Chapter 1

Securit Blanket or Security Theatre?

Introduction

• How Dependent Are We on Computers?
• What is Computer Security?
• Threats
• Harm
• Vulnerabilities
• Controls
• Analyzing Security with Examples
How dependant are we on Computers?

• What if all computers suddenly fail?
• No internet!
• No smart phones working…

How dependant are we on Computers?

• Cascading effects…
  – Software defect causes mobile phone disruption
  – Revered to desktops – consequently overloading
  – Air traffic control depends on wired comms
  – Stock exchange cannot function
  – Power grid shuts down – uncontrollable
  – …
How dependant are we on Computers?

- Testing cyber security readiness
  - Cyber security exercises organised by govs. etc.
  - Goals:
    - Anticipate unwanted cyber events
    - Awareness of cyber risks
    - Test existing response plans

- Nov 2010: First cyber security stress test by EU
- Attended by 22 EU nations and 8 observers
- Lessons learnt:
  - Private sector must be involved
  - International level lacking – still testing on national level!
  - First step towards building international trust.
  - Incident handling much different amongst nations.
  - International directory of contacts created: coordination
- US Cyber Storm event - biannual
How dependant are we on Computers?

• Are these scenarios realistic or implausible?
  – Are they designed to:
    • Confirm our readiness (security blanket)?
    • Or exacerbate our worries (security theatre)?
  – Can we determine causes?
  – Can we mitigate them?

How dependant are we on Computers?

• Imagine…
  – You drive and suddenly car breaks/accelerates self
  – Bank account overdrawn, yet you know there’s fund
  – Dr phones – you have terminal illness…
  – Leading party loses election, but it’s impossible…
• Should you be worried?
• Could it be real, or result of computer failure?
How dependant are we on Computers?

- Computers are everywhere!
- Can we and should we depend so much on computers?
  - How much should we trust them?
  - Engineers used them to build bridges – bridges nowadays do not often fail!
  - We started to trust them…
  - But software are engineered differently!

<table>
<thead>
<tr>
<th>How dependant are we on Computers?</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Protecting software in automobiles</td>
</tr>
<tr>
<td>– Dozens of microcontrollers in cars:</td>
</tr>
<tr>
<td>• Radio, GPS, Media players, Climate control, Intelligent mirrors, Instrument panel, Engine management, Airbags, Central computer, Telematics unit, Park distance control, Heads-up display, Tyre pressure control, ABS, DSC, Traction control, Light sensors, Bluetooth, Wifi… and many more!</td>
</tr>
<tr>
<td>– Engineers do not expect bolt-ons to be modified</td>
</tr>
<tr>
<td>– Software engineers do! They expect hackers!</td>
</tr>
</tbody>
</table>
How dependant are we on Computers?

- Protecting software in automobiles
  - What automotive functions have hackers hacked?
    - Disabling the brakes
    - Selectively braking individual wheels on demand
    - Stopping the engine
    - Bypass rudimentary network security protections in car
    - Maliciously bridging between car’s two internal subnets
    - Embedding malicious code in a car’s telematics unit that will erase any evidence of its presence after a crash

How dependant are we on Computers?

- Does this sound familiar?
  - “Security experts have long sought to inform designers and developers of security risks and countermeasures. Unfortunately, all too often the pleas of the security community are ignored in the rush to add and deliver features that will improve sales.”
How dependant are we on Computers?

• Types of failures for a bridge:
  – Physical damage
  – Warning signs, e.g. cracks, earthquake warnings…
  – So failure is slow and anticipated, directly proportional to time.

How dependant are we on Computers?

• Types of failures for a computer:
  – Physical damage, but even without physical touch:
  – Hardware failure (wear and tare)
  – Hardware working intermittently (worse!)
  – Failure happens instantly (no sign of wear and tare)
  – Small and harmless, like (“click here” button)
    • Harmless: do nothing, redirect to site…
    • Harmful: corrupts a file, erase the disk…
  – Can be readily apparent: a screen goes blank
  – Can be stealthy: keystroke logger.
How dependant are we on Computers?

• Computer security addresses all these failures
• The computers we consider range from:
  – small chips, embedded devices, stand-alone computers, clouds, private networks, public networks, and the rest of the Internet
  – They constitute the backbone of what we do: commerce, communication, health care, and more.
• Understanding failure can lead us to improvements in the way we lead our lives

What is Computer Security?

• Computer security is the protection of the items you value on a computer, called the assets
• Many types of assets:
  • Hardware
  • Software
  • Data
  • People
  • Processes, or combinations of these.
What is Computer Security?

• To determine what to protect, we must first identify what has value and to whom.

• Much hardware/software are off-the-shelf
  – Commercially available
  – Not custom-made

• What is important on your computer is content:
  – photos, tunes, papers, email, messages, projects, calendar information, ebooks (with your annotations), contacts, information, code you created, and the like.
What is Computer Security?

- After identifying the assets to protect, we next determine their value.

- We make value-based decisions frequently, even when we are not aware of them:
  - when you go for a swim you can leave a bottle of water on a towel on the beach
  - But not your wallet or cell phone.

- The difference relates to the value of the assets:
The Vulnerability–Threat–Control Paradigm

- Goal of computer security: protecting assets.
- We use a framework that describes how assets may be harmed and how to counter harm.
- A vulnerability is a weakness in the system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm.

For instance, a particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user’s identity before allowing data access.
- A threat to a computing system is a set of circumstances that has the potential to cause loss or harm.
The Vulnerability–Threat–Control Paradigm

• There are many threats to a computer system, including human/computer-initiated ones.

• We have all experienced the results of inadvertent human errors, hardware design flaws, and software failures.

• But natural disasters are threats, too; they can bring a system down when the computer room is flooded or the data center collapses from an earthquake, for example.

The Vulnerability–Threat–Control Paradigm

• A human who exploits a vulnerability perpetrates an attack on the system.

• An attack can also be launched by a system,

• As when one system sends an overwhelming flood of messages to another, virtually shutting down the second system’s ability to function.

• Unfortunately, we have seen this type of attack frequently, as what kind of attack?
  – Denial-of-service attacks
The Vulnerability–Threat–Control Paradigm

• How do we address these problems?
• We use a control/countermeasure
• A control is an action, device, procedure, or technique that removes/reduces a vulnerability.
• In general, we can describe the relationship among threats, controls, and vulnerabilities as:
• A threat is blocked by control of a vulnerability.
• In Figure 1-3, the difference between a threat, vulnerability, attack and control is illustrated:
The Vulnerability–Threat–Control Paradigm

• Another real-life example:

  – Chrysler said it was issuing a voluntary recall to update the software in affected vehicles.
  – The company added that hacking its vehicles was a "criminal action".
  – Security researchers Charlie Miller and Chris Valasek demonstrated that it was possible for hackers to control a Jeep Cherokee remotely, using the car's entertainment system which connected to the mobile data network.
The Vulnerability–Threat–Control Paradigm

- Another real-life example:
  - The two security researchers are due to reveal more information about their work at Def Con 2015.
  - Shortly after the recall was announced, Mr Miller tweeted: "I wonder what is cheaper, designing secure cars or doing recalls?"
  - Fiat Chrysler said exploiting the flaw "required unique and extensive technical knowledge, prolonged physical access to a subject vehicle and extended periods of time to write code".

The Vulnerability–Threat–Control Paradigm

- Another real-life example: **Affected vehicles**
  - 2013-2015 MY **Dodge Viper** specialty vehicles
  - 2013-2015 **Ram 1500, 2500 and 3500 pickups**
  - 2013-2015 **Ram 3500, 4500, 5500 Chassis Cabs**
  - 2014-2015 **Jeep Grand Cherokee/Cherokee SUVs**
  - 2014-2015 **Dodge Durango SUVs**
  - 2015 MY **Chrysler 200, Chrysler 300 and Dodge Charger sedans**
  - 2015 **Dodge Challenger sports coupes**
The Vulnerability–Threat–Control Paradigm

• Yet another recent real-life example:

  – A security researcher for airline vulnerabilities told FBI he hacked into controls while on board a flight and made the aircraft climb and briefly fly sideways
  – He entered the IFE system and overwrote code on the aircraft’s Thrust Management Computer
  – He stated that he caused one of the airplane engines to climb resulting in a lateral or sideways movement of the plane during one of these flights
The Vulnerability–Threat–Control Paradigm

• Yet another recent real-life example:
  – He used Vortex software after comprising/exploiting or ‘hacking’ the airplane’s networks. He used the software to monitor traffic from the cockpit system.
  – Roberts attempted to board a United flight to speak at a major security conference but was stopped by the airline’s corporate security at the gate.
  – He’d been removed from an earlier United flight and was questioned for four hours after jokingly suggesting on Twitter he could get the oxygen masks on the plane to deploy.

The Vulnerability–Threat–Control Paradigm

• In this course we take the approach of picking a particular type of threat, usually an attack.
• From that threat we determine vulnerabilities that could allow the threat to cause harm.
• We then explore the countermeasures that can control the threat or neutralize the vulnerability.
• This course is about protecting assets by countering threats that exploit vulnerabilities.
The Vulnerability–Threat–Control Paradigm

• Before we can protect assets, we have to know the kinds of harm we have to protect them against, so now we explore threats to valuable assets.

Threats

• We can consider potential harm to assets in two ways:
  – First, we can look at what bad things can happen to assets
  – Second, we can look at who or what can cause or allow those bad things to happen
• These two perspectives enable us to determine how to protect assets.
Threats

• Think for a moment about what makes your computer valuable to you.

• First, you use it as a tool for sending and receiving email, searching the web, writing papers, and performing many other tasks, and you expect it to be available for use when you want it.

• Without your computer (or mobile device) these tasks would be harder, if not impossible.

Threats

• Secondly, you rely heavily on your computer’s integrity.

• When you write a paper or do an assignment and save it, you trust that it will reload exactly as you saved it, without it being corrupted.

• Similarly, you expect that the photo a friend passes you on a flash drive will appear the same when you load it into your computer as when you saw it on your friend’s computer.
Threats

• Finally, you expect the “personal” aspect of a computing device to stay personal, meaning you want it to protect your confidentiality.
  • For example, you want your email messages to be just between you and your listed recipients; you don’t want them broadcast to other people.
  • And when you write an essay, you expect no one else to be able to copy it without your permission.

Threats

• These three aspects, availability, integrity, and confidentiality, make your computer valuable to you.
  • Also known as C-I-A as early as 1973
  • The C-I-A triad can be viewed from a different perspective: the nature of the harm caused to assets. Harm can also be characterized by four acts (threats): interception, interruption, modification, and fabrication.
Confidentiality

• Definition:
  – Only authorized people or systems can access protected data.

• Confidentiality can be difficult:
  – Who determines which people or systems are authorized to access the current system?
  – By “accessing” data, do we mean that an authorized party can access a single bit/whole collection/piece?
  – Can someone who is authorized disclose data to other parties?
Confidentiality

• Sometimes there is even a question of who owns the data:
  – If you visit a web page, do you own the fact that you clicked on a link, or does the web page owner, the Internet provider, someone else, or all of you?
• Confidentiality relates most obviously to data, although we can think of the confidentiality of a piece of hardware (a novel invention) or a person (the whereabouts of a wanted criminal).

Confidentiality

• Some properties that could mean a failure of data confidentiality:
  – An unauthorized person accesses a data item
  – An unauthorized process/program accesses an item
  – A person authorized to access certain data accesses other data not authorized (specialized version of unauthorized person accesses an item)
  – An unauthorized person accesses an approximate data value (for example, not knowing someone’s salary but knowing the salary range)
Confidentiality

• Some properties that could mean a failure of data confidentiality:
  – An unauthorized person learns the existence of a piece of data (for example, knowing that a company is developing a certain new product or that talks are under way about the merger of two companies).

Confidentiality

• Notice the general pattern of these statements:
  – A person, process, or program is (or is not) authorized to access (in particular way) a data item.

• We call the person, process, or program a subject, the data item an object, the kind of access (such as read, write, or execute) an access mode, and the kind of authorization a policy, as shown in Figure 1-5.
Integrity

Examples are many:

- A malicious macro in a Word document inserted the word “not” after some random instances of the word “is”; you can imagine the havoc that ensued. Because the document was generally syntactically correct, people did not immediately detect it.
- A model of the Pentium computer chip produced an incorrect result in certain circumstances of floating-point arithmetic. Circumstances of failure were rare, but Intel decided to replace the chips anyway.
- Many more examples in textbook…
Integrity

- Integrity is also difficult to pin down:
  - For example, if we say that we have preserved the integrity of an item, we may mean that the item is:
    - Precise, accurate, unmodified, modified only in acceptable ways, modified only by authorized people, modified only by authorized processes, consistent, internally consistent, meaningful and usable
  - Integrity can be enforced in much the same way as confidentiality: by rigorous control of who or what can access which resources in what ways.

Availability

- A computer user’s worst nightmare: you turn on the switch and the computer does nothing.
- Your data and programs are presumably still there, but you just cannot get to them.
- Could also be a degree of availability: slow
Availability

• As with integrity, different people expect availability to mean different things. Examples: An object/service is available if:
  – It is present in a usable form.
  – It has enough capacity to meet the service’s needs.
  – It is making clear progress without errors, and, if in wait mode, it has a bounded waiting time.
  – The service is completed in an acceptable period of time (performance).

FIGURE 1-6 Availability and Related Areas
Types of threats

• One way to analyze harm is to consider the cause or source.
• We call a potential cause of harm a threat. Different kinds of threats are shown in Fig 1-8.
Types of threats

• Too many ways to interfere with use of assets.
• Two retrospective lists of known vulnerabilities:
  – CVE, the Common Vulnerabilities and Exposures
    • http://cve.mitre.org/
    • dictionary of publicly known information security vulnerabilities and exposures.
  – CVSS, the Common Vulnerability Scoring System
    • http://nvd.nist.gov/cvss.cfm
    • Standard measurement system
    • to measure the extent of harm using accurate and consistent scoring of vulnerability impact
Types of Attackers

• Who are attackers? There’s no single profile!
  – People convicted for computer crimes. Convicted?
  – They wear business suits, have university degrees, and appear to be pillars of their communities.
  – Some are high school or university students.
  – Others are middle-aged business executives.
  – Some are mentally deranged, overtly hostile, or extremely committed to a cause, and they attack computers as a symbol.
  – Others are ordinary people tempted by personal profit, revenge, challenge, advancement...

• Individuals
  – Motives of fun, challenge, or revenge (personal).
  – “Quick and dirty” attacks
  – Robert Morris
    • Brought down the internet in 1988
    • Morris worm
  – Kevin Mitnick
    • Famous hacker
    • Spent time in jail
Types of Attackers

• Organized Worldwide Groups
  – Motives of mainly financial gain
  – Groups of (“underground”) hackers
  – Estonia case (see Chapter 15)
  – In October 2004, U.S. and Canadian authorities arrested 28 people from 6 countries involved in a global organized cybercrime ring to buy and sell credit card information and identities.

• Organized Crime
  – Motives of fraud, extortion, money laundering, and drug trafficking
  – Groups of (“underground”) hackers combining with traditional crime bosses to make money
  – Organized crime wants a resource; such criminals want to stay under the radar to be able to extract profit from the system over time.
Types of Attackers

• Terrorists
  – Motives of war and conflict
  – Terrorist groups using computers in following cases:
    • Computer as target of attack:
      – Denial-of-service attacks and web site defacements are popular
      – Attract attention and bring undesired negative attention
    • Computer as method of attack:
      – Stuxnet worm: disabling C&C systems in Iran nuclear reactors
    • Computer as enabler of attack:
      – Using computers to coordinate, e.g. cell phones, GPS, maps etc.
    • Computer as enhancer of attack:
      – Spread propaganda, hire attackers etc.

Harm

• The negative consequence of an actualized threat is harm
• Difficult to assign a specific number as the value of an asset, you can usually assign a value on a generic scale: low, medium, high
• Choosing the threats we try to mitigate involves a process called risk management, and it includes weighing the seriousness of a threat against our ability to protect.
Risk and common sense

• The nature and number of threats in the computer world reflect life in general:
  – The causes of harm are limitless and unpredictable.
  – Natural disasters like volcanoes and earthquakes happen with little/no warning, as do car accidents, heart attacks, flu, and random acts of violence.
  – To protect against accidents/flu, you can stay inside. But then you trade one set of risks for another: you are vulnerable to building collapse.
  – There are too many possible causes of harm to protect ourselves—or our computers—completely.

Risk and common sense

• Computer security is similar. We cannot protect against everything, but then we prioritize:
  – Only so much time/energy/money is available, so we address some risks and let others slide.
  – Or we consider alternatives, such as transferring risk by purchasing insurance
  – Or even do nothing if the side effects of the counter-measure could be worse than the possible harm.
  – The risk that remains uncovered by controls is called residual risk.
Risk and common sense

• A simplistic model of risk management:
  – calculate the value of all assets
  – determine the amount of harm from possible threats
  – compute the costs of protection
  – select safeguards (that is, controls or countermeasures) based on the degree of risk and on limited resources, and applying the safeguards to optimize harm averted.

• This approach is logical and sensible, but it has significant drawbacks: difficult, continuous, $$$

Method-Opportunity-Motive

• A malicious attacker must have three things to ensure success:
  – Method
  – Opportunity
  – Motive

• Depicted in Figure 1-10.
• Deny the attacker any of those three and the attack will not succeed.
Method

- Skills, knowledge, tools, and other things with which to perpetrate the attack.
- Lots of computer and hacker books + Internet
- scripts, model programs, and tools to test for weaknesses, often free of charge
- What is a script kiddie?
Method-Opportunity-Motive

• Opportunity
  – The right time and access to execute an attack
  – Many computer systems present ample opportunity:
    • Systems available to public are, by definition, accessible;
    • Other people are oblivious to the need to protect their systems, so unattended PCs are soft targets.
    • Some systems have private/undocumented entry points for administration or maintenance, but attackers can also find and use those entry points to attack the systems.

Method-Opportunity-Motive

• Motive
  – A reason why someone would like to execute an attack
  – Many already mentioned:
    • Money
    • Fame
    • self-esteem
    • Politics
    • terror
Vulnerabilities

- Weakness in a system
- Examples: weak authentication, lack of access control, errors in programs, finite or insufficient resources, and inadequate physical protection.
- Each of these vulnerabilities can allow harm to confidentiality, integrity, or availability.
- Each attack vector seeks to exploit a particular vulnerability.
- Next step: find ways to block threats by neutralizing vulnerabilities.

Controls

- A control or countermeasure is a means to counter threats.
- Harm occurs when a threat is realized against a vulnerability.
- To protect against harm, then, we can neutralize the threat, close the vulnerability, or both.
- The possibility for harm to occur is called risk.
Controls

• We can deal with harm in several ways:
  – prevent it, by blocking the attack or closing the vulnerability
  – deter it, by making the attack harder but not impossible
  – deflect it, by making another target more attractive (or this one less so)
  – mitigate it, by making its impact less severe
  – detect it, either as it happens or some time after the fact
  – recover from its effects

Controls

• Just like we have fences, gates, locks, re-enforced walls etc. as controls to protect our homes, we have them in computer security:
  – We have many controls at our disposal
  – Some are easier than others to use or implement.
  – Some are cheaper than others to use or implement.
  – Some are more difficult than others for intruders to override.
Controls

- classes and examples of each type of control:
  - **Physical** controls stop something tangible:
    - walls and fences
    - locks
    - (human) guards
    - sprinklers and other fire extinguishers
  - **Procedural** or **administrative** controls use a command that requires/advises people how to act:
    - laws, regulations
    - policies, procedures, guidelines
    - copyrights, patents, contracts, agreements
Controls

- classes and examples of each type of control:
  - Technical controls counter threats with technology (hardware or software):
    - passwords
    - access controls enforced by an operating system or application
    - network protocols
    - firewalls, intrusion detection systems
    - encryption
    - network traffic flow regulators

As shown in Figure 1-12, you can think in terms of the property to be protected and the kind of threat when you are choosing appropriate types of countermeasures.

None of these classes are necessarily better than or preferable to the others; they work in different ways with different kinds of results.

It can be effective to use overlapping controls or defense in depth: more than one control/class of control to achieve protection.
Analyzing Security With Examples

- The rest of the course will examine real-world examples of threats and analyse them by constantly identifying the following:
  - Threat
  - Harm
  - Vulnerability
  - Control
Conclusion

• Security situations arise every day
  – Difficult to distinguish between a security attack and an ordinary human or technological breakdown.
• A threat is an incident that could cause harm
• A vulnerability is a weakness through which harm could occur.
• Seldom can we achieve perfect security
• An attacker needs three things: method, opportunity, motive