

CSE 127 Computer Security

Alex Gantman, Spring 2018, Lecture 2

Security Foundations

Review

- What is a secure system?
 - “System that remains dependable in the face of malice” – Ross Anderson
- Reflections on Trusting Trust by Ken Thompson
- Chapter 1 from The Craft of System Security
- **First project is due next week (4/9 @ 10pm)**
 - Getting comfortable with the debugger and project submission system
- Office Hours: Th 6:30-7:30pm, CSE(EBU3B) 2106

Lecture Objectives

- Develop a mental framework for:
 - Thinking about security risks
 - Evaluating potential mitigation options
 - Analyzing tradeoffs

Threat Model

- **Threat Model** defines **Security Goals**: What are we trying to protect from whom?
 - **Assets** and **Attackers**
- Let's return to definition of a secure system
 - “System that remains dependable in the face of malice” -Ross Anderson
- What does “remains dependable” mean?
 - **Confidentiality, Integrity, and/or Availability (C.I.A.)** of particular system, component, or data are preserved
 - These are the **Assets** we need to protect
 - “An attacker must not be able to compromise the [Confidentiality | Integrity | Availability] of...”
 - Assets have value: Must understand value in order to decide how much to spend on protection
- What does “malice” mean?
 - Malice from whom? Curious roommate? Criminal? Law enforcement? Foreign military?
 - These are potential **Attackers**
 - Defined by **Capability** and **Intent**

Threat Model

- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
 - ***Assets*** and ***Attackers***
 - ***Confidentiality, Integrity, and/or Availability (C.I.A.)*** of particular system, component, or data

Confidentiality

- Prevention of unauthorized access to information
 - Unauthorized parties can't view protected information
 - aka Secrecy
- What are some scenarios where you want secrecy?

Integrity (& Authenticity)

- Integrity: Prevention of unauthorized changes
- Authenticity: Identification and assurance of origin

Integrity (& Authenticity)

- Prevention of unauthorized modification of information, process, or function
 - Unauthorized parties can't modify protected information or process
- Examples
 - Increasing your bank account balance without depositing money
 - Getting snacks from vending machine without paying for them
- What are some other examples?

Integrity (& Authenticity)

- Prevention of impersonation of another principal
 - aka origin integrity
- Example
 - Getting money from someone else's bank account using their credentials
- What are some other examples?

Availability

- Prevention of unauthorized denial of service to others
 - Unauthorized parties can't deny access to system for other users
- Examples
 - Physically rendering ATM unusable
 - Network denial of service attacks
- What are other examples of denial of service?

C.I.A. + Privacy

- Privacy: A person's right or expectation to control the disclosure of his/her personal information, including activity metadata
- What is the difference between privacy and secrecy?
 - Secret from whom?
 - Activity metadata
 - What can you figure out about a person just from their location history?

C.I.A. + Privacy

- Which security property is violated if someone ...
 - unplugs your alarm clock while you're sleeping?
 - changes the time on your alarm clock?
 - installs a camera in your room?

Threat Model

- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
 - ***Assets*** and ***Attackers***
 - Defined by ***Capability*** and ***Intent***

Attackers/Adversaries

- **Types**

- Individual
 - Outsider
 - Insider
 - Trusted/Privileged Insider
- Group
 - Ad hoc
 - Established
- Organization
 - Competitor
 - Supplier
 - Partner
 - Customer
- Nation-State

- Capabilities

- Time
- Money
- Training
- Access

- **Motivation/Intent**

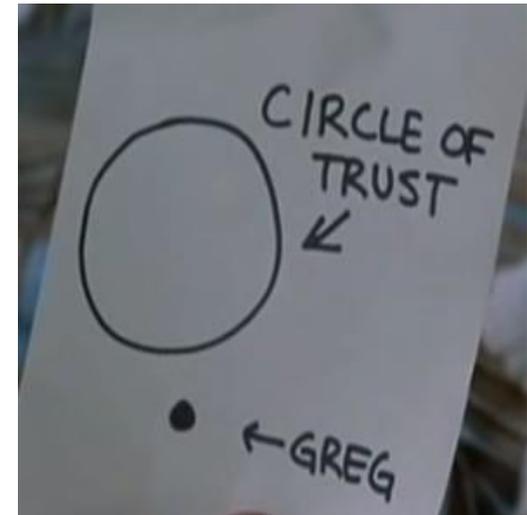
- Curiosity
- Fame
- Money
- National interest

Trust Boundary/Security Boundary

- Security Boundary
 - Perimeter around components of same trust level
 - Any data or signals coming in from outside is untrusted and potentially malicious
- Attack Surface
 - Set of interaction points across a security boundary
 - Parts of your system handling input from or otherwise interacting with less trusted and potentially malicious entities
 - Network perimeter, communication port, IPC, API, parser, etc.
 - Some highly sensitive systems are “air-gapped” to minimize the attack surface

Trust Boundary/Security Boundary

- One person's security boundary is another person's attack surface
- "A trust boundary exists any time we (the system designers or system owners) trust code, data, or human actors on one side of an interface more than we trust the other side of the interface."
 - John Regehr



Threat Model

- Your very first question in any security discussion should be

What's the threat model?



- Do not argue about attacks or defenses without understanding the threat model
- The threat model defines the problem to be solved.
 - If there is no consensus on the problem, there will be no consensus on solutions

Threat Model Example: Personal Items

- Asset
 - TV, jewelry
- Trust Boundary
 - Spouse, roommate
- Attacker
 - Roommate, thief
 - Do you have to worry about nation state attackers?



Threat Model Example: Web Browser

- Assets
 - Page contents (confidentiality? integrity?)
 - Cookies
 - Stored passwords
 - Browsing history
- Trust Boundary
 - Browser? Site? Tab? Frame?
- Attacker
 - Local malware? Malicious site? Roommate? Spouse? Law Enforcement? Nation States?

Threat Model

- A cautionary note
 - Your threat model is your problem scope.
 - Attackers don't care about your threat model.
 - Just because an asset or an attacker are outside your threat model, does not mean that they do not exist.
 - It just means that you have explicitly decided that you will not address them in your solution.
- PS. A note on problems and solutions

Risk Assessment

- Now that we know what we want to protect and from whom, we can reason about what risks attackers can pose to our assets
- Security is not binary
- Everything has some measure of risk
 - A rough, informal scale: from Very Low to Very High
 - Calculated as some combination of:
 - Likelihood – the probability that a security threat will materialise
 - Impact – the adverse affects that will occur if a threat does materialise
- We evaluate security risk relative to value
 - Is this risk worth taking?
 - Similar to health and financial risks

Risk Assessment

- Risk assessment/risk analysis process for assessing risk in a system:
 1. Start by understanding system requirements
 2. Identify assets and attackers
 3. Establish security requirements
 4. Evaluate system design
 5. Identify threats and classify risks
 6. Address identified risks

1. Understand system requirements

- You can't make a system secure if you don't know what it is supposed to do
- What is the system supposed to do?
 - What is the security boundary?
 - Look for vague or open-ended requirements. These will likely be interpreted in different ways by different architects and developers.
- What is the lifecycle? How will it be deployed?

2. Identify assets and attackers

- Who are the stakeholders?
 - Who has a vested interest in the system running correctly/securely.
- What needs to be protected?
 - What is the value? To whom?
- From whom does it need to be protected?

3. Establish security requirements

- Think about the CIA+P classification
- “An attacker must not be able to compromise the [Confidentiality | Integrity | Availability] of...”

4. Review system design

- You can't make a system secure if you don't understand how it works
- What are the different components in the system?
 - How will the components communicate?
 - How will they store and/or pass data?
- Draw a block diagram of the system
 - Indicate security boundaries and information flow
 - Describe protocol interactions
- Again, look for ambiguity and hidden assumptions
- This is only the starting point
 - The design will evolve as risks are identified and mitigated
- Later, make sure that implementation matches the design

5. Identify threats and classify risks

- Using the security goals and current design, identify **Threats** to security
 - Possible ways in which attackers (as defined by your threat model) could exploit vulnerabilities in system design or implementation to compromise the assets (as defined by your threat model)
- For each threat, determine likelihood and impact and convert to a risk “score”

Identifying Threats

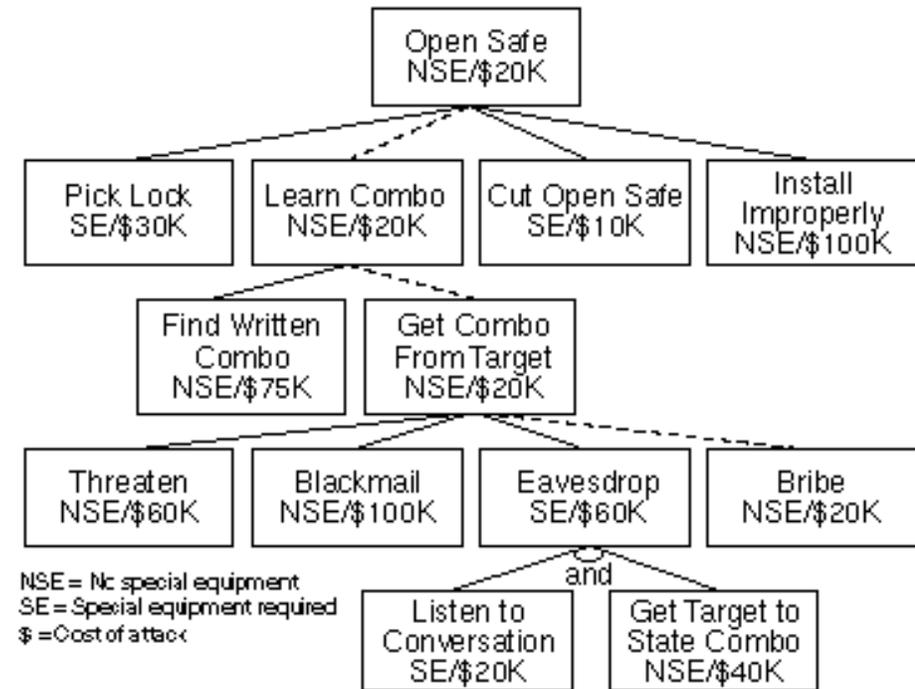
- Adopt an adversarial mindset
 - An attacker does not have to obey any rules established by the defender.
 - Requirements are for builders, not breakers.
 - It doesn't matter what is *supposed* to happen.
 - Review the source (and binaries) to see what actually happens.
 - Assume that the untrusted parts of the system work in the worst possible way – i.e. assume adversarial control.
 - That includes any external components, like storage, etc.
 - Look for assumptions about data coming across the security boundary and try to violate them.
 - If the documentation says that commands have to have a certain format, then evaluate what can happen when the format is incorrect.
 - If the documentation says that some commands are supposed to be executed in a certain order, then evaluate what can happen when the order is wrong.

Identifying Threats

- Strategies
 - Start with the assumption that there is yet another way to attack the system
 - You just need to find it
 - Attack the weakest link
 - Focus on parts that are ambiguous or seem overly complex.
 - Look for ambiguity in requirements and design and see if different parts of the system were implemented using different interpretations.
 - Be curious, ask stupid questions.
 - Consider the security goals one-by-one and look for different types of attacks against that goal.
 - Attack Trees
 - Consider the types of attack one-by-one and look for different ways that these attacks can affect a goal.
 - STRIDE

Identifying Threats

- Attack Trees
 - Start with an attacker's goal as the root of a tree
 - "Read secret ..." or "Modify value..."
 - Different ways of achieving a goal (sub-goals) become child nodes
 - Can be annotated with likelihood, cost, etc.



Identifying Risks

- STRIDE: Microsoft-developed model
 - Start with possible attack types and evaluate how they can violate security goals
 - Spoofing identity.
 - An example of identity spoofing is illegally accessing and then using another user's authentication information, such as username and password.
 - Tampering with data.
 - Data tampering involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet.
 - Repudiation.
 - Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise—for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Nonrepudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package.
 - Information disclosure.
 - Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers.
 - Denial of service.
 - Denial of service (DoS) attacks deny service to valid users—for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability.
 - Elevation of privilege.
 - In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed.

Assessing Risk

- We can “score” identified risk based on Likelihood and Impact:
- Likelihood:
 - Very low: Unlikely to occur in practice.
 - Low: Unlikely or requires significant resources.
 - Medium: Motivated and well-resourced attacker could achieve with persistence.
 - High: Likely to occur within the product’s lifetime.
 - Very high: Attack is inevitable and will occur quickly.
- Impact:
 - Very low: Causes virtually no damage (not even reputational).
 - Low: Causes damage that could be easily tolerated.
 - Medium: Causes damage that will have a noticeable effect on a stakeholder.
 - High: Causes damage that will permanently affect a stakeholder.
 - Very high: Causes damage that will cripple the stakeholder.

Assessing Risk

- Combine likelihood and impact to generate overall risk rating.
 - Allows risks to be ranked.
- Risk assessment is subjective
 - Use your knowledge and common sense to make sure that the risk ranking is reasonable.

		Impact				
		Very Low	Low	Moderate	High	Very High
Likelihood	Very Low	Very low	Very Low	Very Low	Low	Low
	Low	Very Low	Very Low	Low	Low	Moderate
	Moderate	Very Low	Low	Moderate	Moderate	High
	High	Low	Low	Moderate	High	Very High
	Very High	Low	Moderate	High	Very High	Very High

6. Address identified risks

- Once risk is identified and classified we must decide what to do about it
- ***Avoid***: Remove component that creates the risk
 - Example: Disable the feature
- ***Mitigate***: Add measures that decrease likelihood or impact
 - Example: Better locks
- ***Transfer***: Make it someone else's problem
 - Example: Insurance
- ***Accept***: Do nothing
 - There's always residual risk we must accept
 - <https://www.youtube.com/watch?v=gIG3zqvUqJY>
 - Pwnie-nominated performance <https://pwnies.com/>



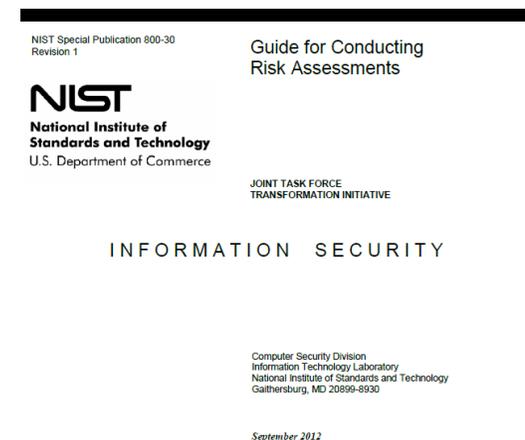
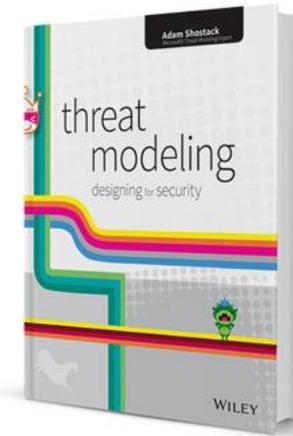
Risk Acceptance

- It is impossible to offer functionality without risk.
- ***Residual risk*** is left over when all the controls and protections have been installed.
- Decision to accept risk should be made by someone who has sufficient authority to do so.
- *Accepting the risk does not make it go away.*
 - Regardless of why you choose to accept it
 - If there's no money or time to deal with a threat, does not make the risk disappear

Additional Resources

- *Threat Modeling: Designing for Security*
 - by Adam Shostack
 - <https://threatmodelingbook.com/>

- *NIST Guide for Conducting Risk Assessments*
 - SP 800-30 Rev. 1
 - <https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final>



Review

- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
 - ***Assets: Confidentiality, Integrity, and/or Availability (C.I.A.)*** of particular system, component, or data are preserved
 - ***Attackers: Intent and Capability***
- Security is not binary, everything has some measure of ***Risk***
 - Calculated as some combination of ***Likelihood*** and ***Impact***
 - Is this risk worth taking?

Review

- Risk assessment/risk analysis process for assessing risk in a system.
- Adversarial Mindset
 - An attacker does not have to obey any rules established by the defender.
 - Assume that the untrusted parts of the system work in the worst possible way.
 - Violate assumptions.
 - An attack is possible, you just need to find it.
- Once risk is identified and classified we must decide what to do about it:
 - Avoid, Mitigate, Transfer, Accept

Homework

- **First project is due next week (4/9 @ 10pm)**
 - Getting comfortable with the debugger and project submission system
- Read *Smashing The Stack For Fun And Profit* by Aleph One
 - <http://phrack.org/issues/49/14.html#article>
- Read Chapter 6 from *The Craft of System Security*

Next Lecture...

Low Level Software Security I: Buffer Overflows