STRIDE

THREAT MODELING BOOK
CHAPTER 3: STRIDE

Prepared by: Dr. Alia Alabdulkarim

What is STRIDE?

انتحال

🔒**STRIDE stands for: S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, **E**levation of privilege

Spoofing ≠ confidence

🔒It is one of the methods that can be used in **threat modeling** to **find threats** (alternative to EoP game and attack trees)

🔒How to use it in threat modeling?

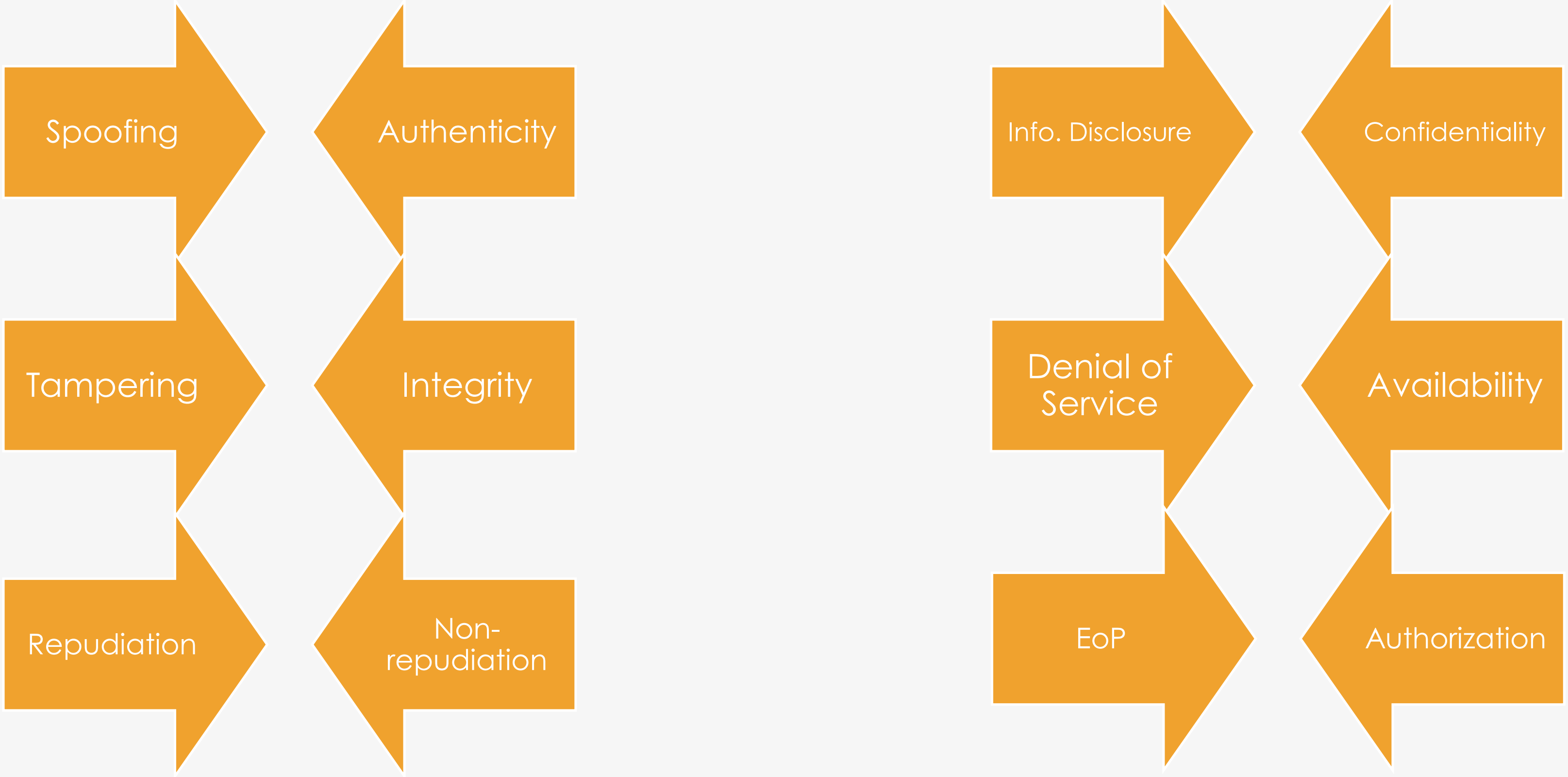
- 🔑Using STRIDE, look at your diagram and look for more threats
- 🔑Make a list of affected areas in the diagram for each threat

Understanding STRIDE and Why It’s Useful

Threat	Property Violated	Definition	Example
Spoofing ≠ انتحال شخصية	Authentication	Impersonating something or someone else.	Pretending to be any of Bill Gates, Paypal.com or ntdll.dll
Tampering تعديل كي ملف أو كود	Integrity	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the network
Repudiation إنكار	Non-repudiation	Claiming to have not performed an action.	“I didn’t send that email,” “I didn’t modify that file,” “I *certainly* didn’t visit that web site, dear!”
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site.
Denial of Service ??	Availability	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole.
Elevation of Privilege ⇒ أخطرهم	Authorization	Gain capabilities without proper authorization	Allowing a remote Internet user to run commands is the classic example, but going from a limited user to admin is also EoP.

Understanding STRIDE and Why It’s Useful

- The STRIDE threats are the opposite of some of the properties you would like your system to have



Spoofing → ← Authenticity

Tampering → ← Integrity

Repudiation → ← Non-repudiation

Info. Disclosure → ← Confidentiality

Denial of Service → ← Availability

EoP → ← Authorization

Understanding STRIDE and Why It's Useful

- Note that as you're using STRIDE to look for threats, you're **simply enumerating the things that might go wrong**
- The exact mechanisms for how it can go wrong are something you can develop later
- STRIDE is **not intended for categorizing attacks or threat**
  - The **goal** of STRIDE is **to help you find attacks**
  - Categorizing them might help you figure out the right defenses, or it may be a waste of effort

Spoofing

- Spoofing is pretending to be something or someone other than yourself:
  - Spoofing on the local machine
  - Spoofing over a network

Spoofing On the Local Machine

🔒The attacker may supply data that your code interprets, thinking that your code (or a previous instantiation or thread) wrote that data

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a **process**	Creates a file before the real process	Then your process trust it and use it
	Renaming/linking	Create a version of "**sudo**" and alter **PATH variable** ⇒ تغير المعنى

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a **filename**	Creates a file in the local directory	Library, executable or config file
	Creates a link, changes it	Also called '**TOCTOU race condition**'

time of checke Time to use

Spoofing Over a Network

- 🔒They can spoof ARP requests
- 🔒They can spoof IP packets to make it appear that they're coming from somewhere they are not
- 🔒They can spoof DNS packets Domain name Servir ⇒ يترجم آيبيا
- 🔒Spoofing websites and/or emails: Phishing attacks ⇒ Spoofing DNS

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a **machine**	ARP spoofing ⇒	يترجم الآيبي الى ماك
	IP spoofing	
	DNS spoofing	
	DNS compromise	Can be at the TLD, registrar or DNS server

Spoofing Over a Network

🔒Access to the person’s account OR pretending to be them through an alternate account

🔒Phishing is a common way to get access to someone else’s account.

🔒However, there’s often little to prevent anyone from setting up an account and pretending to be you (E.g. creates fake account in social media using your name).

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a **person**	Take over account	
	Set the display name	
Spoofing a **role**	Declares themselves to be that role	Sometimes opening a special account with a relevant name

Tampering ⇒ تلاعب تغيير

- 🔒Tampering is modifying something:
  - 🔑Tampering with a File
  - 🔑Tampering with Memory
  - 🔑Tampering with a Network

Tampering with a File

- 🔒Attackers can modify files wherever they have write permission
- 🔒When your code has to rely on files others can write → malicious
  - 🔑Example: cache poisoning attacks

Threat Example	What the Attacker Does	Notes/Examples
Modifying a file…	… which you own and you rely on	
	… which they own and you rely on	
Modifying a file on a server…	…you own	
	…they own (or take over)	

Tampering with Memory

Threat Example	What the Attacker Does	Notes/Examples
Modifying code	Changes your code to suit themselves	Hard to defend against if the attacker is running code inside the trust boundaries (running the code as the same user)
Modifying data they've supplied	Supplies data to a pass by reference API, then changes it	It's recommended to pass by value, not by reference when crossing a trust boundary

Tampering with a Network

- Network tampering often involves a variety of tricks to bring the data to the attacker’s machine
- With radio interfaces like WiFi and Bluetooth, more and more data flow through the air which bring you data that is not always needed.
- To defend against tampering (and/or spoofing), many network protocols were designed with the assumption you needed special hardware to create or read packets
    - Software-defined radio (SDR) has invalidated the need for special hardware.

Threat Example	What the Attacker Does	Notes/Examples
Redirects the flow of data to their machine	Uses an attack at some network layer to redirect traffic	
Modifies data flowing over the network		Easier (and more fun) with wireless networks

Repudiation

- Repudiation is claiming you didn’t do something, or were not responsible for what happened:
  - **R**epudiating on Action
  - **R**epudiation Attacks on Logs

Repudiating on Action

🔒 People can repudiate honestly or deceptively

Threat Example	What the Attacker Does	Notes/examples
Repudiating an action	Claims to have not clicked	Maybe they did, maybe they didn’t, maybe they’re honestly confused
	Claims to not have received	1. Electronic or physical 2. Receipt is strange; does a client downloading email mean you’ve seen it? Did a network proxy pre-fetch images? Was a package left on a porch?
	Claims to be a fraud victim	
	Uses someone else’s account	

Repudiating on Action

- Those who repudiate are often not actually attackers, but people who have been failed by technology or process
  - Maybe they really didn’t click
  - Maybe the spam filter really did eat that message
  - Maybe Fedex didn’t deliver, or maybe Fedex delivered by leaving the package on a porch
- Good technological systems that both authenticate and log well can make it easier to handle repudiation issues

Repudiation Attacks on Logs

- Repudiation threats are also associated with your **logging system** and process
- There is also a class of attacks in which attackers will drop data in the logs to make **log analysis** tricky
- If you don’t properly define what you will be logging, an attacker may be able to break your log analysis system

Threat Example	What the Attacker Does	Notes/Examples
	Discovers there are no logs	
Modifies data flowing over the network	Puts data in the logs to confuse you	</tr></html>

Information Disclosure

🔒Leaking information that shouldn't be disclosed:

- 🔑Information Disclosure (Processes)
- 🔑Information Disclosure (Data Stores)
- 🔑Information Disclosure (Data Flow)

Information Disclosure (Processes)

- 🔒Leaking memory addresses can help bypass Address space layout randomization (ASLR) and similar defenses
- 🔒Leaking design details might mean exposing anti-fraud rules “your account is too new to order a diamond ring”

Threat Example	What the Attacker Does	Notes/Examples
Extracts user data	Exploits bugs like SQL injection to read DB tables	Can find this by looking to data stores, but here the issue is the process returning data it shouldn’t
	Reads error messages	
Extracts machine secrets	Reads error messages	Cannot connect to database ‘foo’ as user ‘sql’ with password ‘&IO*(^&’

Information Disclosure (Data Stores)

- Data in file names [May 2020 layoffs, Termination Letter for Alice.docx]
- Data can be extracted from the device using an operating system under the attacker’s control
  - Hard drives are often decommissioned without full data deletion

Sub-category	What the Attacker Does
Permissions	Take advantage of missing or inappropriate ACLs Take advantage of bad database permissions
Security	Find crypto keys on disk or in memory Get data from logs/temp files See interesting information in filenames/directory names
Network	See data traversing a network

Information Disclosure (Data Flow)

- Data flows are particularly susceptible to information disclosure attacks when information is flowing over a network
- However, data flows on a single machine can still be attacked, particularly when the machine is shared

Sub-category	What the Attacker Does
Network	Read data on a network
	Redirects traffics to enable reading data on the network
Metadata	Learns secrets by analyzing traffic
	Learns who talks to whom by watching the DNS
	Learns who talks to whom by analyzing social network information

Denial of Service

- Denial-of-service attacks overwhelm or absorb a resource that is needed to provide service.
- Temporary VS. Persistent
  - **Temporary:**
    - Works while the attacker is attacking (E.g. filling up network bandwidth)
  - **Persistent:**
    - Can remain in effect until a reboot (E.g. while(1){fork();}),
    - Beyond reboot (E.g. filling up disk)
- Amplified VS. unamplified
  - Amplified attacks are those whereby small attacker effort results in a large impact

Denial of Service

Threat Example	What the Attacker Does
Against a process	Absorb memory (ram or disk)
	Absorb CPU
Against a data store	Fills the data store
	Makes enough requests to slow the system
Against a data flow	Consumes network resources

Elevation of Privilege ("EoP")

- 🔒It's important to understand that these exploits are not limited to the attack surface
- 🔒The simplest authorization failure is to not check authorization on every path

Threat Example	What the Attacker Does	Notes/Examples
EoP Against process via corruption	Sends inputs the code doesn't handle properly	Very common, usually high impact
	Gains read/write access to memory	Writing memory more obviously bad
EoP via misused authorization checks		
EoP via buggy authorization checks		Centralizing such checks makes consistency, correctness easier
EoP via data tampering	Modify bits on disk	

Case Study (Appendix E)

Acme's Operational Network (Reading assignment)

Q1. What are you building?

Q2. What can go wrong? (**use STRIDE– Chapter 3**)

Q3. What should you do about those things that can go wrong? (**use STRIDE– Chapter 3**)

Q4. Did you do a decent job of analysis?

In summary, Acme has used STRIDE threat modeling and a model of their operational network to identify many threats. Again, they have moved from a vague sense of unease to a well justified set of concerns, which they can work through. From here, they'd need to decide on a prioritization scheme for those concerns, or consider additional security requirements, depending on their unique needs.

References

- Threat Modeling
  - Chapter 3: STRIDE