CS 419: Computer Security

Week 1 Recitation: Key Concepts

TA: Tao Zhang

Recitations

TA info:

Wed 12:25 – 1:20 pm

Tao Zhang

tz347@cs.rutgers.edu

Recitations will cover:

- Discussion of additional course topics
- Review of course content & and exam review
- Homework/project information

Key Concepts From Lecture 1

The CIA Triad

Triad = group of three things

Model for thinking about computer security

1. **Confidentiality**
2. **Integrity**
3. **Availability**

- Restrict access to data and resources (computing, data, network)
- Access is defined by a policy
- Requires
  - **Identification**: who is the user (or computer or application)?
  - **Authentication**: verify the user (or computer or application)
  - **Authorization**: check the policy to see if the user has access to the resource
- Implemented via **access control mechanisms** or **cryptography**

Authentication establishes the integrity of the user
(is it really the user?)

The CIA Triad

Model for thinking about computer security

1. **Confidentiality**
2. **Integrity**
3. **Availability**

- Establish trustworthiness of data, users, and resources
- Detect tampering
- Validate (authenticate) the identity of users/systems/services
- **Implemented via authentication algorithms and/or cryptography (hash functions, message authentication codes, digital signatures)**

The CIA Triad

Model for thinking about computer security

1. **Confidentiality**
2. **Integrity**
3. **Availability**

- Ensure data and resources are accessible and perform adequately
- Includes recovery from failure
- **Implemented via OS resource management, OS thread scheduler, firewalls, load balancers, replication & backups**

A Denial of Service attack affects availability

Security Engineering

Combination of

1. **Policy** (rules)
2. **Mechanisms** (implementation)
3. **Assurance** (integrity of the mechanisms and policy)
4. **Incentives** (the human factor)

Engineering = not just designing the system
but also understanding and taking into account the trade-offs
(time, money, complexity, features, …)

(1, 2) Policies & Mechanisms

Security Policy: ***the rules defining what is and is not allowed***

Security Mechanisms: **the *implementation enforcing the policy***

- **Mechanisms can be *procedural* or *technical***

  For example:

  - **Procedural**: inspect a student's ID card when they submit an exam
  - **Technical**: an operating system enforces read restrictions to prevent a student from copying another's assignment

- **We assume that a security policy is correct and unambiguous and that mechanisms function correctly**
  - Otherwise, we cannot assure the security of a system

(3) Assurance & Trustworthy components

- **Assurance = how much we can trust a system**
  - This covers the specifications, design, and implementation
  - Most systems are too complex to verify formally
    - Auditing: inspecting the code for possible security-critical bugs
    - Testing to make sure the system works as expected (**functional testing**)
    - Testing to make sure the system is resilient to attacks (**penetration testing**)
- **Trustworthy components: components that follow security policies correctly**
  - We trust that they will not have undocumented APIs, special flags, etc. to allow attackers to get around security policies

Trusted Computing Base (TCB)

Trusted Computing Base (TCB) =
The entire set of components (hardware, firmware, software) that are critical to the security of a system

- **Processors**
- **Boot loader**
- **Operating system**
- **Device drivers**
- **Compilers**
- **Libraries**
- **Network services**
- **Routers**

The **TCB** contains the mechanisms that implement and enforce the security of a system.

Any vulnerabilities in the TCB can affect the security of the entire system

If the TCB is compromised, we can no longer guarantee the security of a system

Incentives ⇒ Human factors

Security is as weak as its weakest link

People are often the weakest link in security

- Users may **masquerade** as another person if you can steal or guess access credentials
- **Corrupt insiders** are considered trusted and can get through most security measures
- Untrained people can make **honest mistakes** that compromise security
  - This includes untrained system administrators (bad configurations), poor programmers (poor implementations), and employees that abuse privileges (bad actions)

- **Social engineering is a powerful tactic**

  People tend to be trusting and usually try to help … that can backfire:
  - Convince someone to give you their credentials or access to the data you need
    - Get their ID/password, convince someone to let you into the building, impersonate yourself as someone else and they believe you ("*I work in IT and need your info*"), …

Security is an Investment

- **Security costs money**
  - Physical security: locks, guards, cameras, barbed wire, …
  - Computer security: skilled employees, specifications, design, implementation, penetration testing, protocol validation, etc.
- **Security provides no reward**
  - Designing a secure system costs extra money and takes more time
  - It does not make a system faster, easier to use, or function better
- **This impacts the business decision of how much effort to put into security**
  - People don't see the benefits directly – so security often does not get much attention
  - What is the value of the possible loss of reputation, money, intellectual property, etc. if there is a security breach?

Vulnerabilities: Terms & Examples

What is a Vulnerability?

Vulnerability:

A flaw or weakness in hardware, software, or processes that can be exploited to compromise security.

Examples

- **Software Bug**: Buffer overflow in an application.
- **Misconfiguration**: Default admin credentials left unchanged.
- **Design Flaw**: Lack of input validation.
- **Weak Encryption**: Using an outdated encryption algorithm

Vulnerabilities are the starting point for many cyberattacks

What is an Exploit?

Exploit:

A piece of software, data, or sequence of commands that takes advantage of a vulnerability to perform unauthorized actions.

Examples

- **Metasploit**: A penetration testing framework with pre-built exploits
- **Zero-Day Exploit**: An exploit targeting a vulnerability before it is publicly known
- **SQL Injection**: Taking advantage of a vulnerability in a database query (we will study these later)

Exploits are tools attackers use to leverage vulnerabilities

What is an Attack?

Attack:

The execution of an exploit to compromise confidentiality, integrity, or availability (CIA Triad) — it is an exploit in action

Examples

- **Phishing Attack**: Tricking users into revealing credentials
- **Distributed Denial of Service (DDoS) Attack**:
  Overwhelming a server with traffic to make it unavailable
- **Ransomware**: Encrypting files and demanding payment

Attacks can have varying goals, including financial gain, espionage, or disruption

What is an Attack Vector?

Attack Vector:

The pathway or method used by an attacker to exploit a vulnerability.

Examples

- **Email Attachments**: Sending malicious attachments or links
- **Network**: Exploiting default services running on open ports or weak firewalls
- **Physical Access**: Plugging in a malicious USB device

Attack vectors define how an attack is initiated

What is an Attack Surface?

Attack Surface:

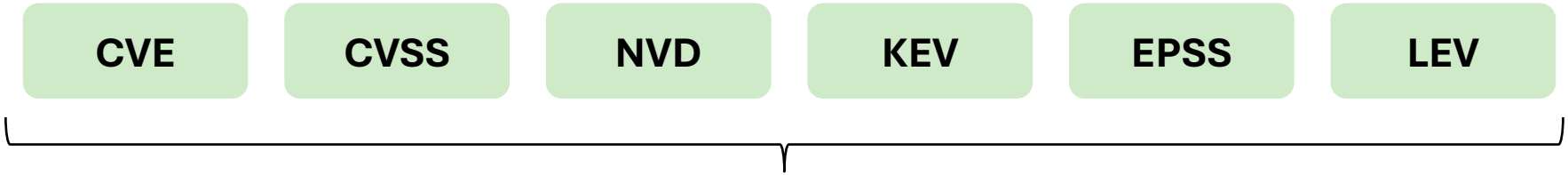
The sum of all points where an attacker can attempt to exploit a system.

Examples

- **Web Applications**: Input fields, APIs, and authentication mechanisms.
- **Network**: Open ports, unpatched services, and misconfigured firewalls.
- **Human Factors**: Social engineering through employees.

Reducing the attack surface minimizes opportunities for exploitation

Tracking vulnerabilities & risks



What are all these letters??

Tracking Vulnerabilities

When new software flaws are discovered, we need a way to track, rate, and respond to them consistently

The security community uses a set of shared systems to do this

- **CVE**: Provides unique identifiers for reported vulnerabilities
- **CVSS**: scores their severity
- **NVD**: U.S. government database that tracks this information
- **KEV**: lists known vulnerabilities that demand urgent attention
- **EPSS**: predicts the likelihood of exploitation
- **LEV**: helps companies prioritize which bugs to address

- **You don't need to memorize these, but you should know they exist**
- **CVEE and CVSS show up in a lot of cybersecurity articles**

Tracking Vulnerabilities With CVE

CVE (Common Vulnerabilities and Exposures)

A standardized system for identifying, defining, and cataloging publicly disclosed cybersecurity vulnerabilities

- Managed by **MITRE Corporation** at www.cve.org
- Sponsored by the U.S. Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA)
  - Currently contains over 240,000 records and growing
  - The latest records are posted on https://x.com/CVEnew

- **Structure: CVE-[Year]-[Number]**
- **Example: CVE-2024-11477**

  7-Zip Zstandard Decompression Integer Underflow Remote Code Execution Vulnerability

CVSS: Vulnerability Scoring

- ***How severe is the vulnerability?***
- ***Should I take steps immediately or is the risk of an exploit low?***

Common Vulnerability Scoring System (CVSS)

- A framework for rating the severity of vulnerabilities on a scale of 0 to 10

- **Severity Levels**
  - 0.0 – 3.9: Low
  - 4.0 – 6.9: Medium
  - 7.0 – 8.9: High
  - 9.0 – 10.0: Critical

These severity ranges are from v3.1. Version 4.0 was released in late 2023 but isn't fully adopted yet.

Example: CVE-2021-44228 (Log4Shell)

- **Base Score:** 10.0 (Critical).
- **Attack Vector:** Remote.
- **Impact:** Full system compromise.

MITRE does not store CVSS data – it only stores the ID, brief description, affected product, and references

NVD: A database that stores CVEs

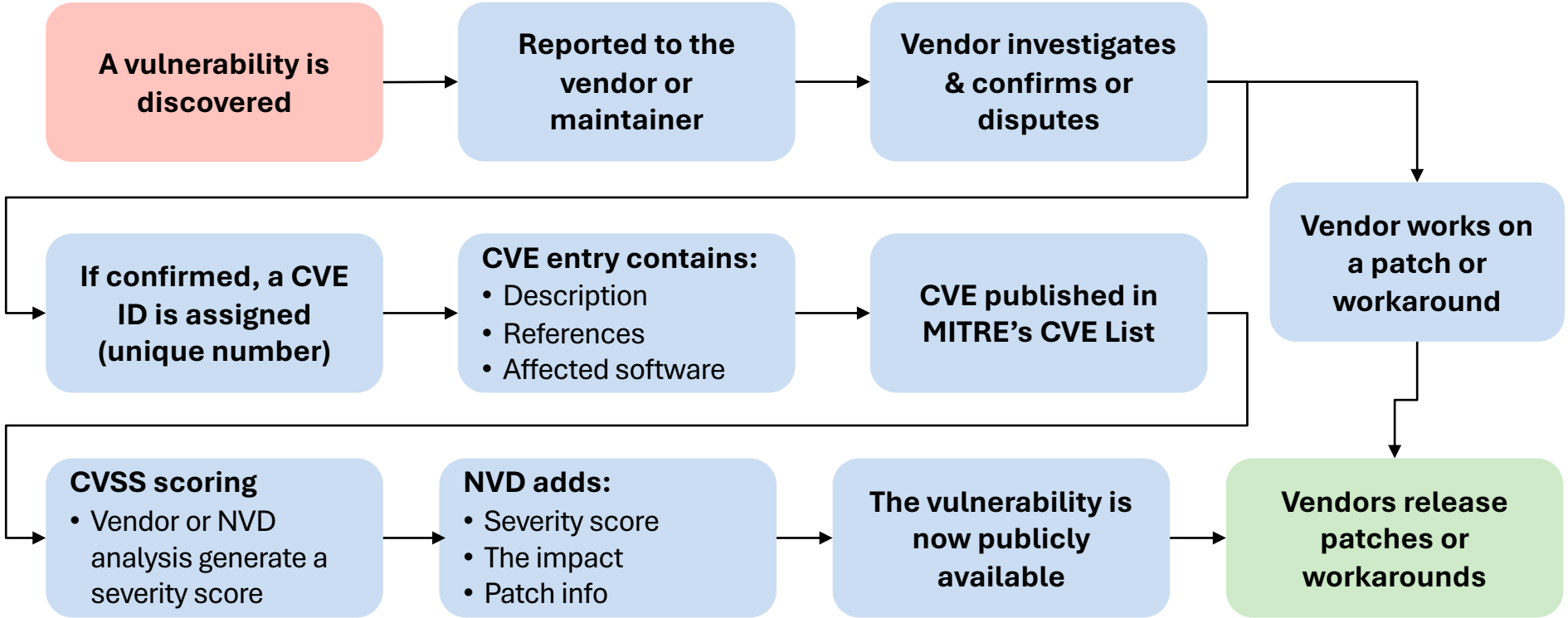
MITRE does not store CVSS data – it only stores:

- ID (CVE identifier)
- A brief description
- Affected product or vendor
- References

National Vulnerability Database (NVD)

- A database that tracks CVEs: https://nvd.nist.gov
- Maintained by the National Institute of Standards & Technology (NIST)
- Enriches the CVE reports with:
  - CVSS scores (impact/severity assignment)
  - Standardized product names
  - Additional metadata and references

Flow: From Bug Discovery to CVE and NVD



Known Exploited Vulnerabilities (**KEV**) Dataset

- **Subset of CVEs that have been confirmed as actively exploited**
  - CVEs cover all documented vulnerabilities
  - KEVs cover only those actively exploited – considered higher priority
- **KEVs are curated by CISA (Cybersecurity and Infrastructure Security Agency)**
  - The purpose is to identify vulnerabilities that present immediate risks
  - It helps organizations focus their efforts on patching vulnerabilities that pose the greatest risk

https://www.cisa.gov/known-exploited-vulnerabilities-catalog

Exploit Prediction Scoring System (EPSS)

- **Data-driven model developed by FIRST.org to estimate the probability that a vulnerability will be exploited in the wild within the next 30 days**
- **Daily file download of a CSV file from `cyentia.com`**
  - **CVE:**
    Identification of the vulnerability
    MITRE's CVE list
  - **EPSS:**
    The probability of exploitation in the wild during the next 30 days
    A CVE with EPSS of 0.90 has a 90% chance of being exploited in the next 30 days
  - **Percentile:**
    Ranking of this vulnerability vs. others on the list
    A CVE with a percentile of 0.82 means that 82% of other CVEs have a lower EPSS score

cve	epss	percentile
CVE-1999-0001	0.01269	0.78463
CVE-1999-0002	0.16835	0.94578
CVE-1999-0003	0.90483	0.99576
CVE-1999-0004	0.04164	0.88098
CVE-1999-0005	0.17478	0.94709
CVE-1999-0006	0.06778	0.90801
CVE-1999-0007	0.05124	0.89313
CVE-1999-0008	0.03952	0.87771
CVE-1999-0009	0.8048	0.99068
CVE-1999-0010	0.02	0.82737

https://www.first.org/epss/

https://orca.security/resources/blog/epss-scoring-system-explained/

Exploit Prediction Scoring System (EPSS)

- **Scores are computed via a machine learning algorithm**
- **Proprietary process – trained on data of**
  - Exploitations observed during the past day
    - From reports from companies and project partners
  - Vendor, age, CVSS, Common Weakness Enumeration (CWE)
- **The model learns the relationship between vulnerabilities and the exploitation attempts observed**

https://www.first.org/epss/

https://orca.security/resources/blog/epss-scoring-system-explained/

Likely Exploited Vulnerabilities (LEV)

A new metric proposed by the NIST in May 2025

Why not use EPSS or KEV?

- **EPSS** : global model that predicts the likelihood that someone, somewhere will exploit a CVE
- **KEV** catalogs known exploited vulnerabilities but lacks metrics to prioritize exploits

LEV

- Organization-specific model: assess how much the vulnerability matters in your own environment
- Is the software actually deployed internally?
- Is the software public-facing or isolated?
- What is the value of the asset (customer database vs. lab test machine)?
- Ease of exploitation in the local configuration

Note: LEV was proposed by NIST in 2025 and is not yet widely adopted – it is not a standard.

https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.41.pdf

https://www.infosecurity-magazine.com/news/nist-metric-lev-likelihood/

Lots of abbreviations!!

- **CVE: an identification of the vulnerability**
- **CVSS: how bad it could be**
- **NVD: the database where this information is stored**
- **EPSS: how likely it is to happen**
- **KEV: proof that it is happening now**
- **LEV: how much it matters to you**

What’s in a name?

Naming the attackers – what we see in news articles

"Chinese state-backed hackers known as Salt Typhoon allegedly targeted their latest victims: Charter Communications, Consolidated Communications, and Windstream."

"APT-36 Uses New TTPs and New Tools to Target Indian Governmental Organizations"

"Iranian Imperial Kitten hackers targeted Israeli organizations in October"

Advanced Persistent Threat (APT)

Sophisticated and prolonged cyberattack conducted by a well-funded and highly skilled group, often nation-states or state-sponsored groups

- **Advanced**
  - Use of highly targeted and customized attack methods, tools, and exploits
- **Persistent**
  - Often operate stealthily over months or years
- **Threat**
  - The attackers are capable of bypassing traditional security defenses

Examples:

China's APT41, Russia's Fancy Bear, North Korea's Lazarus Group

Naming APT Groups is Difficult and Messy

- **Challenges**
  - Attackers conceal their identities
  - Reuse tools
  - Operate through compromised infrastructure to hide their origin
  - Any technical indications can be misleading or shared across multiple groups
- **Naming**
  - Different security vendors, governments, and researchers may interpret the same evidence differently
  - Each organization uses its own system to name groups
  - This creates overlapping names for the same threat actor

Example: one Russia-linked group has multiple names: APT28, Fancy Bear, Forest Blizzard, Sednit, Sofacy, Tsar Team

Naming Schemes for APT Groups

Names usually come from organizations that identify the attacker

- There is no standard naming convention

1. APT Numbering

Coined by Mandiant

- Sequential numbers in order of discovery
  - **APT1** – attributed to the 2nd Bureau of China's People's Liberation Army – Unit 6139
  - **APT29** – attributed to Russia's Foreign Intelligence Service

2. Animal Themes

Used by CrowdStrike

- **Panda**: Chinese APTs (Deep Panda, Gothic Panda, …)
- **Bear**: Russian APTs (Cozy Bear, Fancy Bear, …)
- **Kitten**: Iranian APTs (Charming Kitten)
- **Tiger**: Indian APTs (Patchwork Tiger)

Naming Schemes for APT Groups

3. Numeric Codes

Used by FireEye, Palo Alto Networks

- E.g., **Group-3390**, **UNC2452**
- Tracked until more details are confirmed about their origin

4. Threat Actor Names

Named after campaigns or characteristics

- **Lazarus Group**: North Korean group infamous for the Sony Pictures hack
- **Equation Group**: allegedly linked to the U.S. NSA
- **Buckeye**: either the U.S. or China

5. Microsoft Naming Scheme

Two-word combination of

Weather – identifies geography

Adjective or *noun*: identifies the group

- Weather Terms
  - **Typhoon**: China
  - **Cyclone**: Iran
  - **Blizzard**: Russia
  - **Sleet**: North Korea
  - **Sandstorm**: Middle East
  - **Tempest**: Financially-motivated groups
  - **Storm**: Unknown threat group
- E.g., Salt Typhoon, Midnight Blizzard

Will This Get Easier?

- **No. Companies can't agree on identifying and naming threat actors**
- **But … in June 2025, Microsoft and CrowdStrike worked together to create a mapping of names for the same actor**
- **The hope is that this can eventually lead to some standard for naming**

MSFT	CRWD
AMETHYST RAIN	VOLCANIC TIMBER
AQUA BLIZZARD	PRIMITIVE BEAR
BRASS TYPHOON	WICKED PANDA
BROCADE TYPHOON	GOTHIC PANDA
BURGUNDY SANDSTORM	REMIX KITTEN
CADET BLIZZARD	EMBER BEAR
CANARY TYPHOON	CIRCUIT PANDA
CANVAS CYCLONE	OCEAN BUFFALO
CHARCOAL TYPHOON	AQUATIC PANDA
CHECKERED TYPHOON	DEEP PANDA
CINNAMON TEMPEST	HIGHGROUND
CIRCLE TYPHOON	EMISSARY PANDA

You don't have to remember any of this
but it shouldn't be confusing when you read about it

The end