1. Zero day exploits

A zero day exploit is an attack that is able to exploit an unknown vulnerability in a computing system. It is an exploit that the developers have not had a chance to patch. These exploits are very dangerous and worth a lot of money on the black market, or even to governments. Since the exploit is taking advantage of an unknown vulnerability, they can become a huge weakness for any system that is running the application that is vulnerable. The best example of a zero day exploit is Stuxnet, which utilized several zero day exploits to spread to a specific target and disable it.

<http://www.zdnet.com/blog/security/stuxnet-attackers-used-4-windows-zero-day-exploits/7347>

2. Data breeches

A data breach occurs when a company fails to properly secure their data. This happens when the data is improperly secured and a hacker is able to successfully gain access to the data.

<http://www.nj.com/business/index.ssf/2014/10/kmart_latest_victim_of_major_credit_card_hack.html>

An example of a recent data breach happened a Kmart. Someone was able to gain access to their data and stole payment information that was used at the stores.

There is no real clear solution to prevent a data breach, other than change passwords often, make sure they are complex, and limit outside access to the data. Another way to help prevent a breach is by utilizing a DMZ and encrypting the data.

3. Buffer overflow attack

<http://en.wikipedia.org/wiki/Buffer_overflow>

A buffer overflow attack is directed towards the memory on the system. It intentionally creates a situation that will over write the bounds of the buffer memory, or by copying something from memory without checking to see if it fits in the new memory location. A common attack for this is on a stack, where a user can manipulate the stack in a manner to gain access to the information in the buffer.

<http://theconversation.com/iworm-hack-shows-macs-are-vulnerable-too-32640>

This article highlights how buffer overflows are out in the real world today. Most of the problems are from flawed programming and have been around for a long time (some as long as 23 years!). This article also discusses how Macs are just as vulnerable as PC and Linux users.

There is no real way to prevent a buffer overflow, other than to enforce good programming techniques. A lot of mis-allocated memory in the programs (usually C and C++) and therefore is quite wide spread.

4. Fuzzing

<http://en.wikipedia.org/wiki/Fuzz_testing>

Fuzzing occurs when a software application is “attacked” with varying input. This technique can create unexpected results from the software due to unexpected input. Fuzzing itself is generally used more for stability testing for a software application, but it can easily be used by a malicious user to test the integrity of the application.

<http://www.meddeviceonline.com/doc/how-cybercrime-can-impact-medical-devices-0001>

In this article, it mentions how fuzzing is capable of effecting medical devices. If a device was not properly tested for stability under varying input, then the sky is the limit on what can be done to a device. I would not personally want a pacemaker that someone could wirelessly hack into and crash.

At the end of the article, it mentions that the FDA now has cybersecurity requirements for new devices and that would seem to be the best solution. Another solution is making sure there are no loop holes, and properly securing the device with a strong password.

5. Injection Attacks (SQL, Code)

A sql injection attack is usually aimed at a webserver and occurs when the server does not protect against a string literal escape key. This causes the system to jump out of the input and “read” the next line of code, which the user maliciously inserted to do something with the database. This usually results in the data being pulled to the user who launched the attack.

<http://www.itpro.co.uk/security/23220/four-hackers-accused-of-stealing-military-software>

This article highlights on SQL injection attacks that were used to steal $100m USD worth of software from the military. The group stole user names and passwords and then utilized SQL injection attacks to obtain the software.

The best solution to prevent a SQL injection attack is to clean any user input before running a query with it. Utilizing good programming techniques is part of cleaning user input. Also, passwords once again should be strong and changed frequently to prevent an attack like this from happening.

6. BYOD

<http://www.computerweekly.com/opinion/Security-Think-Tank-ISFs-top-security-threats-for-2014>

Bring your own device is an up and coming security threat for networks of various kinds. This happens when an employer allows employees to bring their own device to connect to the employers network. The problem here is that not everyone secures their computers and not every ones computers are free from malicious software. The threat here is that an infected computer has the capability to infect any other non-protected computers.

<http://www.zdnet.com/five-security-risks-of-moving-data-in-byod-era-7000010665/>

This article over views some of the common risk just with data when you BYOD to a network. The largest threat to the data is unauthorized access via an app or security breach.

A solution for BYOD is to once again enforce strong passwords. Another solution is to use a hierarchy for data access as well as having rules for antivirus programs.

7. Mobile devices

<http://en.wikipedia.org/wiki/Mobile_security>

Mobile smart phones are become more and more dominate in today’s market place. The biggest problem with this from a security standpoint is that these are literally mobile computers that no one bothers to secure. Humans are used to having a cell phone that was not a smart phone and did not require a security program on it. This mentality has moved over to smart phones and is creating a large security gap that a malicious user could exploit.

<http://www.theguardian.com/technology/2014/oct/01/hong-kong-protesters-targeted-by-smartphone-spy-apps-security-company-says>

In this article, it discusses how some of the applications on the market are actually targeted to spy on people. This creates another interesting aspect of mobile phone security sense a 3rd party app could easily compromise a smart phone.

The best solution to secure a smart phone is to get a reputable security app as well as securing your phone with a complex password. Another method is to verify application makers as honest and limit the amount of applications that you put on your phone.

8. USB attacks

<http://www.itbusinessedge.com/slideshows/show.aspx?c=87289&slide=2>

USB’s pose a very serious physical threat to a computer. Since it is possible for a script to execute on the USB whenever it is plugged in (and the user would never know). They are also small, highly mobile, and vary in size. This makes them prone to be picked up by an unknowing user and then plugged into a computer, which it promptly infects. The USB can also “replicate” itself as a keyboard to key log any other key boards on a system.

<http://www.azfamily.com/news/technology/Beware-rogue-USB-devices-279064081.html>

This article highlights a recent USB program called BadUSB. This program is undetectable and is currently not able to be patched. The USB itself can become a vessel for any malicious program though and it is not limited to just one style of attack.

The obvious solution here is to not use USB’s that you do not know the origin of. It is simple and easy for someone to load a USB stick and leave it laying somewhere for someone to pick up and use.

9. BOT net attacks

<http://www.itbusinessedge.com/slideshows/show.aspx?c=87289&slide=3>

The largest threat from a BOT net comes from its ability to spam a web site with excessive amounts of traffic, thus causing it to crash or go down. This targeted attack is called a DDOS attack (direct/dedicated denial of service) and is used by hackers throughout the world to crash web pages, web servers, and more. BOT nets are usually comprised of large quantities of comprised computers that are controlled remotely from one location. These BOT nets are usually extremely difficult to track back to a specific person since the scale can be so large.

<http://rt.com/news/193032-mac-infected-hackers-reddit/>

This article describes a recent infection on Mac books. It created a large spam BOT net and shows that not even macs are free from problems. In this case, they are still trying to track down where the commands are coming from and who is responsible for the infection.

The solution here is to regularly scan your computer, keep it patched and up to date, maintain a strong password and make sure to filter emails too (since it sends spam, who is to say it doesn’t spread via email).

10. Social engineering/ networks

http://www.forbes.com/sites/davidcoursey/2012/01/10/top-12-security-risks-for-2012-is-your-company-ready/

This security threat actually covers a few sub topics such as phishing and spear phishing. The large problem here stems from human beings. In this case, a malicious user asks for a password and user name, or tricks someone into giving them up. This can be extremely effective, because humans usually don’t even realize what they are doing. It can also include getting someone to do something malicious for them (it doesn’t have to be credentials).

<http://www.foxbusiness.com/industries/2014/10/06/jpmorgan-bracing-for-spear-phishing-campaign-sources/>

This article talks about how JP Morgan is about to get hit with a lot of phishing and spear phishing. They are being targeted by hackers from outside the US who have already hit their Database for customer info. They are likely hoping to get access to the database in a more discreet way (ie with employee credentials) and then executing whatever malicious intent they have from there.

The best way to secure against this is to tell employees how to identify a phishing attempt. There really is not a patch for human stupidity and someone is likely to contact an African prince in an attempt to get a few million dollars from him for paying for his plane ticket to America. The other solution to this is to limit user access to various data structures and utilize a good DMZ.