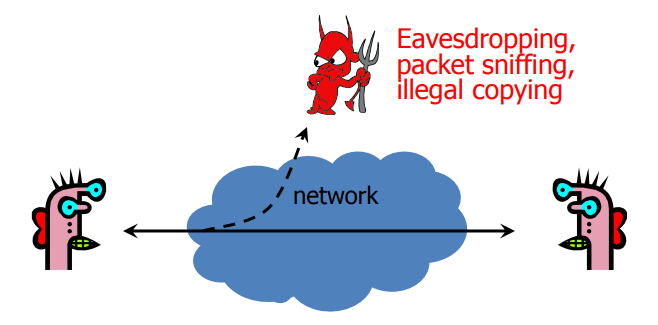
Attack on Authenticity

Authenticity is **identification** and **assurance** of **origin** of **information**.



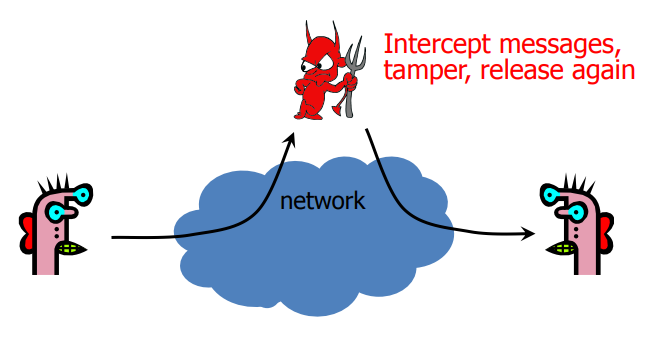
Attack on Confidentiality

Confidentiality is **concealment** **of** **information**.



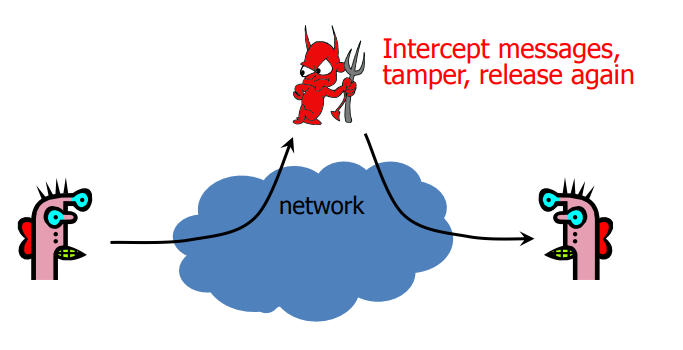
Attack on Integrity

Integrity is **prevention** **of unauthorized changes**.



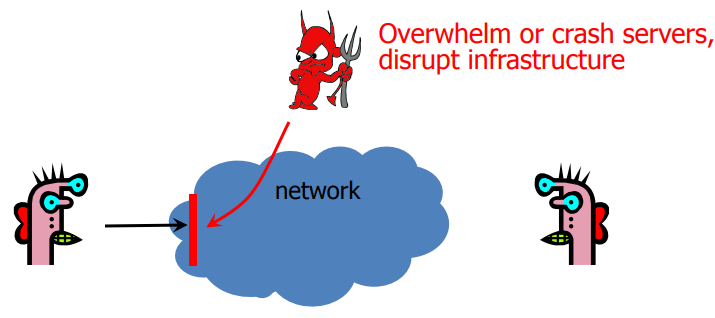
Attack on Integrity

Integrity is **prevention of unauthorized changes**.



Attack on Availability

Availability is **ability to use information or resources desired**.



Security Attacks

* **INTERRPUTION** is an attack on: **Availability**
* **INTERCEPTION** is an attack on: **Confidentiality**
* **MODIFICATION** is an attack on: **Integrity**
* **FABRICATION** is an attack on: **Authenticity**

Sniffing

* It is the **easiest** **attack** to **launch** since **all the packets transit through the attacker**.
* All the “**plaintext**” **protocols** are **compromised** (*the attacker can sniff user and password of many widely used protocol such as telnet, ftp, http*)

Data/Packet Injection

* Possibility to **add** **packets** to an **already** **established** **connection**.
* The attacker can **modify the sequence numbers** and **keep the connection synchronized while injecting packets**.
* If the **MITM attack is a “proxy attack”** it is even easier to inject (*there are two distinct connections*)

Command Injection

* Useful in scenarios where a **one-time authentication** is used.
* In such scenarios, sniffing the password is useless, but **hijacking an already authenticated** **session** is **critical**.
* Injection of **commands to the server**.
* **Emulation** of **fake replies to the client**.

Malicious Code Injection

* Insertion of **malicious** **code** **into web pages** or **mail** (*JavaScript, trojans, virus*).
* Modification **on the fly of binary files** during the **download** **phase** (*virus, backdoor*).

*Parameters Substitution*

* **Parameters** **exchanged** by server and client can be **substituted** in the **beginning** of a **connection** (*algorithms to be used later*).
* Example: the attacker can force the client to initialize a SSH1 connection instead of SSH2.
* The server replies in this way:
* SSH-1.99 -- the server supports ssh1 and ssh2.
* SSH-1.51 -- the server supports ONLY ssh1.
* The attacker **makes a filter to replace “1.99” with “1.51”**

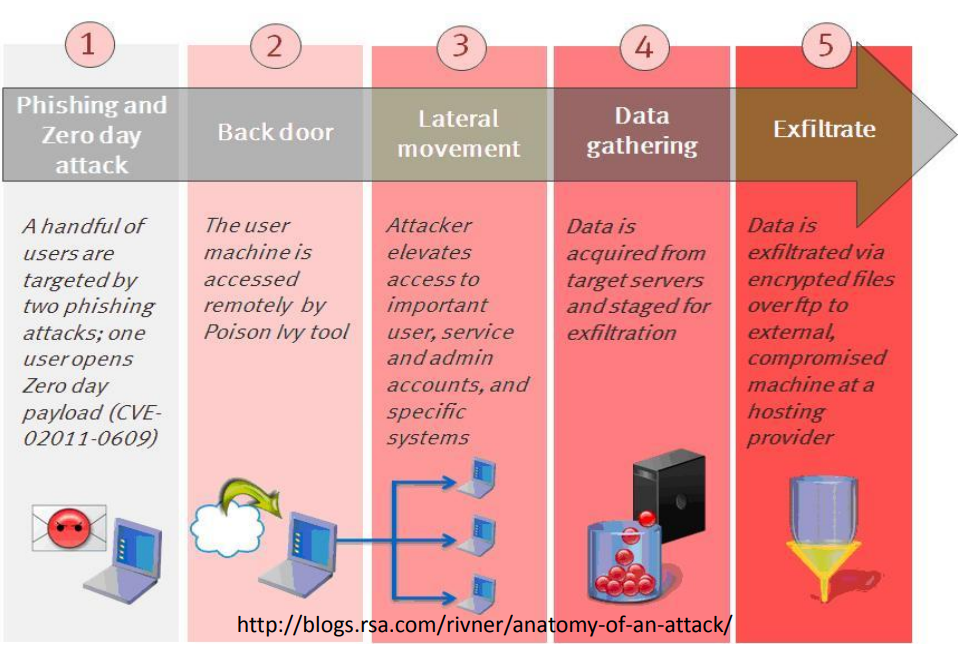
DDoS Attack

**Flooding**

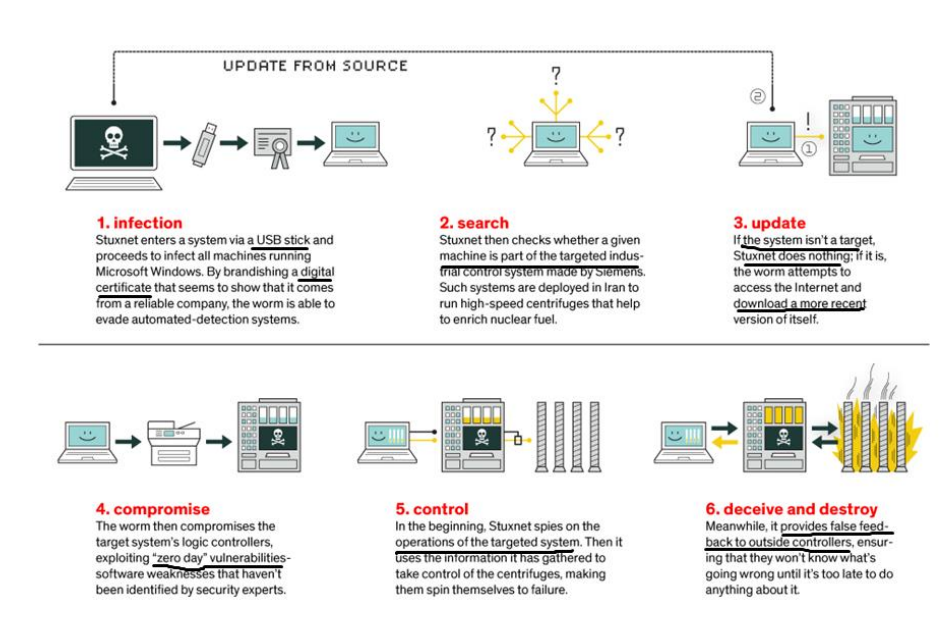
* Attacker sends an **overwhelming** **number of messages to your machine**; great congestion.
* The congestion may occur in the path before your machine.
* Messages from legitimate users are crowded out.
* Usually called **a Denial of Service (DoS) attack**, because that’s the effect.
* Usually involves **a large number of machines**, hence **Distributed** Denial of Service (**DDoS**) attack.

0-Day (Zero Day) Vulnerabilities

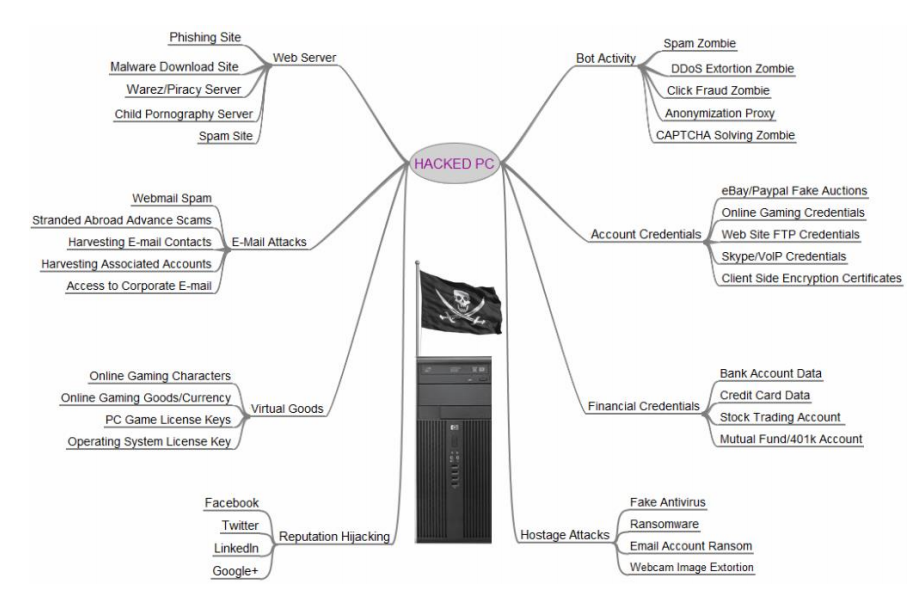
* It is an unknown exploit that **exposes** a **vulnerability** in **software** or **hardware** and can create **complicated** **problems** well **before** **anyone** **realizes** something is wrong.
* In fact, a zero-day exploit leaves **NO opportunity for detection** ... at first.
* The term **‘zero-day’** refers to an **unknown vulnerability** that the **developer** is **newly** **aware o**f, and thus an **official patch** or **update** to fix the issue **has not been released**.

Advanced Persistent Threat (APT) 

Stuxnet



Why Should I Care?



Human Factors/Usable Security

**Usability**

* assesses how easy user interfaces are to use.
* also refers to methods for improving ease-of-use during design process.

Usability is defined by 5 quality components:

1. **Learnability**:

* How easy is it for users to accomplish basic tasks the first time they encounter the design?

1. **Efficiency**:

* Once users have learned the design, how quickly can they perform tasks?

1. **Memorability**:

* When users return to the design after a period of not using it, how easily can they re-establish proficiency?

1. **Errors**:

* How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

1. **Satisfaction**:

* How pleasant is it to use the design?