A BEGINNER'S GUIDE TO

COMPUTER VULNERABILITIES

WHAT IS A COMPUTER VULNERABILITY?

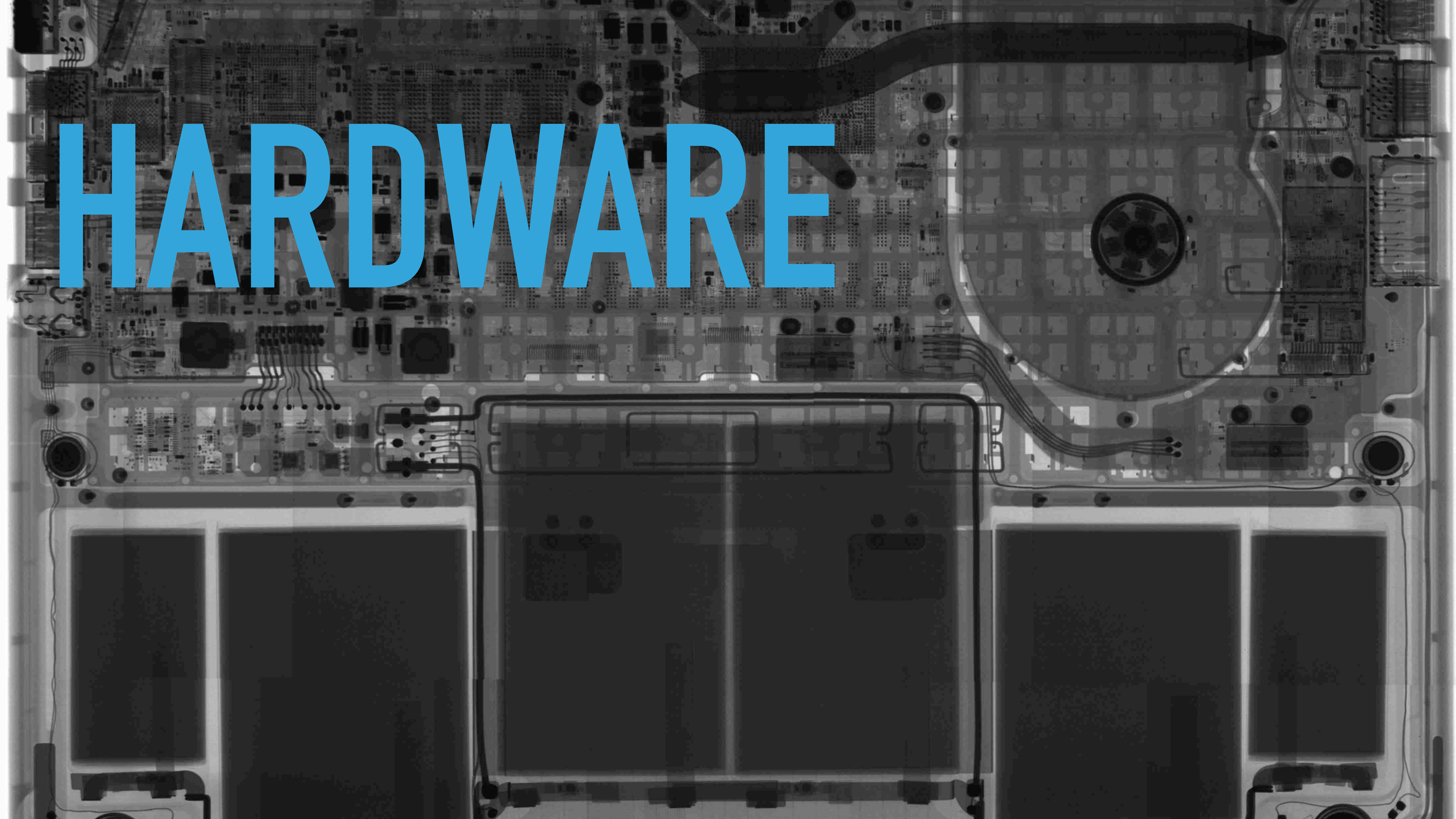
A STATE OF BEING EXPOSED TO A POSSIBLE ATTACK REDUCING A SYSTEM'S INFORMATION ASSURANCE

CLASSIFICATION OF VULNERABILITIES

CLASSIFICATION OF TYPES OF VULNERABILITIES ACCORDING TO ASSETS

- Hardware
- Software
- Network
- Personnel
- Physical site
- Organizational

We will take a look at hardware, software and network vulnerabilities


HARDWARE

HARDWARE VULNERABILITIES

- Hardware vulnerabilities used to only consist of the hardware (i.e. laptops, server, storage drives, etc.) being exposed to unsuitable conditions such as high temperatures, water (flooding) etc.
- Last year huge hardware design vulnerabilities, Meltdown and Spectre, were discovered in all major microprocessor products by Intel, AMD and ARM.

MELTDOWN AND SPECTRE

- Meltdown and Spectre were essentially present due to the design of the processors (i.e. the way they work). This vulnerability allowed attackers to access the cache of a CPU which usually held encryption keys, passwords and other sensitive information.
- Although Meltdown and Spectre could be remediated using a software update, it caused CPU performance to decrease by up to 30% in some cases. The only true way to remediate these hardware vulnerabilities is to replace the hardware.


SOFTWARE

SOFTWARE VULNERABILITIES

- This is a big one!
- National Institute of Science and Technology (NIST) has an excellent catalogue that lists all identified software security vulnerabilities in the National Vulnerability Database (NVD).
- Apple's Secure Coding Guide categorizes software security vulnerabilities into the following sets:
  - Buffer overflows
  - Unvalidated inputs
  - Race conditions
  - Access-control problems
  - Weakness in authentication, authorization, or cryptographic practices

BUFFER OVERFLOWS

- To understand a buffer overflow attack, we first need to understand how a computer program is stored in memory. There are three main parts to a program, *code*, *stack*, and the *heap*. When a program is ran by an operating system, the program's thread is allocated space for the stack and the heap. These grow during runtime as a program requests space on these sections of memory but they are ultimately bounded in size. Once a program goes past the end of the buffer of a stack or heap, the program starts accessing memory of another program (can even be kernel memory).
- Here is a great video on how a buffer overflow attack works!

UNVALIDATED INPUTS

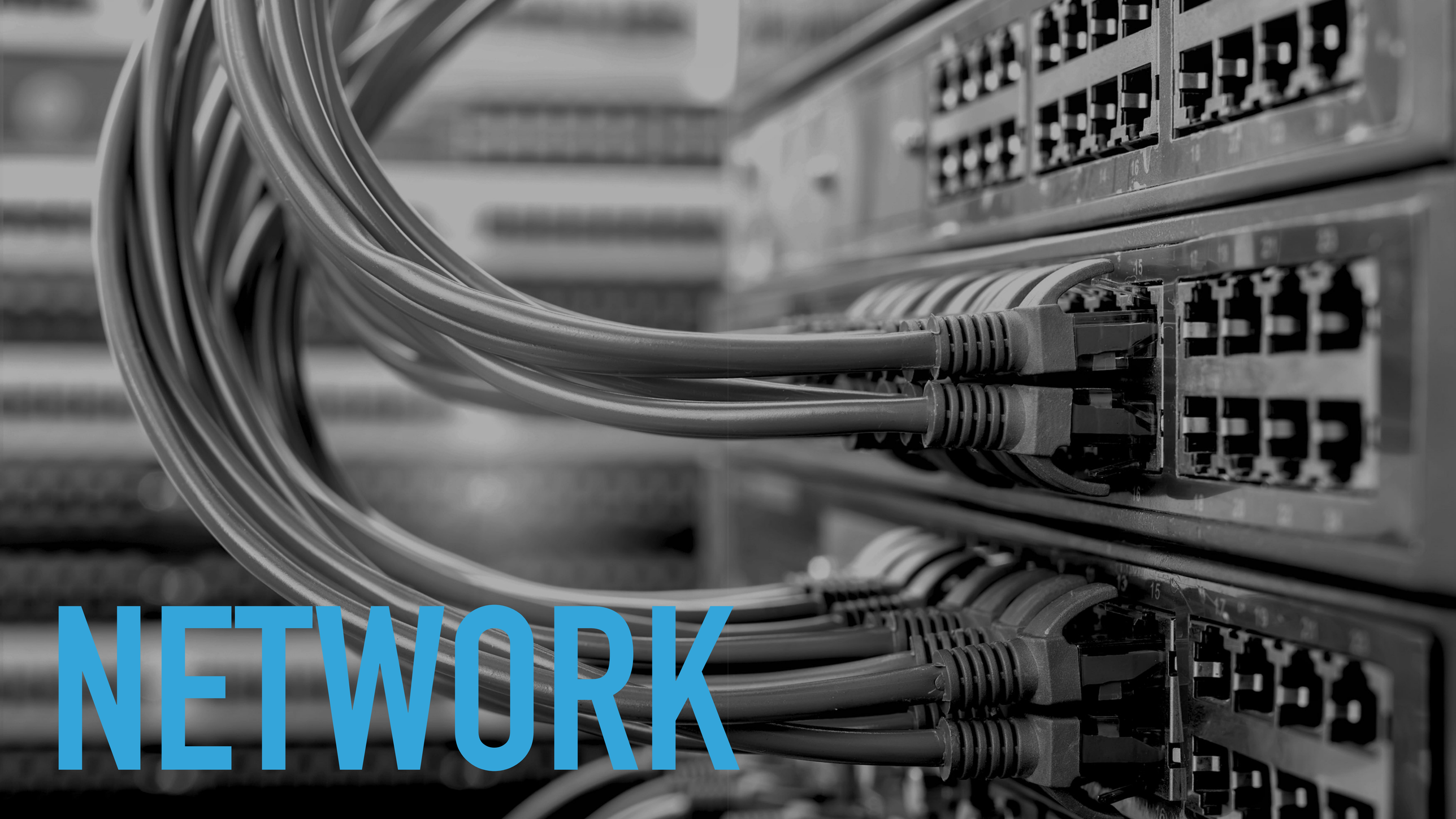
- Its a general rule of thumb, always check the external inputs before handling or storing the data.
- DEMO

RACE CONDITIONS

- If your program requires a certain order of execution to function properly and it breaks if it is executed out of order, that is a race condition. An attacker can exploit this property of your program to gain access to sensitive or unauthorized data segments of memory.
- To prevent race conditions, make use of proper method synchronizations and make your code fault tolerant.

ACCESS-CONTROL PROBLEMS

- There are many access control systems/mechanisms that are used when an application runs on a device. The operating system’s memory management system determines which parts of memory a program can access. Files on disk have permissions written to them telling the operating system to check before granting user access. Most of the time, problems in access control occur because the programmer doesn’t properly implement access control in their programs or they grant root permissions to the program. This creates a vulnerability because in case of a breach, the attacker can gain access to unauthorized sections of memory/files.


NETWORK

NETWORK SECURITY VULNERABILITIES

- Making sure an organization's network is secure is just as important as securing an application that runs on the organization's servers. The end goal is always to secure information. An attacker that can get into a network can cause just as much harm as an attacker exploiting an application's vulnerability.

RULE #1: FIX YOUR PASSWORDS

- To authenticate a user on a network, use a very strong password. Educate all users on the network to choose a password that is virtually impregnable. Weak passwords that are about 7 or 8 characters in length can be brute-force cracked within a day with the current processing power at hand.
- Checkout this COMPUTERPHILE video on how to choose a password.

RULE #2: PROPERLY CONFIGURE FIREWALL RULES

- “If it ain’t broke, why fix it” is not the right way to go about it when it comes to setting and configuring a network firewall. Networks are like a creature that morphs as time goes by. Some parts of the network get abandoned and forgotten but they still maintain high level access in the DMZ which attackers can use as an easy entry into an organizations information database.
- Regular review of firewall rules and network analysis is crucial to maintain a secure network.

RULE #3: UPDATE DEVICES/ENDPOINTS

- A network is made up of multiple devices/endpoints. It just takes one to be vulnerable to an attack to compromise the network.
- It is important to apply patches and updates to all devices on the network to ensure network safety.

QUESTIONS ?

FURTHER READINGS/VIDEOS

- NIST - Risk Management Guide for Information Technology Systems
- A Structured Approach to Classifying Security Vulnerabilities - Robert C. Seacord and Allen Householder
- BS ISO/IEC 27005:2008 - Information technology - Security techniques - Information security risk management
- OWASP Top 10 2013
- Web Application Security Stats
- Dumb Ideas in InfoSec
- Apple Security Practices
- The Top 5 Network Security Vulnerabilities
- Vulnerabilities in Network Systems
- National Vulnerability Database (NVD)
- Krack Attack (WiFi Vulnerability)
- SQL Injection Attack (Web Application)
- Denial of Service Attack (Network)
- Man in the Middle & Superfish (Distributed Application)
- Cross-Site Scripting (Web Application)
- CSRF Attack (Web Application)