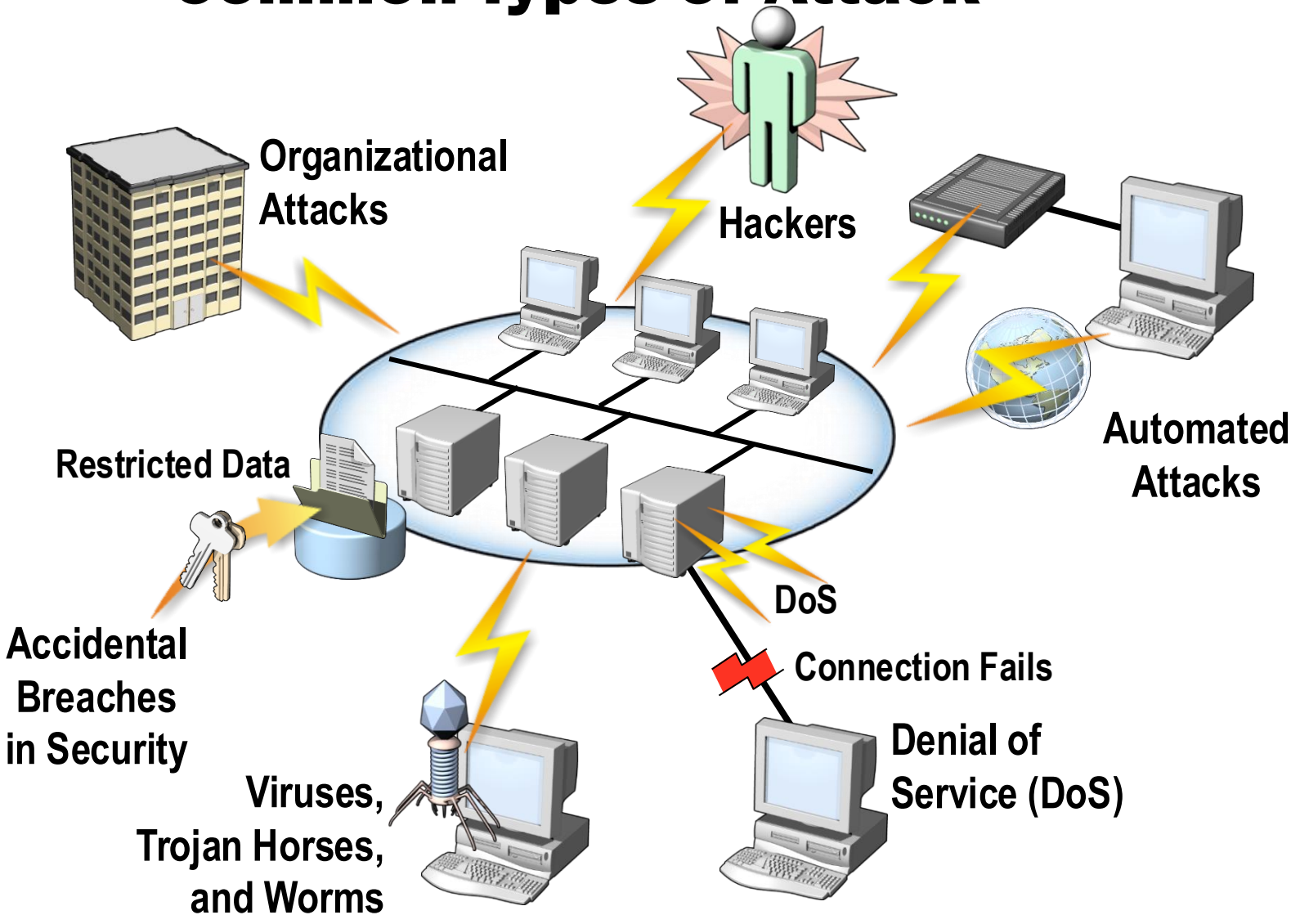
Threat and Vulnerability Management

CYS403

Intended Learning Outcomes

- CLO 1: Evaluate the issues and concerns indicative to protecting information, systems and users.

Common Types of Attack



Introduction

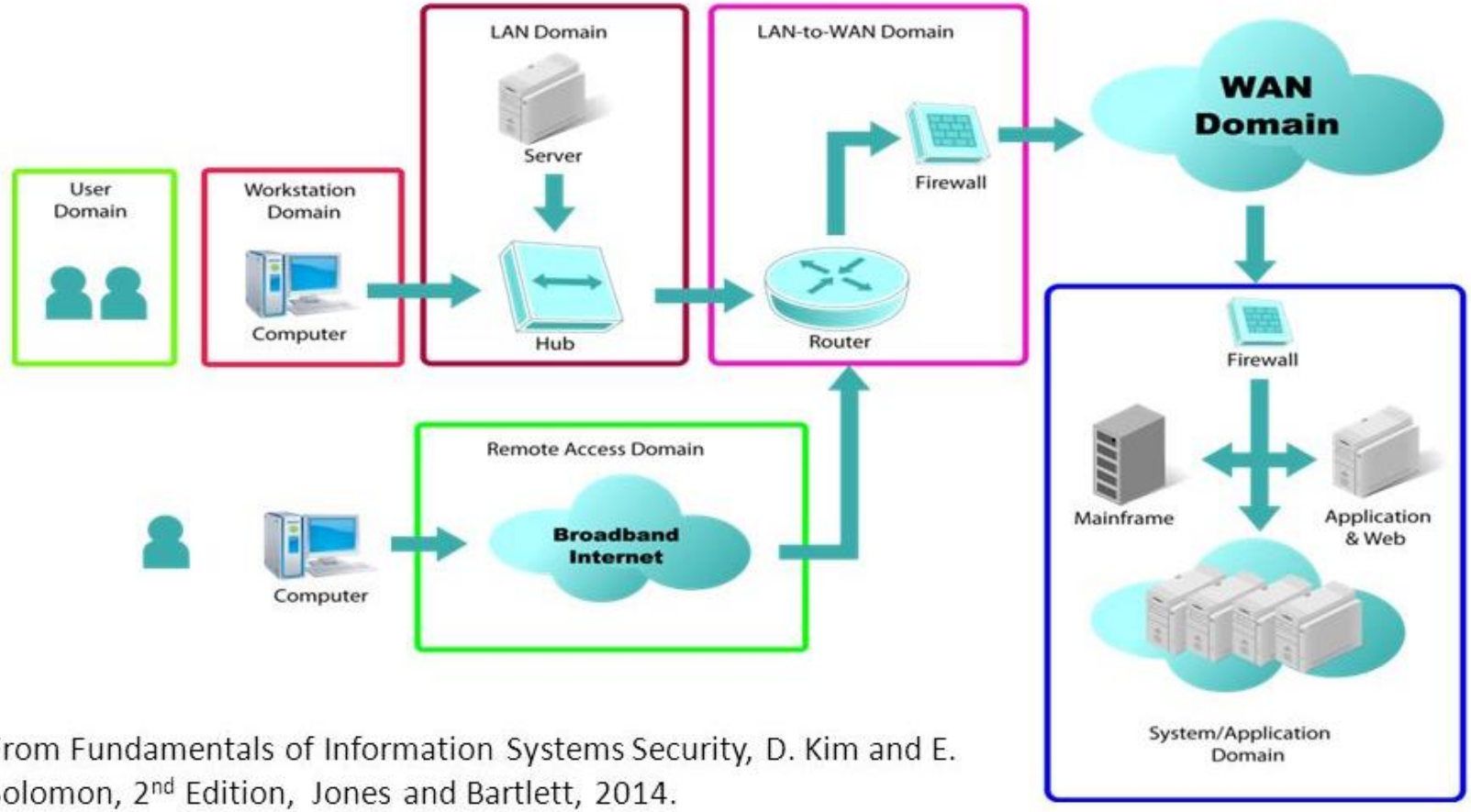
- a key step when managing risks is to first understand and manage the source.
  - This includes threats and vulnerabilities, and especially **threat /vulnerability pairs**.
- Once you understand these elements, it's much easier to identify mitigation techniques.

Introduction

- threat is referred as any activity that represents a possible danger.
- This includes any circumstances or events with the potential to adversely impact confidentiality, integrity, or availability of a business's assets.

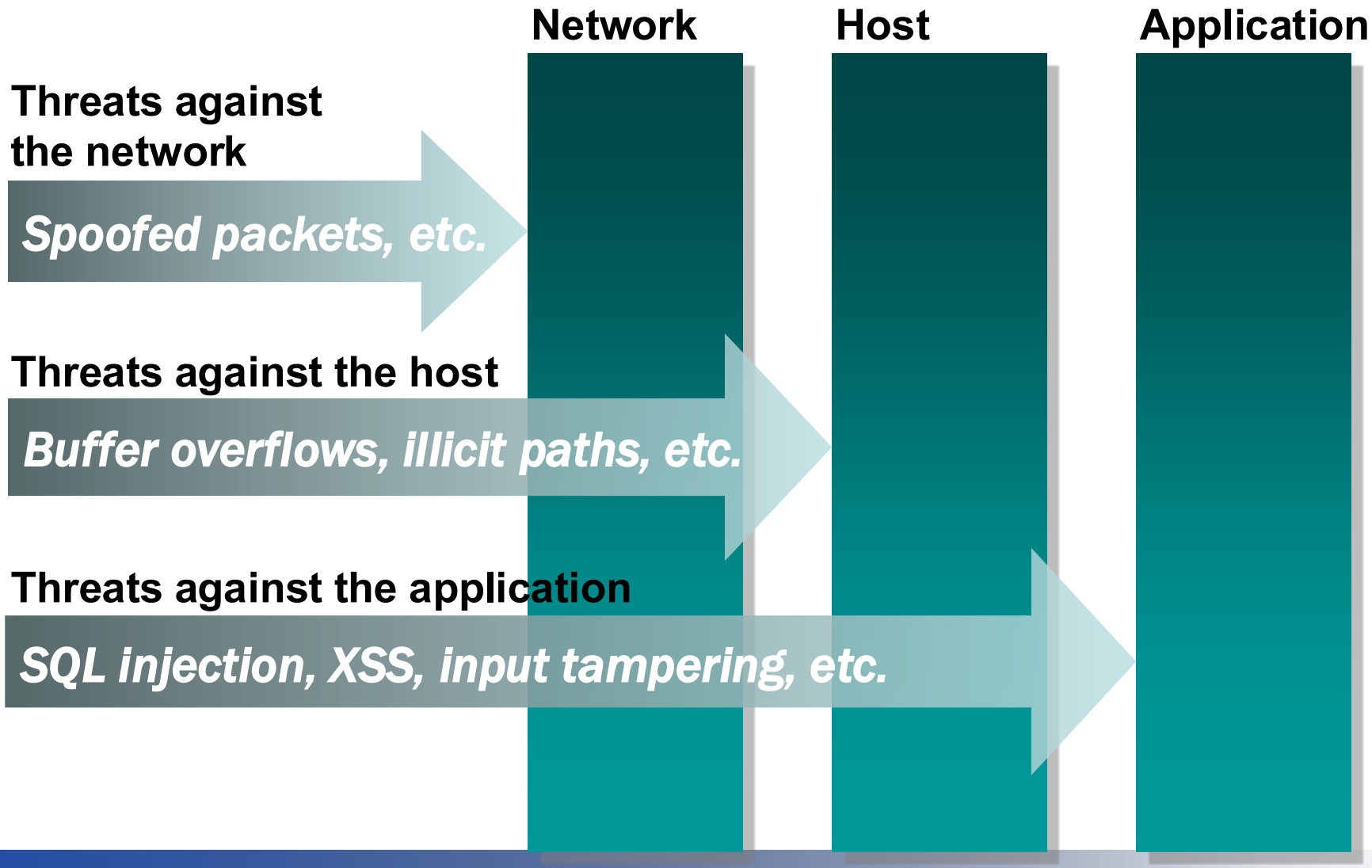
Possible threat sources

7 IT infra. Domains



From Fundamentals of Information Systems Security, D. Kim and E. Solomon, 2nd Edition, Jones and Bartlett, 2014.

Types of Threats



Threats Against the Network

Threat	Examples
Information gathering	Port scanning
	Using trace routing to detect network topologies
	Using broadcast requests to enumerate subnet hosts
Eavesdropping	Using packet sniffers to steal passwords
Denial of service (DoS)	SYN floods
	ICMP echo request floods
	Malformed packets
Spoofing	Packets with spoofed source addresses

http://msdn.microsoft.com/library/en-us/dnnetsec/html/THCMCh15.asp?frame=true#c15618429_004

Threats Against the Host

Threat	Examples
Arbitrary code execution	Buffer overflows in ISAPI DLLs (e.g., MS01-033)
	Directory traversal attacks (MS00-078)
File disclosure	Malformed HTR requests (MS01-031)
	Virtualized UNC share vulnerability (MS00-019)
Denial of service (DoS)	Malformed SMTP requests (MS02-012)
	Malformed WebDAV requests (MS01-016)
	Malformed URLs (MS01-012)
	Brute-force file uploads
Unauthorized access	Resources with insufficiently restrictive ACLs
	Spoofing with stolen login credentials
Exploitation of open ports and protocols	Using NetBIOS and SMB to enumerate hosts
	Connecting remotely to SQL Server

Threats Against the Application

Threat	Examples
SQL injection	Including a DROP TABLE command in text typed into an input field
Cross-site scripting	Using malicious client-side script to steal cookies
Hidden-field tampering	Maliciously changing the value of a hidden field
Eavesdropping	Using a packet sniffer to steal passwords and cookies from traffic on unencrypted connections
Session hijacking	Using a stolen session ID cookie to access someone else's session state
Identity spoofing	Using a stolen forms authentication cookie to pose as another user
Information disclosure	Allowing client to see a stack trace when an unhandled exception occurs

Components of a Threat

- **Threat agents**—criminals, terrorists, subversive or secret groups, state sponsored, disgruntled employees,, hackers, pressure groups, commercial groups.
- **Threat agent motivators**—political, secular, personal gain, religion, power, terrorism, curiosity
- **Capability**—software, technology, facilities, education and training, methods, books and manuals.

- **Threat catalysts**—events, technology changes, personal circumstances
- **Threat amplifiers**—peer pressure, fame, access to information, changing high technology, deskilling through scripting, skills and education levels, law enforcement activity, target vulnerability, target profile, public perception, peer perception
- **Threat inhibitors**—fear of capture, fear of failure, level of technical difficulty, cost of participation, sensitivity to public perception, law enforcement activity, target vulnerability, target profile, public perception, peer perception

Threat Agents

- Natural—fire, floods, power failure, earthquakes, etc.
- Unintentional—insider, outsider—primarily non-hostile
- Intentional—Insider, outsider—hostile or non-hostile (curious)
  - Foreign agents, industrial espionage, terrorists, organized crime, hackers and crackers, insiders, political dissidents, vendors and suppliers

Top ten Database Security Threats

www.schell.com/Top_Ten_Database_Threats.pdf

1. **Excessive Privilege Abuse**---users are granted database access privileges that exceed the requirements of their job function; e.g., a university administrator whose job requires only the ability to change student contact information may take advantage of excessive database update privileges to change grades
2. **Legitimate Privilege Abuse** ---- Users may abuse legitimate database privileges for unauthorized purposes; e.g. a rogue health worker who is willing to trade patient records for money
3. **Privilege Elevation**---Attackers may take advantage of database platform software vulnerabilities to convert access privileges from those of an ordinary user to those of an administrator. Vulnerabilities may be found in stored procedures, built-in functions, protocol implementations, and even SQL statements
4. **Database Platform Vulnerabilities**--- Vulnerabilities in underlying operating systems (Windows 2000, UNIX, etc.) and additional services installed on a database server may lead to unauthorized access, data corruption, or denial of service.
5. **SQL Injection**--- a perpetrator typically inserts (or "injects") unauthorized database statements into a vulnerable SQL data channel. Using SQL injection, attackers may gain unrestricted access to an entire database
6. **Weak Audit Trail**--- Weak database audit policy represents a serious organizational risk on many levels.--- regulatory risk, deterrence, and detection and recovery
7. **Denial of Service (DoS)---** access to network applications or data is denied to intended users
8. **Database Communication Protocol Vulnerabilities---** e.g., Four out of seven security fixes in the two most recent IBM DB2 FixPacks address protocol vulnerabilities; similarly, 11 out of 23 database vulnerabilities fixed in the most recent Oracle quarterly patch relate to protocols
9. **Weak Authentication**--- allowing attackers to assume the identity of legitimate database users by stealing or otherwise obtaining login credentials
10. **Backup Data Exposure**--- Backup database storage media is often completely unprotected from attack. As a result, several high profile security breaches have involved theft of database backup tapes and hard disks.

Ten web threats

http://www.darkreading.com/vulnerability/10-web-threats-that-could-harm-your-busi/240150315

1. **Bigger, Subtler DDoS Attacks**---Distributed Denial of Service Attacks
2. **Old Browsers, Vulnerable Plug-Ins**---e.g., browser vulnerabilities and, more frequently, the browser plug-ins that handle Oracle's Java and Adobe's Flash and Reader.
3. **Good Sites Hosting Bad Content**---in VOHO watering hole attack, attackers infected legitimate financial and tech industry websites in Massachusetts and Washington, D.C., commonly accessed by their intended victims
4. **Mobile Apps And The Unsecured Web**--- bring-your-own-device movement has led to a surge in consumer-owned devices inside corporate firewalls
5. **Failing To Clean Up Bad Input**---e.g., Since 2010, SQL injection has held the top spot on the Open Web Application Security Project's list of top 10 security vulnerabilities
6. **The Hazards Of Digital Certificates**--- a series of hacks against certificate authorities gave attackers the tools they needed to issue fraudulent SSL certificates that could disguise a malicious website as a legitimate
7. **The Cross-Site Scripting Problem**--- An attacker going after a banking site with a cross-site scripting vulnerability could run a script for a login box on the bank's page and steal users' credentials.
8. **The Insecure 'Internet Of Things'**--- Routers and printers, videoconferencing systems, door locks and other devices are now networked via Internet protocols and even have embedded Web servers. In many cases, the software on these devices is an older version of an open source library.
9. **Getting In The Front Door**--- Automated Web bots scrape from Web pages information that can give a competitor better intelligence on your business.
10. **New Technology, Same Problems**--- People click links all day long -- people are pretty trained to think that clicking a link on the Web is safe.

Major Security Threats on Information Systems http://ubiquity.acm.org/article.cfm?id=1117695

- **Intrusion or Hacking**---gaining access to a computer system without the knowledge of its owner---Tools: . Poor Implementation of Shopping Carts, Hidden fields in the html forms, Client-side validation scripts, Direct SQL attack, Session Hijacking, Buffer Overflow Forms, Port Scan
- **Viruses and Worms**--- programs that make computer systems not to work properly---Polymorphic Virus, Stealth Virus, Tunneling Virus, Virus Droppers, Cavity Virus
- **Trojan Horse**--- These programs are having two components; one runs as a server and another one runs as a client; data integrity attack, steal private information on the target system, store key strokes and make it viewable for hackers, sending private local as an email attachment.
- **Spoofing**---fooling other computer users to think that the source of their information is coming from a legitimate user---IP Spoofing, DNS Spoofing, ARP Spoofing
- **Sniffing**---used by hackers for scanning login_ids and passwords over the wires. TCPDUmp and Snoop are better examples for sniffing tools.
- **Denial of Service**---The main aim of this attack is to bring down the targeted network and make it to deny the service for legitimate users. In order to do DoS attacks, people do not need to be an expert. They can do this attack with simple ping command

Vulnerabilities

- "Some weakness of a system that could allow security to be allowed."
- Types of vulnerabilities
  - Physical vulnerabilities
  - Natural vulnerabilities
  - Hardware/software vulnerabilities
  - Media vulnerabilities (e.g., stolen/damaged disk/tapes)
  - Emanation vulnerabilities—due to radiation
  - Communication vulnerabilities
  - Human vulnerabilities

How do the vulnerabilities manifest?

- The different types of vulnerabilities manifest themselves via several misuses:
  - **External misuse**—visual spying, misrepresenting, physical scavenging
  - **Hardware misuse**—logical scavenging, eavesdropping, interference, physical attack, physical removal
  - **Masquerading**—impersonation, piggybacking attack, spoofing attacks, network weaving
  - **Pest programs**—Trojan horse attacks, logic bombs, malevolent worms, virus attacks
  - **Bypasses**—Trapdoor attacks, authorization attacks (e.g., password cracking)
  - **Active misuse**—basic active attack, incremental attack, denial of service
  - **Passive misuse**—browsing, interference, aggregation, covert channels

Examples of Information Security Vulnerabilities

Ref: http://simplicable.com/new/the-big-list-of-information-security-vulnerabilities

Information security vulnerabilities are weaknesses that expose an organization to risk.

- **Through employees**: Social interaction, Customer interaction, Discussing work in public locations, Taking data out of the office (paper, mobile phones, laptops), Emailing documents and data, Mailing and faxing documents, Installing unauthorized software and apps, Removing or disabling security tools, Letting unauthorized persons into the office (tailgating) , Opening spam emails, Connecting personal devices to company networks, Writing down passwords and sensitive data, Losing security devices such as id cards, Lack of information security awareness, Keying data
- **Through former employees**---Former employees working for competitors, Former employees retaining company data, Former employees discussing company matters
- **Through Technology**---Social networking, File sharing, Rapid technological changes, Legacy systems, Storing data on mobile devices such as mobile phones, Internet browsers
- **Through hardware**---. Susceptibility to dust, heat and humidity, Hardware design flaws, Out of date hardware, Misconfiguration of hardware

Examples of Information Security Vulnerabilities (Cont.)

- **Through software**---Insufficient testing, Lack of audit trail, Software bugs and design faults, Unchecked user input, Software that fails to consider human factors, Software complexity (bloatware), Software as a service (relinquishing control of data), Software vendors that go out of business or change ownership
- **Through Network**---Unprotected network communications, Open physical connections, IPs and ports, Insecure network architecture, Unused user ids, Excessive privileges, Unnecessary jobs and scripts executing , Wifi networks
- **Through IT Management**---Insufficient IT capacity , Missed security patches, Insufficient incident and problem management, Configuration errors and missed security notices , System operation errors, Lack of regular audits, Improper waste disposal, Insufficient change management, Business process flaws, Inadequate business rules, Inadequate business controls, Processes that fail to consider human factors, Overconfidence in security audits, Lack of risk analysis, Rapid business change, Inadequate continuity planning Lax recruiting processes
- **Partners and suppliers**---Disruption of telecom services, Disruption of utility services such as electric, gas, water, Hardware failure, Software failure, Lost mail and courier packages, Supply disruptions, Sharing confidential data with partners and suppliers

Common Vulnerabilities and Exposures (CVE) List

- CVE defines a vulnerability as:
- "A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety)."
- All vulnerabilities in the NVD have been assigned a CVE identifier and thus, abide by this definition.A CVE entry contains:
  - CVE ID
  - Description
  - References

Common Vulnerabilities and Exposures (CVE) List

CVE-ID	
CVE-2016-3191	Learn more at National Vulnerability Database (NVD) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description

The compile_branch function in pcre_compile.c in PCRE 8.x before 8.39 and pcre2_compile.c in PCRE2 before 10.22 mishandles patterns containing an (*ACCEP substring in conjunction with nested parentheses, which allows remote attackers to execute arbitrary code or cause a denial of service (stack-based buffer overflow) via a crafted regular expression, as demonstrated by a JavaScript RegExp object encountered by Konqueror, aka ZDI-CAN-3542.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CONFIRM:http://vcs.pcre.org/pcre2?view=revision&revision=489
- CONFIRM:http://vcs.pcre.org/pcre?view=revision&revision=1631
- CONFIRM:https://bugs.debian.org/815920
- CONFIRM:https://bugs.debian.org/815921
- CONFIRM:https://bugs.exim.org/show_bug.cgi?id=1791
- CONFIRM:https://bugzilla.redhat.com/show_bug.cgi?id=1311503
- CONFIRM:http://www.oracle.com/technetwork/topics/security/linuxbulletinapr2016-2952096.html

https://www.cvedetails.com/vulnerability-list/

Example on CVE

- CVE-2016-7280
  - **Cross-site scripting (XSS) vulnerability** in Microsoft Edge allows remote attackers to inject arbitrary web script or HTML via unspecified vectors, aka "Microsoft Edge Information Disclosure Vulnerability," a different vulnerability than CVE-2016-7206.
- CVE-2016-3273
  - The XSS Filter in Microsoft Internet Explorer 9 through 11 and Microsoft Edge does not properly restrict JavaScript code, which allows remote attackers to obtain sensitive information via a crafted web site, aka "Microsoft Browser Information Disclosure Vulnerability.“

Threat Modeling

- Structured approach to identifying, quantifying, and addressing threats
- Threat-modeling methods are used to create an abstraction of the system
  - profiles of potential attackers, including their goals and methods
  - a catalog of potential threats that may arise

12 threat models

- STRIDE and Associated Derivations
- PASTA
- LINDDUN
- CVSS
- Attack Trees
- Persona non Grata
- Security Cards
- hTMM
- Quantitative Threat Modeling Method
- Trike
- VAST Modeling
- OCTAVE

STRIDE and Associated Derivations

- STRIDE evaluates the system detail design. It models the in-place system.
- building **data-flow diagrams** (DFDs).
- STRIDE is **used to identify system entities, events, and the boundaries of the system.**
- Microsoft no longer maintains STRIDE, it is implemented as part of the **Microsoft Security Development Lifecycle** (SDL) with the **Threat Modeling Tool**

STRIDE and Associated Derivations

S **Spoofing**
Can an attacker gain access using a false identity?

T **Tampering**
Can an attacker modify data as it flows through the application?

R **Repudiation**
If an attacker denies doing something, can we prove he did it?

I **Information disclosure**
Can an attacker gain access to private or potentially injurious data?

D **Denial of service**
Can an attacker crash or reduce the availiability of the system?

E **Elevation of privilege**
Can an attacker assume the identity of a privileged user?

STRIDE and Associated Derivations

Threat	Definition	Property	Example
Spoofing	Pretend to be someone else.	Authentication	Hack victim's email and use to send messages in name of the victim.
Tampering	Change data or code.	Integrity	Software executive file is tampered by hackers.
Repudiation	Claiming not to do a particular action.	Non-repudiation	"I have not sent an email to Alice".
Information Disclosure	Leakage of sensitive information.	Confidentiality	Credit card information available on the internet.
Denial of Service	Non-availability of service	Availability	Web application not responding to user requests.
Elevation of privilege	Able to perform unauthorized action	Authorization	Normal user able to delete admin account

http://allabouttesting.org

PASTA

- The Process for Attack Simulation and Threat Analysis (PASTA)
- is a risk-centric threat-modeling framework developed in 2012.
- It contains seven stages, each with multiple activities
- PASTA aims to **bring business objectives and technical requirements together.**
- This method elevates the threat-modeling process to a strategic level by involving key decision makers and requiring security input from operations, governance, architecture, and development.

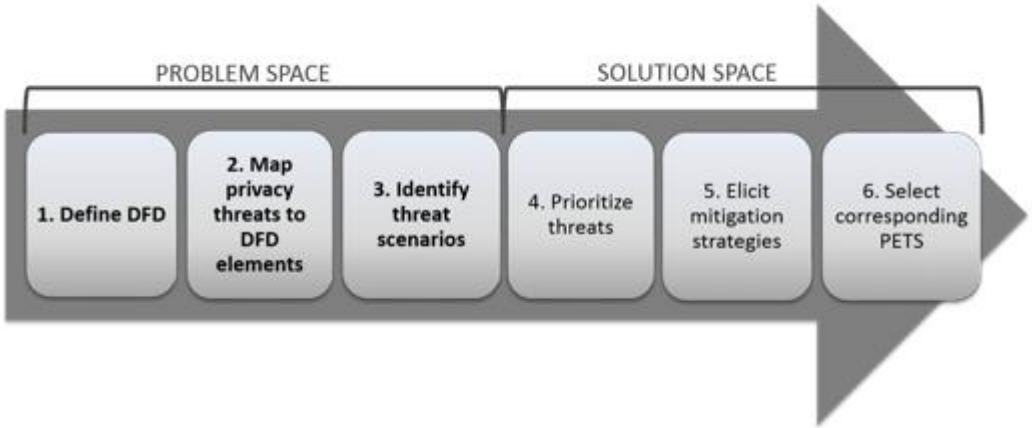
PASTA

1. Define Objectives	• Identify Business Objectives • Identify Security and Compliance Requirements • Business Impact Analysis
2. Define Technical Scope	• Capture the Boundaries of the Technical Environment • Capture Infrastructure | Application | Software Dependencies
3. Application Decomposition	• Identify Use Cases | Define App. Entry Points & Trust Levels • Identify Actors | Assets | Services | Roles | Data Sources • Data Flow Diagramming (DFDs) | Trust Boundaries
4. Threat Analysis	• Probabilistic Attack Scenarios Analysis • Regression Analysis on Security Events • Threat Intelligence Correlation and Analytics
5. Vulnerability & Weaknesses Analysis	• Queries of Existing Vulnerability Reports & Issues Tracking • Threat to Existing Vulnerability Mapping Using Threat Trees • Design Flaw Analysis Using Use and Abuse Cases • Scorings (CVSS/CWSS) | Enumerations (CWE/CVE)
6. Attack Modeling	• Attack Surface Analysis • Attack Tree Development | Attack Library Mgt. • Attack to Vulnerability & Exploit Analysis Using Attack Trees
7. Risk & Impact Analysis	• Qualify & Quantify Business Impact • Countermeasure Identification and Residual Risk Analysis • ID Risk Mitigation Strategies

LINDDUN

- **LINDDUN** (**L**inkability, **I**dentifiability, **N**onrepudiation, **D**etectability, **D**isclosure of information, **U**nawareness, **N**oncompliance)
- **focuses on privacy concerns** and can be used for data security. INDDUN provides a systematic approach to privacy assessment.
- LINDDUN starts with a DFD of the system that defines the system's data flows, data stores, processes, and external entities.
- Consisting of six steps.

Refer to https://linddun.org/linddun.php for more explanation

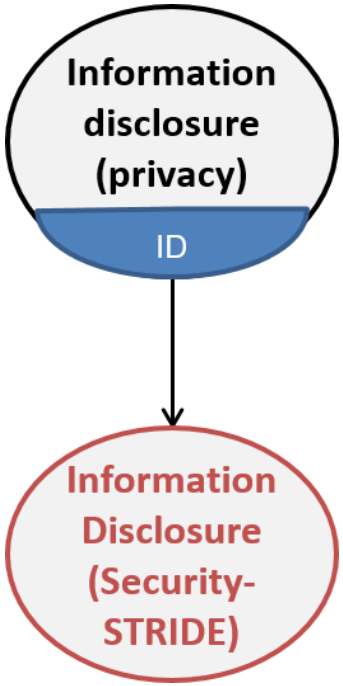


LINDDUN

- **LINDDUN**
  - **L**inkability:Being able to sufficiently distinguish whether 2 IOI (items of interest) are linked or not.
    - Examples are: anonymous letters written by the same person, web page visits by the same user, entries in two databases related to the same person, people related by a friendship link, etc.
  - **I**dentifiability: Being able to sufficiently identify the subject within a set of subjects
    - Examples are: identifying the reader of a web page, the sender of an email, the person to whom an entry in a database relates, etc.
  - Nonrepudiation:
    - Typical non-repudiation examples exist in the context of anonymous online voting systems, and whistleblowing systems where plausible deniability is required.

LINDDUN

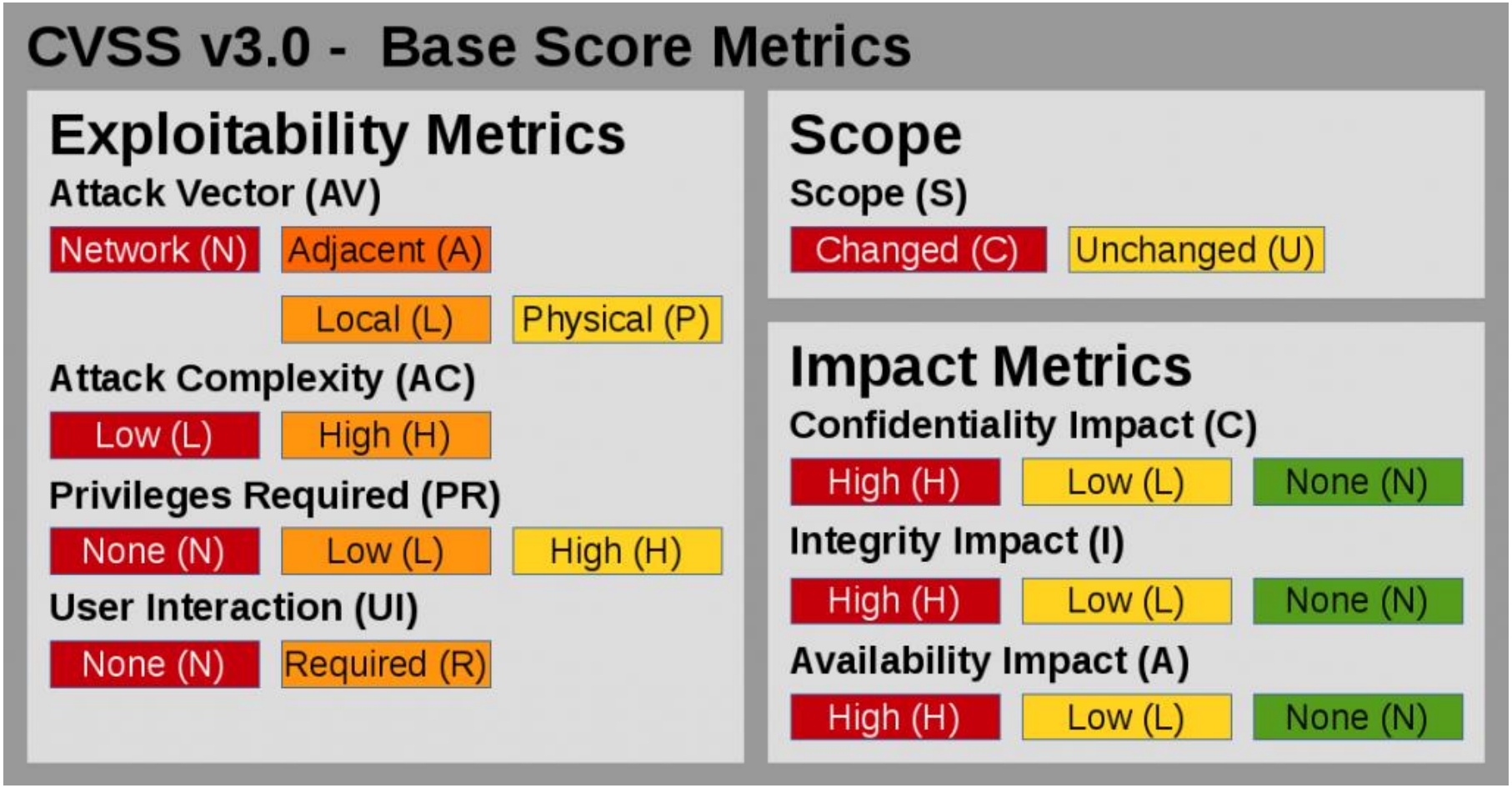
- **LINDDUN**
- **D**etectability: Being able to sufficiently distinguish whether an item of interest (IOI) exists or not.
  - **D**isclosure of information: a part of data privacy (STIRDE will be used here)
  - **U**nawareness: Being unaware of the consequences of sharing information.
- **N**oncompliance: Not being compliant with legislation, regulations, and corporate policies.



Common Vulnerability Scoring System (CVSS)

- **The Common Vulnerability Scoring System (CVSS)** captures the principal characteristics of a vulnerability and produces a numerical severity score.
- The CVSS provides users a common and standardized scoring system within different cyber and cyber-physical platforms.
- A CVSS score can be computed by a calculator that is **available online**.
- A CVSS score is derived from **values assigned by an analyst for each metric**.

Common Vulnerability Scoring System (CVSS)



Threats Analysis (Attack Tree)

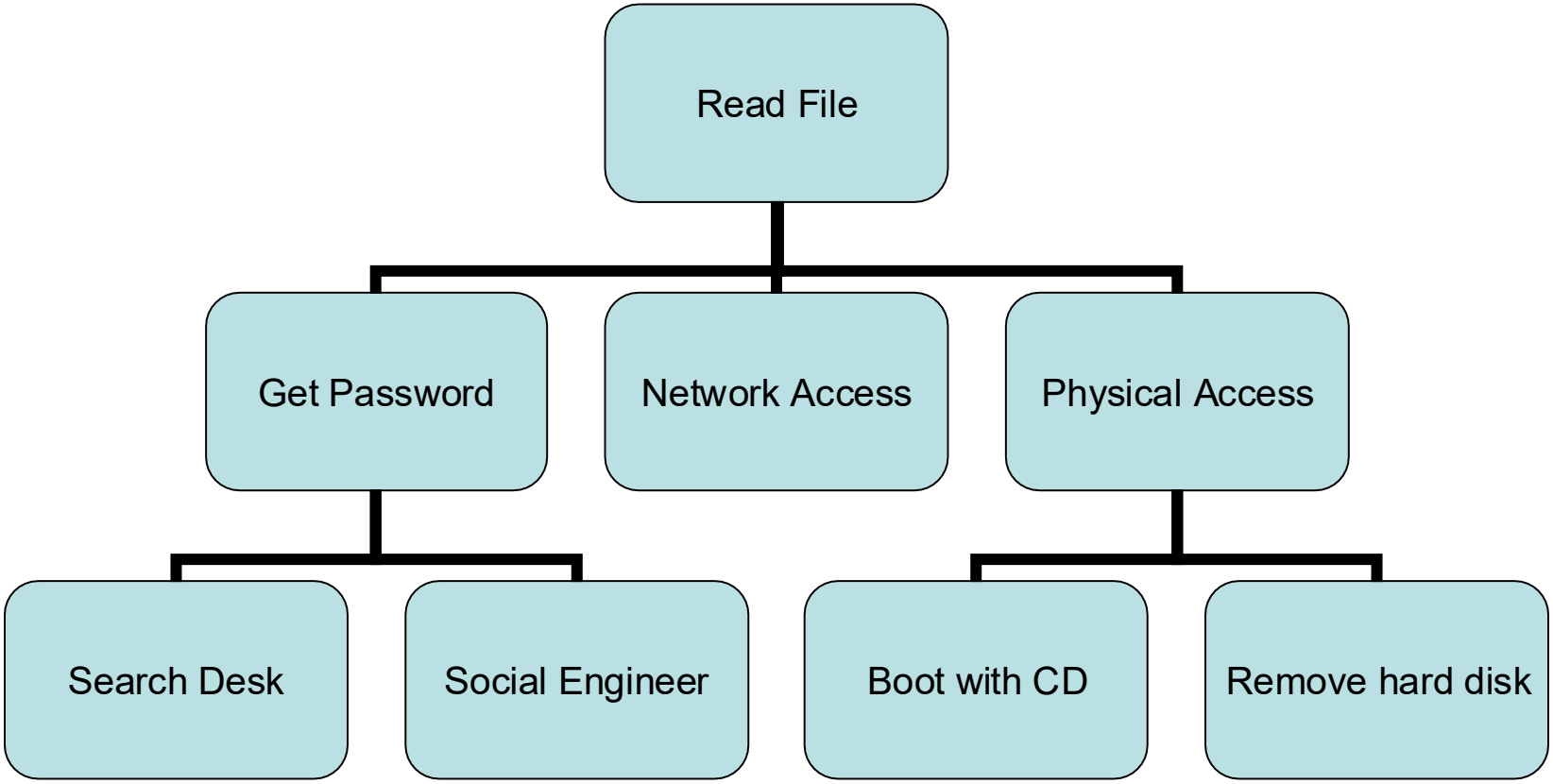
Decompose threats into individual, testable conditions using attack trees.

Attack Trees

- Hierarchical decomposition of a threat.
- Root of tree is adversary's goal in the attack.
- Each level below root decomposes the attack into finer approaches.
- Child nodes are ORed together by default.
- Special notes may indicate to AND them.

Attack Trees—Graph Notation

Goal: Read file from password-protected PC.

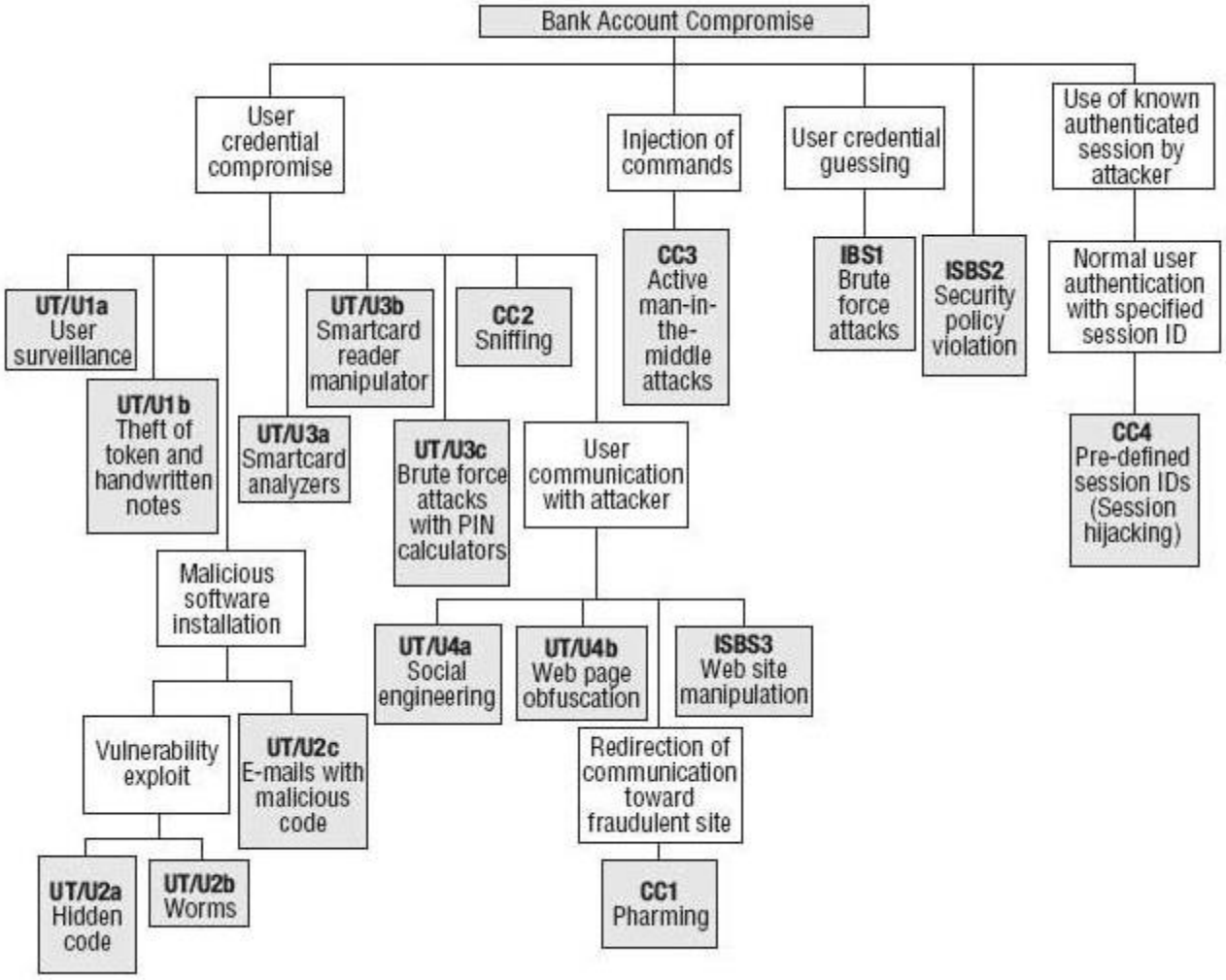


Attack Trees—Text Notation

Goal: Read message sent from one PC to another.

1. Convince sender to reveal message.
    - 1.1 Blackmail.
    - 1.2 Bribe.
2. Read message when entered on sender's PC.
    - 1.1 Visually monitor PC screen.
    - 1.2 Monitor EM radiation from screen.
3. Read message when stored on receiver's PC.
    - 1.1 Get physical access to hard drive.
    - 1.2 Infect user with spyware.
4. Read message in transit.
    - 1.1 Sniff network.
    - 1.2 Usurp control of mail server.

Attack Trees



Persona non Grata

- Persona non Grata (PnG) focuses on the motivations and skills of human attackers.
- Users act as models to try the possible misuse of the system and forces analysts to view the system from an unintended-use point of view.
- **PnG can help visualize threats from the counterpart side, which can be helpful in the early stages of the threat modeling.**
- The idea is to introduce a technical expert to a potential attacker of the system and examine the attacker's skills, motivations, and goals.
- This analysis helps the expert understand the system's vulnerabilities from the point of view of an attacker.
-

Security Cards

- **Security Cards identify unusual and complex attacks.**
- It is a kind of brainstorming technique. With help from a deck of cards, analysts can answer questions about an attack, such as
  - Who might attack?
  - Why might the system be attacked?
  - What assets are of interest?
  - How can these attacks be implemented?"

Human Impact	Adversary's Motivations	Adversary's Resources	Adversary's Methods
• the biosphere • emotional well-being • financial well-being • personal data • physical well-being • relationships • societal well-being • unusual impacts	• access or convenience • curiosity or boredom • desire or obsession • diplomacy or warfare • malice or revenge • money • politics • protection • religion • self-promotion • world view • unusual motivations	• expertise • a future world • impunity • inside capabilities • inside knowledge • money • power and influence • time • tools • unusual resources	• attack cover-up • indirect attack • manipulation or coercion • multi-phase attack • physical attack • processes • technological attack • unusual methods

hTMM

- The **Hybrid Threat Modeling Method (hTMM)** was developed by the Software Engineering Institute in 2018.
- It consists of a combination of *SQUARE (Security Quality Requirements Engineering Method), Security Cards, and PnG activities.*
- The targeted characteristics of the method include no false positives, no overlooked threats, a consistent result regardless of who is doing the threat modeling, and cost effectiveness.
- The main steps of the method are
  - Identify the system to be threat-modeled.
  - Apply Security Cards based on developer suggestions.
  - Remove unlikely PnGs (i.e., there are no realistic attack vectors).
  - Summarize the results using tool support.
  - Continue with a formal risk-assessment method.

Quantitative Threat Modeling Method

- This hybrid method consists of *attack trees, STRIDE, and CVSS* methods applied in synergy.
- It aims to address a few urgent issues with threat modeling for cyber-physical systems that had complex interdependences among their components.
- The first step of the **Quantitative Threat Modeling Method (Quantitative TMM)** is to build component attack trees for the five threat categories of STRIDE.
  - To show the dependencies among attack categories and low-level component attributes.
- After that, the CVSS method is applied and scores are calculated for the components in the tree.

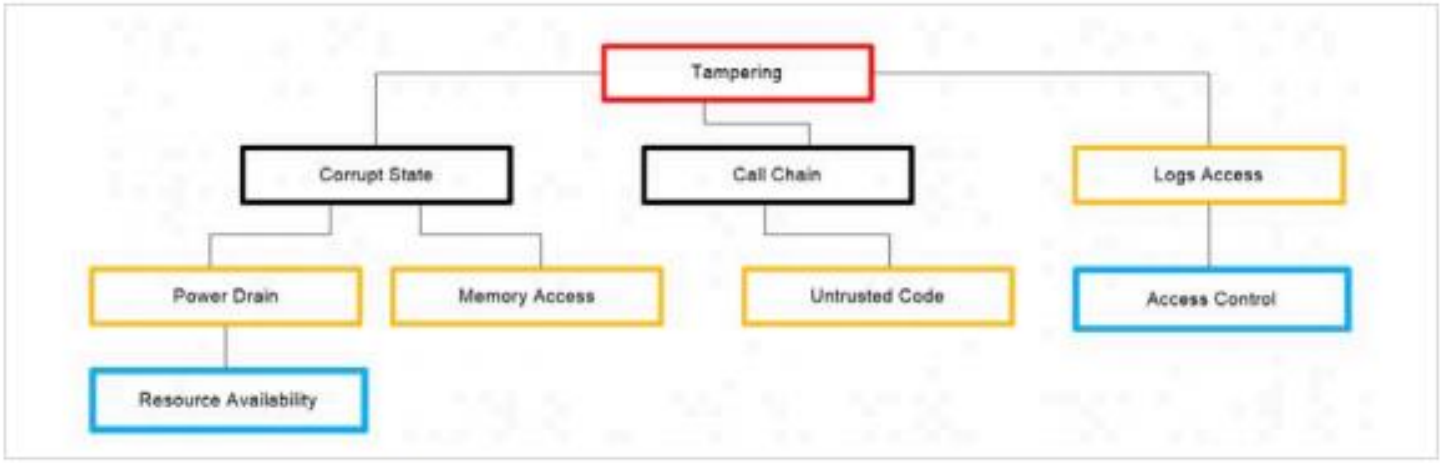


Figure 9: Component Attack Tree [3]

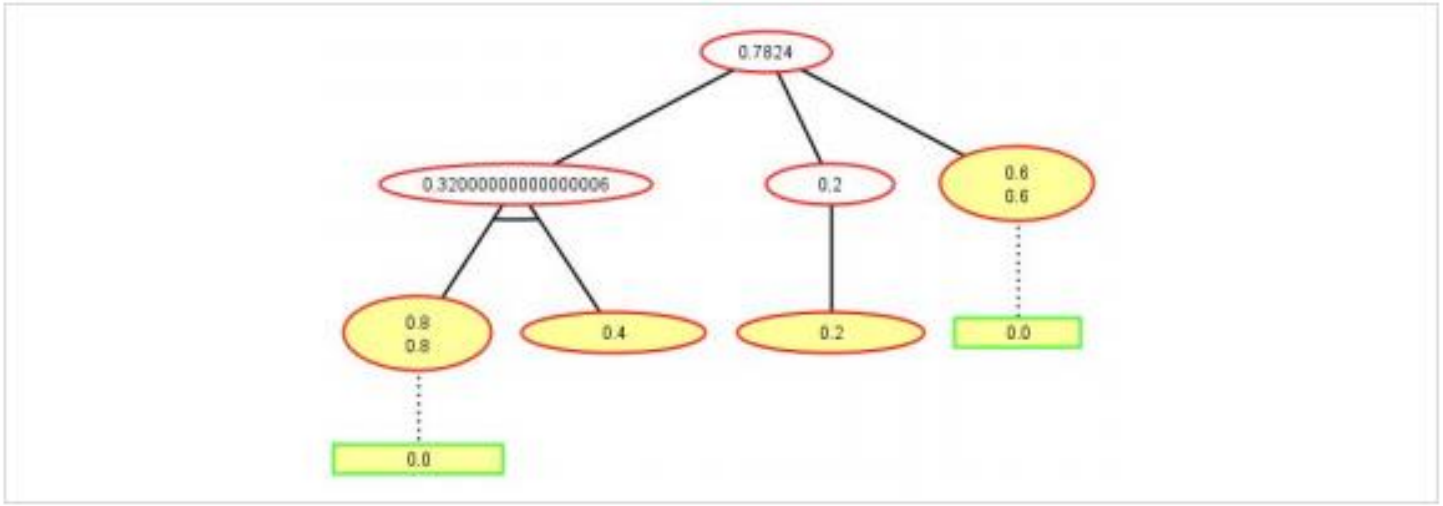


Figure 10: CVSS Scoring for Tampering Attack Tree [3]

Trike

- **Trike** was created as a **security audit framework** that uses threat modeling as a technique.
- It looks at threat modeling from a risk-management and defensive perspective.
- Trike steps
  - Define the system: system's actors, assets, intended actions, and rules.
  - actor-asset-action matrix:.
  - Each cell has four parts (CRUD), and associated rules.
  - data flow diagram (DFD) is built. Each element is mapped to a selection of actors and assets. Iterating through the DFD.
  - the analyst identifies threats, which fall into one of two categories: elevations of privilege or denials of service.
  - Each discovered threat becomes a root node in an attack tree.
  - To assess the risk of attacks that may affect assets through CRUD,
  - Trike uses a five-point scale for each action, based on its probability.

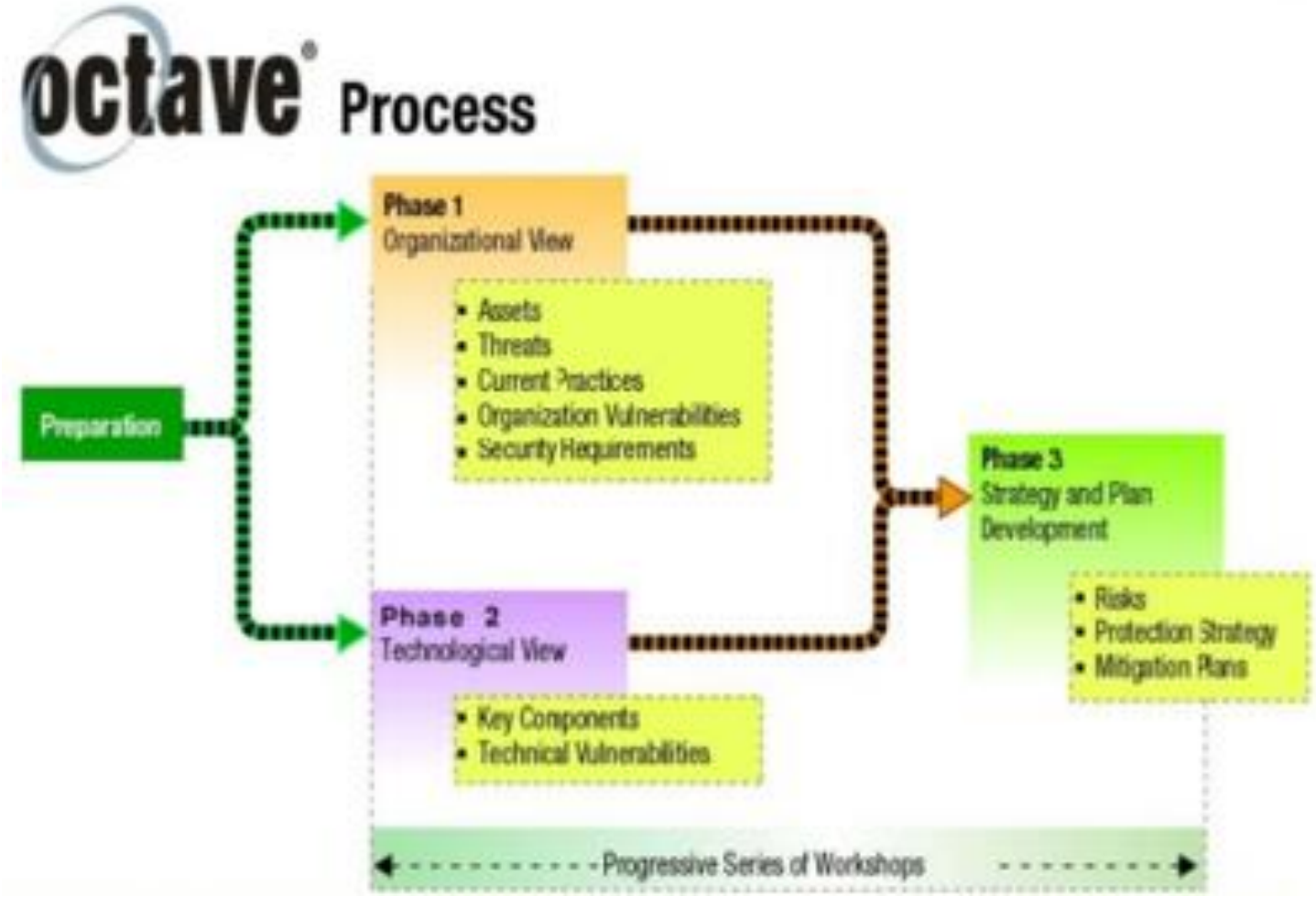
Visual, Agile, and Simple Threat modeling (VAST)

- The founding principle is that, in order to be effective, threat modeling must scale across
  - the infrastructure and entire DevOps portfolio, integrate seamlessly into an Agile environment and
  - provide actionable, accurate, and consistent outputs for developers, security teams, and senior executives alike.
- **VAST: Application Threat Models :** For application developer to investigate DFD & Process flow
  - map the features and communications of an application in much the same way as developers and architects think about the application during an SDLC design session.
- **VAST: Operational Threat Models:** for infrastructure team to investigate the running of operation and the data flow information is presented from an attacker

OCTAVE

- The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) method is a risk-based strategic assessment and planning method for cybersecurity.
- OCTAVE focuses on assessing organizational risks and does not address technological risks.
- Its main aspects are operational risk, security practices, and technology.
- OCTAVE has three phases.
  - Build asset-based threat profiles. (This is an organizational evaluation.)
  - Identify infrastructure vulnerability. (This is an evaluation of the information infrastructure.)
  - Develop a security strategy and plans. (This is an identification of risks to the organization's critical assets and decision making.)

octave® Process



DREAD

D **Damage potential**
What are the consequences of a successful exploit?

R **Reproducibility**
Would an exploit work every time or only under certain circumstances?

E **Exploitability**
How skilled must an attacker be to exploit the vulnerability?

A **Affected users**
How many users would be affected by a successful exploit?

D **Discoverability**
How likely is it that an attacker will know the vulnerability exists?

Example

Threat	D	R	E	A	D	Sum
Auth cookie theft (eavesdropping)	3	2	3	2	3	13
Auth cookie theft (XSS)	3	2	2	2	3	12

Potential for damage is high (spoofed identities, etc.)

Cookie can be stolen any time, but is only useful until expired

Anybody can run a packet sniffer; XSS attacks require moderate skill

All users could be affected, but in reality most won't click malicious links

Easy to discover: just type a <script> block into a field

Threat Modeling Method	Features
STRIDE	• Helps identify relevant mitigating techniques • Is the most mature • Is easy to use but is time consuming
PASTA	• Helps identify relevant mitigating techniques • Directly contributes to risk management • Encourages collaboration among stakeholders • Contains built-in prioritization of threat mitigation • Is laborious but has rich documentation
LINDDUN	• Helps identify relevant mitigation techniques • Contains built-in prioritization of threat mitigation • Can be labor intensive and time consuming
CVSS	• Contains built-in prioritization of threat mitigation • Has consistent results when repeated • Has automated components • Has score calculations that are not transparent
Attack Trees	• Helps identify relevant mitigation techniques • Has consistent results when repeated • Is easy to use if you already have a thorough understanding of the system
Persona non Grata	• Helps identify relevant mitigation techniques • Directly contributes to risk management • Has consistent results when repeated • Tends to detect only some subsets of threats
Security Cards	• Encourages collaboration among stakeholders • Targets out-of-the-ordinary threats • Leads to many false positives
hTMM	• Contains built-in prioritization of threat mitigation • Encourages collaboration among stakeholders • Has consistent results when repeated
Quantitative TMM	• Contains built-in prioritization of threat mitigation • Has automated components • Has consistent results when repeated
Trike	• Helps identify relevant mitigation techniques • Directly contributes to risk management • Contains built-in prioritization of threat mitigation • Encourages collaboration among stakeholders • Has automated components • Has vague, insufficient documentation
VAST Modeling	• Helps identify relevant mitigation techniques • Directly contributes to risk management • Contains built-in prioritization of threat mitigation • Encourages collaboration among stakeholders • Has consistent results when repeated • Has automated components • Is explicitly designed to be scalable • Has little publicly available documentation
OCTAVE	• Helps identify relevant mitigation techniques • Directly contributes to risk management • Contains built-in prioritization of threat mitigation • Encourages collaboration among stakeholders • Has consistent results when repeated • Is explicitly designed to be scalable • Is time consuming and has vague documentation

Threat/Vulnerability Pairs

- A **threat/vulnerability pair** occurs when a threat exploits a vulnerability.
- The vulnerabilities provide a path for the threat that results in a harmful event or a loss.
- It's important to know that both the threat and the vulnerability must come together to result in a loss.
- Vulnerabilities depend on your organization. For example, if you're hosting public facing servers, the servers have several potential weaknesses. However, if you don't have any public-facing servers, there aren't any vulnerabilities for the organization in this area.
- Thus, the risk is zero.

Threat Matrix

- Capabilities of a threat versus type of vulnerabilities
- Similar to risk assessment or risk analysis matrix

TABLE 2-1 Examples of threat/vulnerability pairs and potential losses.

THREAT	VULNERABILITY	HARMFUL EVENT OR LOSS
Fire	Lack of fire detection and suppression equipment	Can be total loss of business
Hurricane, earthquake, tornado	Location	Can be total loss of business
Malware	Lack of antivirus software Outdated definitions	Infection (impact of loss determined by payload of malware)
Equipment failure	Data not backed up	Loss of data availability (impact of loss determined by value of data)
Stolen data	Access controls not properly implemented	Loss of confidentiality of data
Denial of service (DoS) or distributed denial of service (DDoS) attack	Public-facing servers not protected with firewalls and intrusion detection systems	Loss of service availability
Users	Lack of access controls	Loss of confidentiality
Social engineer	Lack of security awareness	Loss depends on the goals and success of attacker