Edge Security Challenges and Issues

Alessandro Carrega

TNT Lab – DITEN
University of Genoa

Information 1/2

- Lecturer: **Dr. Alessandro Carrega**.
  - *Email*: alessandro.carrega@unige.it.
  - *Skype*: alessandro.carrega@gmail.com.
  - *Telegram / Whatsapp*: **3487485497**.
- Duration: **20 hours**.
- Language: **English**.
- Lesson in site and remote (Teams).

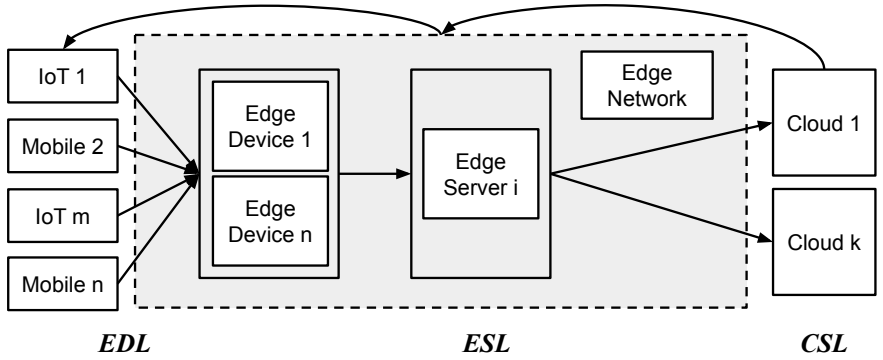
Information 2/2

- Dedicated Teams channel:
  - **PhD STIET Cyber security approaches for Cloud/Edge Environments**

    https://teams.microsoft.com/l/team/19%3a-Dtnw_NHUAl1AjZZV4HixIifmU8gywbskeeQwSV--uk1%40thread.tacv2/conversations?groupId=bdafff5c-0ab9-44b2-aef2-5a14e1dd6e15&tenantId=6cd36f83-1a02-442d-972f-2670cb5e9b1a
- GitHub repository:
  - https://github.com/tnt-lab-unige-cnit/phd-stiet-cyber-security-approaches-cloud-edge-environments
- Optional homework.
  - Available in Teams and GitHub.
- Final Exam with 3 options:
  - **Theoretical**: *short survey with 3 papers*.
  - **Pratical**: *2 exercises*.
  - **Quiz**: *100 multiple choice questions (**60%** to pass exam)*.

Edge Computing Framework

- General architecture of edge computing
- Three layers:
  - an edge device layer (EDL) – *most* power,
  - an edge server layer (ESL),
  - and cloud server layer (CSL) – *least* power.



Edge Device Layer (EDL)

- *Edge devices* grouped in
  - *IoT devices,*
  - *Mobile devices.*
- Conduct field tasks such as *sensing*, *actuating*, and *controlling.*
- Common edge devices controlled by microcontrollers (MCUs)
  - MCU implements firmware that provides low-level software interfaces to control device's hardware.

IoT Devices

- Lightweight electronic devices interconnected or connected to ESL through wireless protocols such as 4G/5G, WiFi, and Bluetooth.
- Examples include smart home devices, health monitoring devices, and smart warehouse carts in industrialized IoT.
- Most IoT devices use Cortex-M series MCUs produced by STMicroelectronics. Once real-time operating systems (RTOS) [33], [147], [234] is burned into IoT devices, no additional programming interfaces are provided under normal circumstances.

Mobile Devices

- More advanced operating systems, such as Android and iOS, which provide programmable interfaces for programmers to develop applications compatible with OS.
- Common mobile devices include smartphones, tablets, and central controllers of smart vehicles, and usually adopt Cortex-A series MCUs produced by high-performance chip manufacturers such as Qualcomm.

Edge Server Layer (ESL) 1/2

- Edge servers handle core computing functions in edge computing
  - including authentication, authorization, computation, data analytics, task offloading, and data storage.
- Consists of multiple hierarchical sublayers of edge servers with increasing computational power.
- Devices such as wireless base stations and access points (APs) sit at lowest sub-layer.
  - Mainly responsible for receiving data from edge devices and transmitting control flows back to them.
- Base stations/APs forward data received from edge devices to edge servers in higher layer to perform individual computation tasks.

Edge Server Layer (ESL) 2/2

- When task is too complicated to be handled on current edge server, task will be delegated to servers with more computational power at even higher sub-layer.
- After task is handled properly, sequence of control flows will be passed back to base stations/APs, and eventually transmitted to edge devices.
- Popular state-of-the-art edge servers include:
  - NVIDIA Jetson Nano,
  - Raspberry Pi,
  - Marvell OCTEON 10 DPU, and
  - Mac Mini.

Cloud Service Layer (CSL)

- Hosts cloud servers and data centers.
- Cloud servers are responsible for highest level of operations and integration of tasks offloaded from EDL and ESL.
- Data centers storing vast amount of data generated in edge computing infrastructure.
- Consists of clusters of powerful machines.

Security Characteristics of Edge Computing

- Edge computing and cloud computing are similar with respect to offered services and functionalities.
- Scopes of security measures are largely different.
- New challenges due to unique characteristics of edge computing platforms.

Weaker Computation Power

- Compared to cloud server, computation power of edge server is relatively weaker.
- Edge server might be more vulnerable to existing attacks that are less effective towards cloud server.
- Compared to general-purpose computers, edge devices have less robust defense systems.
- Attacks that have been mitigated for desktop computers can still pose serious threats.

Large Volume of Interconnected Devices

- Client devices in cloud computing are usually not interconnected;
  - limiting influential impact when few are compromised.
- Edge devices are typically interconnected.
- Small intrusion can have more significant impact if attack is spread and propagated among devices (e.g,. Mirai botnet).

Heterogeneous Device Form Factors

- Devices of various form factors can co-exist in edge computing, especially in EDL.
- Depending on purpose of device and physical location being deployed, two devices might have totally different hardware and software stacks.
- Such heterogeneity of devices poses extreme challenges when designing general solutions to potential threats and issues.
- Number of different scenarios of devices interacting with each other increases exponentially when number of devices increases.
- Handling all those scenarios either sacrifices generality if target specific subgroup of edge devices, or loses accuracy when searching for panacea.

Unavailability of Security Features

- Various hardware security features have landed on modern platforms to mitigate existing and unforeseen threats.
- Meanwhile, architectural improvements are being introduced to CPU, as well as its chipset.
- Not to mention collection of software mitigation techniques built on top of those platform-specific features.
- Unfortunately, due to different form factors and diverse of platforms, desired security features are not always available on specific edge computing platforms, and tough decision of security versus cost-effectiveness has to be made.

Maintaining Quality of Service

- Last but not least, any additional security measurements should try to maintain original quality of service (QoC) (e.g., availability and real-timeness) at best effort, as it is initial purpose of edge computing.

Multi-dimensional Security Analysis of Edge Computing

- Logical components of edge computing stack are:
  - device hardware,
  - firmware and system,
  - network and communication,
  - cloud stack (such as Kubedge),
  - machine learning as workload infrastructure and compute apps,
  - data protection, privacy, cryptography, users, identity and
  - access management,
  - and regulatory compliance.

Device and System 1/2

- As backbone of edge computing, edge device itself and system on top serve as shield of edge computing system at lowest level.
- System software such as OS often runs at highest privilege and mediates management tasks across process domains.
- At same time, hardware provides resources for OS to correctly fulfill tasks as intended by user.
- As result, hardware and system software define trustworthiness of standalone edge device:
  - fundamental building block to achieving secure edge computing service and fully unleashing its power.

Device and System 2/2

- Defined hardware and system software of edge device as edge platform.
- To support desired security requirements of edge computing services, ideal edge platform should consist secure
  1. physical protection,
  2. hardware components,
  3. firmware, and
  4. system software.

Re-Imagining Existing Threats Under Edge Computing

- As computing devices themselves of different form factors, edge platforms are susceptible to existing security threats if they possess targeted attack surface.
- Worse, unique characteristics of edge computing may exacerbate impact of such attacks.
- Together with privileged attackers that possess and provision edge devices of diverse form factors, there are various challenges to guarantee trustworthy edge platform.

Insecure Physical Protection 1/2

- Physical access is commonly assumed as last layer to defend against attackers.
  - It is always excluded from threat models of cloud computing.
  - One of initial weapons granted in attacker's arsenal under edge computing.

1. Intrusive attacks require physical connections to device, such as accesses to communication ports and channels (e.g., USB, PCIe), or direct tempering motherboard (e.g., via soldering).

Insecure Physical Protection 2/2

2. Physical side-channel attacks focus on leaking secrets based on physical behaviors of components during security sensitive workloads, including power analysis, electromagnetic analysis, and so on.
   Launching such attacks requires an attacker to access target device physically or through malicious apps, which is highly feasible under edge computing, where attack can be conducted in an isolated and stable environment.

Physical Protection Boundary at Edge

- Cloud servers' physical access control models do not apply to edge devices due to high volume and diversity.
- Not to mention procedure of access control needs to be audited by cloud server from remote.
- Hence, an adequate physical access control model tailored for edge computing is under urgent call.

Insecure Hardware Components 1/3

1. Common source of insecure hardware components stems from broken chain of trust of hardware components
   - such as due to an untrusted supply chain or
   - lack of unforgeable Hardware Security Modules (TPMs),
   - and Hardware Security Modules (HSMs).
- Such vulnerabilities might hide in original hardware package and are difficult to mitigate thoroughly without replacing whole flawed component.

2. Other hardware attacks exploit inherent design of components that are critical to its correct functionalities and thus are even more challenging to mitigate holistically.

Insecure Hardware Components 2/3

- *Energy attack* aims to render device inoperable by draining equipped batteries through excessive legitimate operations. Such an attack can be launched across different layers of device stack, including hardware resources (e.g., GPS, sensors, and related operations), software resources (e.g., system calls, API, memory allocation, locking), network operations (e.g., data transfer, handshaking protocols, bandwidth, and antenna).

Insecure Hardware Components 3/3

- *Rowhammer attack* tries to trigger random bit flips in RAM/DRAM via electronic interference of neighboring memory cells to tamper with security-sensitive states (e.g., access control bit, root bit, etc.).
- *Covert channels* have drawn increasing attention as they exploit unintended communication channels stemming from normal operations of shared components such as DRAM and Last Level Caches (LLC).
  - They can break data protection mechanisms enforced by system to restrict unintended communications and provide malicious applications stealthy way to transfer (security-sensitive) data between each other.
- *Mircoarchitectural side-channel attacks* exploit secret related micro-architectural events, including
  - those inside CPU caches (e.g., Prime+Probe, Flush+Reload), and Translation Lookaside Buffers, branch predictors with speculative executions (e.g., Spectre and Meltdown).

Edge Hardware

- Attackers under edge computing environments are even more advantageous in hardware attacks.
- Such attackers directly possess devices and can perform physical attacks within an isolated and stable environment tailored for their needs, making traditional hardware attacks more feasible.

Insecure Firmware 1/4

- Computing platform delegates complexity of hardware initialization tasks to earlier boot stage using firmware in order to simplify operation system code.
- Under edge computing, hardware and firmware's diverse and proprietary nature proliferates fear of effective widespread exploitation.

Insecure Firmware 2/4

- *Firmware modification attacks* aim to inject malicious logic into target device firmware.
- Usually, such attacks are achieved via firmware update features instead of directly exploiting flaws in software.
- Firmware update is common feature in most modern systems, while not ubiquitously protected by sufficient security measurements.
- Lack of security checks (e.g., signature verification) prior to firmware updates directly facilitates successful firmware modification attacks.

Insecure Firmware 3/4

- Secondary payload following successful exploitation of device via traditional attack vectors, such as memory modification attack, can be used to bypass checks if they exist.
- Malicious code running at firmware level could be used to compromise any components that are loaded later in boot process, such as boot loader and OS (or hypervisor).

Insecure Firmware 4/4

- *Hard-coded and weak passwords* in firmware is another major concern, especially in devices that embed default passwords while lacking administrative management.
- Although use of hard-coded or weak passwords can save maintenance overhead, it leaves devices vulnerable to naive password-based attacks such as dictionary attacks.
- Such attacks are extremely easy to conduct using existing tools (e.g., John the Ripper, HashCat) without domain-specific knowledge.

Firmware in Edge Hardware 1/2

- Besides threats to integrity of firmware, edge devices with legitimate but outdated firmware are also in attackers' interest.
- Because of different form factors of devices, diversity of environment for devices in field, and specific management requirements, legacy devices cannot be updated in time and therefore expose known vulnerabilities to attackers.

Firmware in Edge Hardware 2/2

- As attacker under edge computing has more authority over operation environment (e.g., by directly possessing device or by less strict physical access control), firmware updating process could be spoofed (e.g., via MITM) to preserve stale version.
- As result, edge computing may magnify impact of legacy firmware attacks due to attacker's overseeing of firmware management process and much larger amount of stale devices.

Insecure System Software 1/3

- When OS and system software are trusted, i.e., requests from users are faithfully fulfilled, traditional software vulnerabilities can still exist.

1. *Memory corruptions* (e.g., memory overflow, improper boundary check, lack of sanitization, double-free, use after free) can lead to control flow hijacking (e.g., ROP), and result in violations of integrity and confidentiality of system data, or even grant complete access to system (e.g., root-access).

Insecure System Software 2/3

2. *Race conditions* in system can allow TOCTOU attacks and lead to similar consequences.
3. Even with formally verified OS that are free of such vulnerabilities, *flaws in device drivers and third-party libraries* can still completely break established security guarantees.
4. Besides traditional software vulnerabilities, *improper access control implementation* allows attacker to gain access to sensitive data without launching end-to-end exploit and generate data flows that are otherwise disallowed.

Insecure System Software 3/3

5. *System software cannot always be trusted*, as devices are directly possessed and provisioned by untrusted administrative parties.
   - When facing privileged attackers such as malicious OS and hypervisors, Trusted Execution Environments (TEEs) always come into play to isolate security-sensitive content.
   - However, privileged attackers can still use side-channel attacks to study program runtime behaviours, such as through control flow or memory access patterns, and leak secret, as OS is still responsible for management tasks such as handling page faults.

SELinux and Security Monitors

- Edge devices and servers that enable defensive features SELinux and security monitors may temporarily turn off these features for power and performance considerations, leaving chances for attacks.
- Not to mention that features are not free of vulnerabilities.
- An untrusted OS can maliciously manage edge devices, and temper with overall infrastructure of edge computing entity.

Network and Communication

- Looking into network-level security concerns of edge computing, one must deal with various threats and vulnerabilities pertaining to communication among network nodes.

Device Vulnerabilities 1/3

- Physical layout of edge network introduces new threats.
- One such threat is physical access to machine itself.
- Edge nodes are physically located close to where data are generated or need to be processed.
  - It is beyond vendor's or service provider's physical control.
- Threat actor may leverage physical access to device to perform various activities.
  - For example, they could cause Denial of Service (DoS) by damaging or unplugging device or attempting to penetrate system by connecting to an open port, or replacing device with rogue one.
  - Wiretapping is also possible, allowing packet sniffing or even injection.

Device Vulnerabilities 2/3

- In real-world deployments, devices are highly heterogeneous and may be developed by different vendors.
- Undebatable fact, especially in context of IoT.
- Such devices are often found running vulnerable code (or firmware) that has not received proper security assessments.
- Some devices may have common vulnerabilities, such as out-of-bounds-write, which adversaries can leverage.
  - For instance, Philips Hue Bridge (model 2.X) contains heap-based buffer overflow which allows remote code execution.
- Often, device may utilize buggy libraries even though they could be well-known.
  - OpenSSL library is widely used cryptographic library that implements secure communication protocols such as SSL/TLS.
  - According to CVE-2014-0160, heartbleed bug in OpenSSL's heartbeat implementation leads to memory leakage from server to client and from client to server. Such leakage could reveal secret keys or other protected material.

Device Vulnerabilities 3/3

- Network nodes such as Network Address Translators (NAT) carry vulnerabilities of their own.
- NAT “hides” IP addresses of internal network devices from outside network.
- All internal devices share one public IP address.
- Threat actor from outside cannot know which devices exist in internal network.
- However, slipstreaming vulnerability found in NAT actually allows outsiders to learn which devices are in internal network and probe them.
- Adversary only needs to convince one of devices in internal network to connect to malicious domain.

Vulnerabilities in Edge Services

- Adversary may be able to inject malicious payload using several underlying techniques such as SQL injection, XSS, and CSRF.
- Main root cause of all those techniques is lack of proper input sanitization, where edge server needs to check validity of input request.
- Servers merely check only for correct syntax, which is trivial.
- They do not reason about contents of request.
  - E.g., code is found where there should be data or vice versa.
  - For instance, vulnerability was found in Cisco Fog Director that allowed unauthenticated attacker to conduct XSS attack against user of web interface of affected software.

Protocol Vulnerabilities 1/2

- Edge network usually uses lightweight communication protocols such as
  - LTE, Wi-Fi, Bluetooth, MQTT, CoAP, AMQP, LoRa, and Zigbee.
- Cloud computing employs heavyweight protocols such as
  - TLS, HTTPS, and FTP.
- Protocol itself becomes part of attack surface.
- Soft spots in communication protocols may provide an attacker grounds to launch attack.
- Zigbee (before version 3.0) required communicating devices to share pre-master secret key.
  - Key was installed on devices by vendors.
  - Considering millions of IoT devices, adversary can inevitably figure out pre-master secret.
  - Another possibility is that adversary may force endpoints to downgrade security of communication. Such attack is possible if endpoints have no way of verifying that security properties agreed are ones that were truly intended.
  - POODLE attack (Padding Oracle On Downgraded Legacy Encryption) on SSL 3.0 allows stealing secret material, such as HTTP cookies.
- HTTP is used in edge-cloud communication.

Protocol Vulnerabilities 2/2

- Message Queuing Telemetry Transport (MQTT) is an application-layer communication protocol widely used for IoT to edge communication.
  - While MQTT protocol supports encrypted communication, it is optional.
  - This configuration could result in critical privacy violations while allowing man-in-the-middle to inject messages.
  - For instance, data generated from wearable devices could include highly sensitive medical data, personal information, and even people's movements.
- CoAP is another application layer for edge devices and applications that works on top of UDP.
  - It has been reported that CoAP is susceptible to attacks such as address spoofing and amplification attacks.
  - Response packet is much larger than request packet, and thus an attacker can use small UPD requests to generate large-size responses from CoAP nodes, thus causing denial-of-service to victim devices (see section 11.3 in RFC 7252).

Establishing Trust

- *Authentication* poses another challenge.
- Low-end devices may be authenticated to edge servers using weak credentials.
- Adversary may even be able to perform dictionary attack to break into system.
- Vulnerabilities may be introduced due to usage of weak cryptographic authentication protocols (e.g., WEP) or unpatched versions of them (e.g., WPA2-PSK dictionary attack).
- Authentication code-level implementation poses additional threats to entity authentication.
  - Code that has not been tested or peer-reviewed may implement authentication in wrong way, granting access to unauthorized entities.
  - Good example of that is Apple's "goto fail" bug, which allowed MITM adversary to compromise end-to-end secure TLS connection.
  - Bug was essentially bypassing certificate verification of server, thus compromising authenticity of connection.

Compute Authorization

- One crucial contribution of edge computing is delegation of complex computing tasks to network's edge.
- While authentication aims to solve problem of verifying identities, authorization deals with problem of verifying whether particular node is authorized to perform particular computing task.
- A similar problem is encountered in Content Delivery Networks (CDNs), where media stream providers (e.g., Netflix) authorize servers to deliver content on its behalf in several regions.
- Same physical spread applies in edge computing as well. Similar to how end-users need to verify content received from CDN server, an edge device needs to be able to prove that it is authorized to perform such computation from core network (e.g., corporate network, service provider).

Side-channel Attacks

- When attacker can gain knowledge about occurring communication and endpoints communicating, they may be able to use this information to attack infrastructure itself.
- Information leaked through side channels often reveal information about secret keys.
- Ronen et al. demonstrated novel side-channel attack on Philips Hue smart lamps, which revealed initialization vector and secret key that lamps use to authenticate and encrypt firmware.
- Thus, information gained from side channel could lead to catastrophic results.
- In addition, side channels raise privacy issues, especially in environments where privacy preservation is of high importance (e.g., smart home).

Denial of Service Attacks 2/2

- Edge computing is complex infrastructure that includes interconnection of edge devices, edge servers, and cloud.
- Edge device vulnerabilities or vulnerabilities in protocol itself may allow an attacker to cause disruptions.
- Edge network is more susceptible to distributed denial of service (DDoS) attacks since it contains computationally less powerful resources than cloud.
- Services deployed at edge are error-prone in their security settings.
  - For instance, if attacker can take control of cluster of edge devices, they essentially create botnet.
  - A very famous attack is Mirai botnet where attackers were able to compromise IoT devices and use them to cause DDoS to edge server network providers such as Krebs and OVH.
  - Botnet floods core network (i.e., edge servers) with enormous requests.
  - Such attack causes DoS due to resource exhaustion.
  - Techniques to cause denial of service include message flooding at internet layer (e.g., ICMP), transport layer (e.g., TCP, UDP), or even application layer (e.g., HTTP).

Denial of Service Attacks 1/2

- While DDoS attacks are more practical in edge network, cloud DDoS is still feasible from attacker's perspective.
- Successful DDoS attacks on cloud can cause significant disruptions in edge network.
- CVEdetails report top 50 vulnerable products, many of which run on cloud servers (e.g., Windows 10 and Linux Debian).

Research Challenges 1/3

Network security configurations and management

- Edge computing is designed to support range of devices spanning different vendors with different security capabilities.
- Different vendors provide different application programming interfaces to configure device. Administrators must make significant efforts to adequately configure devices, making it costly and error-prone task.

Research Challenges 2/3

Adoption of Zero Trust Architecture (ZTA)

- As argued earlier in section, access control is critical when managing large distributed infrastructures like edge computing.
- Frameworks are needed to support fine-grained access control policy specifications; which communications should be allowed and why.
- Defining such access control policies needs to be done in way that keeps errors minimal.
  - For example, administrator may specify at high level which entities need to communicate and have automated process to translate this to network-level details (e.g., using firewall rules, etc.).
  - Often, organizations adopt threat models with weak adversarial assumptions.
    For instance, they may assume that adversary may not be able to compromise end devices.
    - Access control is loosely defined within network while enforcing more access control on network perimeters (i.e., firewalls).
    - Becomes major issue once presumably trusted devices misbehave.
    - For example, compromised IoT device may start probing resources on network or try to spread infection further in network.
    - Such attack is feasible since access control is non-existent or very loosely defined.
    - One needs to precisely define who can access what on network and for what reason.

Research Challenges 3/3

Secure access service edge (SASE)

- SASE extends ZTA by granting access to network resources based on entity's identity (e.g., IoT, device, user application)
- Requires real-time context and continuous assessment of risk and trust throughout sessions.
- Real-time contexts can vary across edge computing deployments and must be standardized for compatibility across SASE services.
- Assessment of risk and trust must be continuous monitoring process as long as entity lives in network.
- Risk and trust must account for threats and vulnerabilities emerging from different architectural stacks.
  - e.g., network, hardware, system.

Cloud Stack

- Cloud composes central processing and maintenance of data collected in edge network.
- Edge network transmits data to cloud for further processing or permanent storage, among other purposes.
- Cloud is distributed system composed of several interconnected machines similar to edge network.
- Cloud resources are shared among many parties.
- In public clouds, those parties can be different users or organizations.
- Security issues emerging in cloud impact edge network as well.
- Despite its architectural incentives and benefits, cloud brings its security issues to edge network.

Cloud Vulnerabilities 1/5

Misconfigurations

- Data can be exposed simply by misconfiguration of cloud machine instances.
- For example, extremely sensitive user data were leaked by cloud leader due to misconfiguration of Amazon Web Services S3.
- Misconfiguration allowed public access to cloud server hosting data while also having no access control enforcement.

Cloud Vulnerabilities 2/5

Insecure APIs

- Cloud services expose their functionality through Application Programming Interface paradigms (e.g., REST).
- Gartner predicts that by 2022, application programming interface (API) attacks will become most-frequent attack vector, causing data breaches for enterprise web applications.
- Most critical issues in API security are insufficient access control, injection, and Excessive Data Exposure, among others.
- Vulnerability in Microsoft Exchange Server allowed attackers to send unauthenticated HTTP requests to Exchange server.
- Broken user authentication and security misconfigurations allowed adversaries to leverage back-end API of server to escalate privileges and maintain persistence.
- Similar scenario could be encountered in edge computing environment.
- Services deployed at edge expose API functionality to be used by devices.

Cloud Vulnerabilities 3/5

Virtualization issues

- Virtualization operations can be distinguished into two components:
  1. Virtual Machine Manager or Hypervisor (VMM);
  2. Virtual Machine instances.
- VMM is crown jewel since its compromise allows adversary to attack many virtual systems at once.
  - In 2021, ransomware attack occurred on VMware ESXi hypervisor.
  - According to Sophos firm, defensive mistakes and unnecessary functionality allowed attackers to deploy crypto-locking malware and disrupt operations of all tenant VMs.
- VM integrity must be verified at all times in cloud.
- VM could be running mission-critical applications for edge network.
  - Cloud providers must ensure that there are no malicious or compromised VM images.
- Vulnerabilities associated with virtualization environments may allow attackers to obtain administrator privilege in host system due to mishandling of privileges.

Cloud Vulnerabilities 4/5

Cloud Service Provider Transparency

- There is no clear understanding of how cloud operations are performed.
- Not feasible to assess security posture of edge network thoroughly.
- For example, unclear what protection measures cloud provider takes for data confidentiality.
  - How are data stored, or when do they get decrypted for processing?
  - Are there any process isolation mechanisms in place?
- Similar considerations apply to data integrity and availability.
  - For example, when does signing process takes place and where?
  - What is mechanism employed for data availability?
- Another interesting aspect is if third parties are involved in cloud operations.
  - For instance, for data backups.
- Transparency is key factor for clients and network administrators to realize risks and see how things can go wrong.

Cloud Vulnerabilities 5/5

Cloud at Edge (aka Edge Cloud)

- Offers cloud computing resources to edge of network or where traffic is.
- Various frameworks have been developed, offering cloud services at edge, such as
  - KubeEdge, EdgeX Foundry [6], and OpenEdge.
- Frameworks are part of cloud stack and bring their flaws into edge infrastructure.
- Recent vulnerability was disclosed where specific binaries within OpenEdge application were susceptible to privilege escalation.
  - Local attacker could elevate their privileges and compromise affected system.
- Similarly, improper access control and weak password requirements in EdgeX.
  - Attacker could make authenticated API calls to EdgeX microservices from an untrusted network.

WebAssembly (Wasm) 1/2

- Wasm was proposed by World Wide Web Consortium (W3C) as platform-independent compilation target for various high-level languages (e.g., C, C++, Rust).
- Originally, Wasm addressed problem of safe, fast, portable low-level code on Web.
- Abstraction over modern hardware, Wasm is language-, hardware-, and platform-independent, with use cases beyond Web.
- Wasm adopts linear memory with configurable size for all memory accesses other than local and global variables.
- Liner memory region is disjoint from other memory regions (e.g., code space, execution stack), containing impact of vulnerabilities within data of program's own memory.

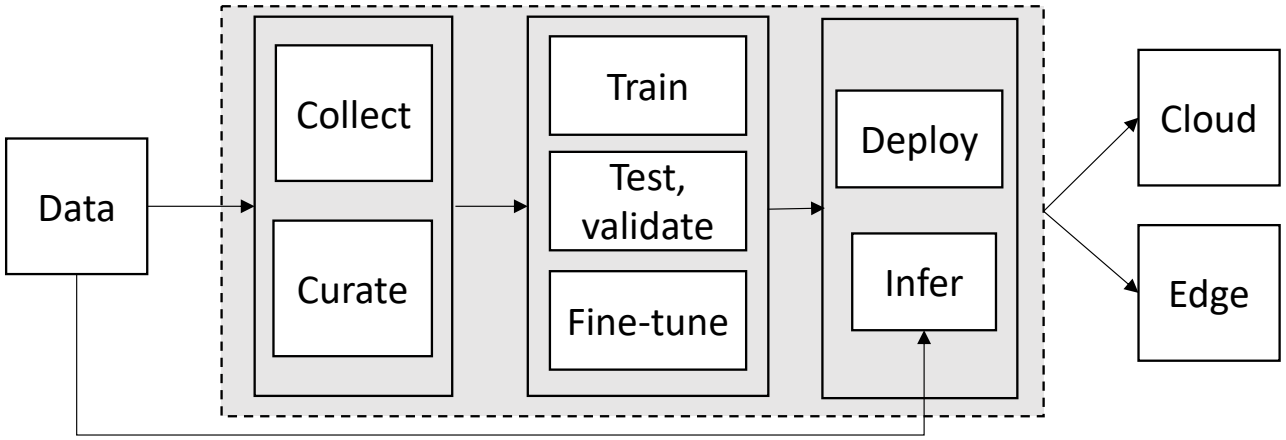
WebAssembly (Wasm) 2/2

1. Beyond all benefits brought by Wasm, research has shown that traditional vulnerabilities re-instantiate in Wasm.
2. Wasm enables unique attacks, such as over-writing constant data or manipulating head using stack overflow.
   - Attack primitives found in Wasm include but are not limited to those that allow an attacker:
     1) to write arbitrary memory,
     2) to overwrite sensitive data, and
     3) to trigger unexpected behavior by tampering with control-flow integrity (CFI) or host environment.
3. Threats to Wasm at edge. Researchers have explored application of Wasm under setting of edge computing.
   - Wasm designed as platform-independent, exposed vulnerabilities will remain in edge devices that adopt Wasm, with potentially higher impacts by propagating outcome across massive and heterogeneous edge computing network.

Machine Learning Workload

- Increasing amount of data generated by large number of devices in edge network.
- Opportunities, challenges, and applications of edge machine learning (Edge-ML) increased.
- Edge-ML widely used for various edge computing tasks.
  - e.g., computer vision for traffic surveillance,
  - decision making for autonomous driving, and
  - speech recognition for personal assistance.
- Considering huge number of devices in edge network, training edge models from scratch will require tons of computational resources.
- Transfer learning along with other techniques introduced to reduce development cost of edge-ML systems.

Edge-ML Life Cycle



- Diverse edge computing tasks atop different edge environments and new learning paradigms bring about set of domain-specific security problems and challenges of edge-ML.
- Building trustworthy edge-ML systems requires securing whole life cycle of edge-ML systems in terms of data, model, and infrastructure.

- Includes stages of data collection and curation, model training, testing and validation, deployment, and inference.
- Key feature of edge-ML is allowing model training across decentralized edge devices or servers that hold local data samples without exchanging them (*federated learning*).

Edge-ML Threats

- Availability and visibility of local data depend on two different training patterns:
  1. local data are collected and uploaded to servers while models are trained in centralized way and
  2. local data is only available to local workers (edge devices) in federated learning scenarios.
- For first training pattern, all known threats for centralized machine learning systems can be applied.
- For second training pattern, there are new threats due to nature of distributed learning.
- Distributed learning makes it difficult to audit quality of local data and training behaviors of local workers.
- Malicious local workers can manipulate final models by modifying training data in silence.
- Attacks can also be performed on data collection, transmission, and processing phases by attacking benign workers.
- When attacks are carried out by any participant (local workers or servers) in decentralized learning system, they are defined as insider attacks.
- Outside attacks are mainly carried out by sniffing/inferring information about data or models.

Edge-ML Vulnerabilities 1/4

- New threats to edge-ML systems introduce new vulnerabilities by which adversaries can further develop exploits to attack systems.

Training-time attacks

- Adversary can manipulate training data to perform poisoning attacks, injection attacks, adversarial perturbation attacks, and byzantine attacks.

1. *Poisoning attacks* aim to poison training datasets by adding small number of poisoned samples (e.g., 5% of all training samples).
   - Data poisoning process can be achieved by providing wrong samples, injecting perturbations, or flipping labels.
   - For example, attackers can add fake data via SSD attacks, compromise edge software systems and devices, and IoT spoofing.
   - Victim edge-ML systems will learn erroneous features and update model weights incorrectly.

Edge-ML Vulnerabilities 2/4

Training-time attacks

2. *Injection Attacks* are prosecuted by inserting trojans and bias to only target data sample called *target attacks*.
   - Triggers can be physical objects (e.g., stickers on stop sign) or digital triggers (e.g., watermarks on images), which can fool edge-ML systems (e.g., autonomous vehicles and face recognition cameras) to predict inputs with triggers as target wrong labels.
   - Attackers can also add bias, resulting in fairness problems to victims.
   - For example, they can manipulate edge sensor data to force classifiers to learn unbalanced labels (e.g., gender).

Edge-ML Vulnerabilities 3/4

Training-time attacks

3. *Adversarial example attacks* are performed by adding perturbations into training samples, carefully generated and indistinguishable to human eyes.
   - Attackers can actively generate dynamic and optimized perturbations to mislead machine learning systems.
   - For example, adversary can attack edge network intrusion detection systems by adaptively mutating network features (e.g., device IP address and port number) by DDoS, reconnaissance, and information theft attacks.

Edge-ML Vulnerabilities 4/4

Training-time attacks

4. *Byzantine attacks* are carried out when cloud or edge servers select malicious clients in federated learning stage.
   - Attackers can contaminate local edge device data via poisoning attacks and use it to train local models.
   - They upload well-trained or directly tampered malicious model weight updates to servers, which can damage final model hosted in servers or leave backdoor.
   - Except for malicious model weights, attackers can also directly upload malicious local data (crafted by data manipulation) to servers.
   - When data security checks fail to identify these vulnerabilities in servers, attacks will succeed.

Edge-ML Vulnerabilities 1/4

Inference-time attacks

- Performed in stages of offline validation/testing and online inference.

1. *Exploratory attacks* aim to explore architectures of machine learning system and build shadow models (or edge-ML systems).
   - Adversaries (e.g., malicious edge devices) have black-box access to victim model (e.g., final model) through MLaaS APIs, where edge or cloud servers provide machine learning service APIs to edge devices.
   - Attackers first query victim models with synthesized samples to get labels and then train functional equivalent shadow models based on labeled data.

Edge-ML Vulnerabilities 2/4

Inference-time attacks

2. *Membership inference attacks* are performed to determine if given data samples are members of victim models' training data.
   - In edge-ML, overfitting final models force them to memorize training samples, which can increase success rate of this attack.

Edge-ML Vulnerabilities 3/4

Inference-time attacks

3. *Evasion attacks* are to manipulate inference results by carefully crafting test samples.
   - Adversary can be either insider attackers or outsider attackers that compromise edge devices or invade edge communication network to generate samples that have close probabilistic distribution as original training data to fool victim models, resulting in prediction errors.

Edge-ML Vulnerabilities 4/4

Inference-time attacks

4. *Spoofing attacks* are performed by directly generating adversarial test samples from scratch rather than crafting existing samples.
   - Adversary (e.g., malicious edge devices) can spoof edge device data by GAN networks [87], which consist of two components:
     - *generators* create spoofing signals, and
     - *discriminators* detect whether samples are spoofed or not, where they minimize their own low by playing minimax games.

Edge-ML Vulnerabilities 1/3

Deployment attacks

- After training, validation, and testing stages, edge-ML models will be deployed in edge networks, where they can be placed in edge/cloud servers or edge devices.

1. *Model stealing attacks* occur in cloud/edge server and edge device sides.
   - For cloud servers, edge-ML models are invisible to attackers except for scenarios where attackers can invade servers.
   - Indirectly steal model by above-mentioned inference-time attacks (e.g., exploratory attacks).
   - When edge-ML models are deployed in edge servers and devices, it is much easier to get models due to lack of model protection techniques.
   - Most of edge-ML models are built atop popular and publicly available machine learning frameworks (e.g., Tensorflow Lite), which means that attackers can easily reuse stolen models.
   - Although many countermeasures, e.g., model deployment with trusted execution environments, have been proposed, limited resources of edge devices make them far from practical.

Edge-ML Vulnerabilities 2/3

Deployment attacks

2. *Supply chain attacks* are performed on edge-ML models distributed in supply chain.
   - As deep neural networks become deeper and deeper, edge-ML models are built on existing pre-trained models (e.g., mobileNet) that are specifically designed for edge computing environments via transfer learning.
   - Models need to be distributed on numerous devices for both federated learning and online inference tasks.
   - Supply chain of pre-trained model acquisition and model distribution, will be target of adversaries.
   - For example, attackers can stealthily modify weights of models by man-in-the-middle attacks on edge network.

Edge-ML Vulnerabilities

- Threats and vulnerabilities in edge-ML frameworks.
- Carefully designed to overcome resource limitation issues.
- Gained more and more attention recently, and many vendors have designed their own frameworks for edge environment, such as:
  - TensorFlow Lite, Apple Core ML and Pytorch mobile.
- Frameworks for general machine learning are used in edge devices:
  - e.g. Caffe2, NCNN, and Parrots.
- One of key vulnerabilities is *calculation* vulnerability.
  - Happens when unexpected data point or value is given to edge-ML frameworks, which involves physical dimensions of edge data, such as size, length, width, and height.
  - Adversary can develop exploits of this attack to trigger infinite loops or deadlock to break edge-ML systems.

Edge-ML Challenges 1/3

- Diverse edge-ML environments bring about set of challenges in applying machine learning to edge computing.

1. *Computational challenges*.
   - Edge devices usually have very limited computational resource budget for edge-ML model training and deployments.
   - For example, memory on chip of IoT devices is $10^3\times$ and $10^5\times$ smaller than mobile devices and cloud servers.
   - Limited resources make it hard to deploy defenses and countermeasures of ahead-mentioned attacks.
   - Even if defenses can be applicable for edge computing, performance of main tasks will be affected while defenses are used.
   - For example, although TEE solution for model protection can be used for some edge devices, TEE enclaves will consume most of on-chip memory to affect other tasks on devices.

Edge-ML Challenges 2/3

2. *Life-cycle protection challenges*
   - Except for limitation of computational resources, protecting all stages of whole life cycle of edge-ML systems itself is very hard because of large number of unprotected edge devices.
   - Taking into account federated learning paradigm, attackers can easily access local data and models due to lack of protection mechanisms adopted in edge devices.
   - Communication channel between edge devices and servers can be vulnerable which also presents challenges to data and model protection.
   - Hierarchical and collaborative protection schemes are required to secure edge-ML workflow, which is difficult to achieve.

Edge-ML Challenges 3/3

3. *Ethics challenges*
   - Edge computing has become inevitable part of our society and millions of edge-ML systems and models have been deployed, there are several unsolved ethic issues.
   - Owner identification issue exists in edge-ML systems.
   - Collecting various user data without user's consent or permission is critical problem to be addressed.
   - No clear line to differentiate private and public data, where edge devices can collect both public and private data from end-users.
   - Absence of clearly defined boundaries for data leaves attack surfaces to adversaries, and they can develop exploits to attack users from edge-ML system.
   - Even if there are many countermeasures, e.g., differential privacy, that have been proposed to address some of problems, edge-ML ethics is still open question.

Cryptography

- Fundamental stone in edge computing security in various aspects ranging from communication and storage to computation and analysis.
- Difference in available resources in edge devices and servers, strength of cryptographic protection on each device or server varies.

Life Cycle

- Refers to different phases of use of cryptography in and across systems, especially for network security.
- Components of cryptography life cycle include: key management life cycle, cryptosystem de-sign and implementation, protocol deployment and adoption, cryptosystem expiration and revocation, and supply chain.
- Key management life cycle refers to phases of using cryptographic keys, such as key generation, updates, and revocation.
- Cryptosystem management focuses on implementation, adoption, and retirement of protocols themselves, whereas supply chains focus on origin and paths in developing cryptographic libraries and packages.

Cryptography 1/3

Vulnerabilities

Key Management

- Security of cryptographic keys is most crucial component in cryptosystem.
- If adversary is able to obtain symmetric key used by edge device and edge server during communication, all past and future messages encrypted with such key will be under full control of adversary.

Cryptography 2/3

Vulnerabilities

Key Management

2. *Key* R*otation*

- Common cryptographic practice to retire old encryption and signing keys and generate new ones for future communications.
- Reduces number of messages each key is linked to, preventing adversaries from batch-decrypting and compromising transmitted messages in network and providing forward secrecy guarantee for protocol.
- Frequently generating and deriving new ephemeral keys is resource-consuming and may not apply to many edge devices with limited resources.

Cryptography 3/3

Vulnerabilities

Key Management

3. *Key* R*evocation*

- Crucial practice to guarantee forward secrecy and prevent currently unauthorized parties from continuing accessing data within edge network.
- Recorded through key revocation certificate, and certificate revocation list.
- Due to large number of edge devices and servers, and necessity of regularly refreshing keys, it is time- and space- consuming to keep track of every revoked key for all devices and servers on network.

Cryptography 1/3

Vulnerabilities

Cryptosystem management

- As edge network is new computation system framework, there are frequently new cryptographic schemes being proposed to help alleviate vulnerabilities in network security, data protection, and privacy.
- Translating from provable theoretical security to software security is not easy task.

1. *Design and implementation*
   - Studies by Lazar et al. on 269 cryptographic vulnerabilities reported in CVE database have shown that vast majority of cryptographic vulnerabilities come from not bugs in cryptographic libraries but misuse of such libraries and packages during protocol and application development.
   - To securely implement new cryptosystems, need profound cryptographic knowledge and extensive experience in cryptographic software development, as otherwise it can introduce many unnecessary risks.

Cryptography 2/3

Vulnerabilities

Cryptosystem management

2. *Deployment and adoption*

- When new protocol is standardized and ready for public deployment after careful creation process, problem of slow adoption process regarding new protocol.
- For example, TLS 1.3 was introduced in 2018 with significant improvement in performance and security guarantees over previous version, yet its adoption rate has just reached 63% by late 2021.
- Slow deployment process can be caused by public unawareness or lack of compatibility with old devices and systems.
- During communication between edge device and edge server, if edge device does not support TLS 1.3, server will have no choice but to downgrade to using less secure cryptographic algorithms and keys, leaving channel more vulnerable to malicious attacks.

Cryptography 3/3

Vulnerabilities

Cryptosystem management

3. *Expiration and revocation*.
   - With discovery of new attacks and deployment of new cipher suites, certain cryptographic algorithms will be removed from common usage to provide stronger security requirements.
   - For example, TLS 1.3 removed MD5, RSA, and weak elliptic curves from its cipher suite pool.
   - Due to lack of updates or computational resources, old edge devices and servers may be incompatible to run secure new protocols.
   - Known weak ciphers and hashes such as DES, and MD5 may still be used in edge network, leading to simple and effective attacks against intercepting in-network communications.

Cryptography 1/3

Vulnerabilities

Supply Chain

- Security of specific cryptographic library is built upon security assumptions on its dependencies, forming chain of trust.
- If one chain link is broken, all packages and libraries that depend on it can be compromised.
- Vulnerabilities of cryptographic supply chain can emerge from many aspects, including cryptographic libraries themselves, their library dependencies, and package developers.

1. *Outdated cryptographic libraries and packages*
   - As cryptographic protocols are deployed for decades or replaced over time, developers may not retain frequent monitoring, maintenance, and updates for packages.
   - Such libraries do not have timely patches and fixes to respond to newly discovered attacks that affect them, making themselves and their dependency successors susceptible to malicious attacks.

Cryptography 2/3

Vulnerabilities

Supply Chain

2. *Insecure dependency sources*

- Detachment between theoretical cryptographic proofs and real-life cryptosystem development can introduce many vulnerabilities in protocol implementation.
- Such issues may not be reflected in theoretical analysis of protocol, since they do not directly originate from protocol itself.
- Physical side-channel attacks such as One&Done analyze signal activity when performing modular exponentiation to recover RSA secret keys in OpenSSL.
- Heart-bleed bug is caused by missing check in TLS heartbeat extension in OpenSSL, which did not affect other TLS implementations such as GnuTLS and Windows platform implementations.
- These vulnerabilities reside in OpenSSL implementation rather than TLS protocol itself.
- Such vulnerabilities not represented in theoretical protocol analyses but have detrimental effects on actual application of protocol.

Cryptography 3/3

Vulnerabilities

Supply Chain

2. *Untrusted developers*

- Third component of chain of trust falls onto its people – developers behind cryptographic libraries and packages.
- Software developers re-implementing their own cryptographic tools instead of referring to established libraries can produce vulnerabilities at rate three times as much as with non-cryptographic software.
- Complexity of cryptographic software has much larger negative impact on implementation security when compared to non-cryptographic software.
- In vast edge computing network, cryptographic software complexity is substantially higher than other systems.
- Not using established cryptographic libraries and tools created by trusted source can impose much higher chance of introducing additional vulnerabilities into system.

Cryptography

Entropy Management Vulnerabilities

- Most modern cryptographic algorithms require strong and secure randomness to ensure security against various attacks.
- Randomness generated from insufficient entropy can lead to serious compromises, such as prediction of secret keys.
- *Entropy poisoning attacks* can restrict, influence, or give adversary complete control over entropy pool used by devices, causing algorithm outputs to be easily predictable or entirely deterministic in eyes of adversary.

Cryptography 1/3

Quantum Safe Cryptography

- Since data may be stored in edges servers for years or decades from now, one must consider new threats from utilizing quantum computers in not-so-distant future.
- In post-quantum age, many popular and traditionally-ecure asymmetric algorithms (e.g., RSA, DSA, ECDSA, etc.) can be easily broken by quantum computer running Shor's algorithm, while security of symmetric schemes has been drastically decreased by Grover's algorithm.
- Adversary may be equipped with storage capabilities to store communications records among edge devices and servers, then utilize quantum computers to help decrypt and recover sensitive data after.
- Research has been done to investigate methods for building quantum-resistant algorithm that helps alleviate such issues.
- Challenges persist in applying quantum-resistant edge devices in many aspects, such as performance, standardization, and physical security.

Cryptography 2/3

Quantum Safe Cryptography

Post-quantum performance concerns.

- For scheme to be considered secure, quantum-secure cryptosystems usually demand significantly heavier workload on key generation process with much larger key size requirement than in traditional setting.
- For example, most symmetric algorithms such as AES can be considered quantum-safe at cost of doubling key length, but post-quantum asymmetric schemes can have private key length as large as 14000 bits!
- Such conditions impose additional computational burdens on resource-constrained edge devices, limiting scope of their usage and adoption.

Cryptography 3/3

Quantum Safe Cryptography

Post-quantum side-channel attacks.

- There have been designs and optimizations of lightweight post-quantum algorithms applied to edge and IoT devices.
- Even though such schemes are theoretically secure against quantum attacks, implementation of quantum-secure algorithms can still be vulnerable to physical side-channel attacks, as edge devices and servers commonly lack strong physical access control and protection mechanisms.

Cryptography

Research Challenges

- Edge computing architecture contains diverse servers and end devices, which raises many challenges when analyse security of entire system:

1. *Cryptosystem configurations*. When numerous end devices and edge servers encompass cloud, each may support different algorithms, packages, and protocols.
    - Hard to apply analysis to general edge computing framework without missing specific details of individual networks.
2. *Adversarial models*. Different edge-device network configurations can apply different restrictions on possible models of outsider attacks.
    - Need to adjust analysis and mitigation efforts specific to each local network while evaluating security of general edge infrastructure as whole.

Data Security

- In edge computing model, large amount of private data is outsourced to edge servers for computation and storage.
- Information collected from end devices is processed, aggregated, and analysed in multiple levels of edge servers, then transmitted to cloud.
- As data travel through edge server hierarchy, its ownership is transferred from edge devices to many servers.
- Questions on how to guarantee security of device's data in an untrusted environment without having direct control.
- Edge computing aims to provide fast computation and real-time responses to reduce latency of device-server communication, which presents additional layer of restriction on applying common data security measures.

Data Security

Life Cycle

- Data flow within an edge network consists of multiple stages that reflect on different states of data, including:
  - data collection and transmission (*data-in-motion*),
  - data processing and analysis (*data-in-use*), and
  - data storage and recovery (*data-at-rest*).
- Data shifts among three states as it travels through edge network from cloud.
- States interleave with one another, yet each presents unique vulnerabilities in security of gathered data.

Data Security 1/3

General Data Threats and Vulnerabilities

Data in-motion

- Process of data being transmitted to different locations in network.
- In edge computing, this includes data collection from various sources, data sharing among edge servers, and data integration from edge servers to cloud.
- First step of data flow in edge network is collecting input data from different sources.
- Such input can come, for example, from IoT sensor or other edge devices in network.
- Then, data is shared among edge servers during collaborative computation tasks.
- In end, accumulated data is transmitted from edge servers to cloud.
- Ensuring authenticity and integrity of data-in-motion warrants correctness of data as it goes through other stages.

Data Security 2/3

General Data Threats and Vulnerabilities

Data in-motion

1. *Weak authentication*. IoT devices and many edge devices lack sufficient resources to perform advanced authentication algorithms.
   - These operations may be offloaded to edge server to alleviate computational burden of edge devices.
   - As result, data transmitted from devices to edge servers are likely coupled with weak digital signatures or message authentication codes.
   - In extreme cases, authentication mechanisms may be skipped entirely.
   - Incoming messages extremely susceptible to interception and tampering by active attacker controlling network.

Data Security 3/3

General Data Threats and Vulnerabilities

Data in-motion

2. *Fabricated data*. Even with proper message authentication methods, incoming data may still be sent by compromised entity.
   - Corrupted edge device or edge server can send fabricated data signed by valid key to pass verification checks without raising suspicion of other parties on network.

Data Security 1/3

General Data Threats and Vulnerabilities

Data in-use

- Refers to phase where data is being processed by edge server, during which it may be decrypted into plaintext form or remain encrypted for certain operations.
- At this stage, data confidentiality may be violated by outside attacker manipulating server's memory units, or untrusted server extracting partial information.

Data Security 2/3

General Data Threats and Vulnerabilities

Data in-use

1. *Memory-based attacks*. During processing stage, stored data is usually decrypted before being used, which provides adversaries with opening to access decryption keys or plaintext data.
   - In context of edge computing, since edge servers commonly lack strong physical protection, attacker with physical access to edge server can launch memory attacks such as installing RAM scraping malware, executing untrusted functions, and exploiting side channels.

Data Security 3/3

General Data Threats and Vulnerabilities

Data in-use

2. *Encryption leakage attacks.* In cases where edge server is operating over encrypted data (e.g., via searchable encryption), information about underlying plaintext can still be exposed to server via leakage.
   - When performing search, update, and retrieval requests from edge devices, honest-but-curious edge server can record and analyse memory access patterns to determine number and frequency of files accessed.
   - Active server can influence user requests and recover partial plaintext of encrypted data via leakage abuse.

Data Security 1/2

General Data Threats and Vulnerabilities

Data at-rest

- Usually, data is encrypted while being stored in edge servers.
- This does not exempt it from potential exploits.

1. *Data remnants and secure deletion.* Even though data is encrypted and safely stored on disk or removed from server, parts of it may remain in memory from processing stage if memory address has not been overwritten.
   - Adversary can target and recover such data remnants via cold boot attacks.

Data Security 2/2

General Data Threats and Vulnerabilities

Data at-rest

2. *Data backup and recovery*. Since edge servers can possess sufficient resources to perform relatively intensive computation on collected data, data may not be forwarded to cloud until task is done.
   - If stored data is corrupted before being processed and forwarded to cloud server, it can cause severe data loss and service interruption within network.
   - Poses an incentive for attackers to target edge servers and inject ransomware, in exchange for file decryption keys.

Data Security

General Data Threats and Vulnerabilities

Data security

- Cloud providers may implement weak security practices or have no data protection provisions.
- Data are processed or stored in clear text, risking exposure in case of compromise (e.g., case of Accenture).
- Data deletion is another critical issue.
- Research has shown that data persists in memory if not using secure deletion techniques.
- Due to data replications and backups, incomplete data deletion processes might not erase data in cloud infrastructure completely.

Data Security 1/3

ML Data

- In edge-ML systems, key characteristic is that data provision on edge devices and edge/cloud servers can be decoupled so that machine learning models can be trained locally with local data.
- Availability and quality of local data are vital for security of edge machine learning system.

Data Security 2/3

ML Data

1. *Data quality*. Quality of edge-ML data plays important role in training edge-ML models.
   - Many prior works have found that poor data quality can dramatically degrade performance of edge-ML models.
   - To undermine edge-ML systems, data quality attacks, aim to decrease quality of collected data from edge devices by various approaches.
   - For example, genetic attacks and probability-weighted packet saliency attacks have been found to be used to compromise edge intrusion detection systems, where attackers inject large number of low-quality network packets through DDoS attacks.

Data Security 3/3

ML Data

2. *Data availability.* Local data have to be available for local training and evaluation to train edge-ML models in decentralized way.
   - Data availability attacks, are new threats to edge-ML systems, where attackers compromise edge device sensors to impede data collection and curation.
   - For example, onboard sensors are used in autonomous driving systems to collect location data, and attackers can perform electromagnetic pulse attacks to damage electronic sensors and make data unavailable for training autonomous driving models.

Privacy

- Since large amount of sensitive data is uploaded from edge devices to numerous offsite servers, it is essential to ensure that user's privacy is protected from outside threats.

Threats

- As data is transferred, processed, and stored on server, it is susceptible to exploits from both an outside adversary attacking system and network and inside adversary sniffing information from data on server.
- Such exploits can include linkability, identifiability, exposure, and policy non-compliance.

Privacy 1/2

User

Location privacy

- Includes preventing user's location information from tracking and profiling attacks.

1. *Location tracking*. When edge device makes request to edge server, its location information and timestamp can be included as metadata of sent message or as direct functionality.
   - Untrusted server or network attacker can extract and record location data from aforementioned device and map out user's movements over time.

Privacy 2/2

User

Location privacy

2. *Location profiling*. Besides activity tracking, edge device location can be used to establish user profile and further help narrow down or pinpoint user's identity.
   - Attacker can analyse device's most-frequent geographic coordinates and infer user's neighbourhood or workplace.
   - Such auxiliary information can then be applied to anonymized dataset to separate entries that belong to user with high probability.

Privacy 1/3

User

Data privacy

- Focuses on protecting user's personally identifiable information from attacks such as linking, exposing, and de-anonymizing offloaded data.

1. *Data leakage and exposure*. User data in edge servers are vulnerable to thievery or leaked through various methods.
   - If privacy policies and data anonymization techniques are not in place, exposed data may include user's name, address, and contact information, among others.

Privacy 2/3

User

Data privacy

2. *Data linkability.* With an anonymized dataset (e.g. with k-anonymity model), attackers cannot directly identify users from decrypted data.
   - Attacks can still be carried out on such datasets.
   - Edge device may upload same user's data to different edge servers, resulting in user's record existing across several datasets.
   - Adversary can perform links attack by analysing overlapping entries and identifying records that correspond to same individual.

Privacy 3/3

User

Data privacy

2. *Data de-anonymization.* Due to increased deployment of large edge networks in age of "*big data*", there is abundance of datasets to cross-reference against each other.
   - With help of publicly available information, datasets stored on edge servers can be de-anonymized via re-identification attacks.
   - Such attacks can be carried out on privacy-insensitive data or privacy-preserving data analytics that are common practices in medical and business fields.

Privacy 1/2

Policies

- Companies use Privacy policies to disclose how users' data are gathered, used, managed, and disclosed.
- These legal documents are usually extremely vague, lengthy, and complicated to read, which can cause users to skip over them and give unaware consent.
- In premises of edge computing, new threats and vulnerabilities emerge as user data is distributed among edge devices and servers in vast network.

Privacy 2/2

Policies

- **Lack of enforcement**. Without proper non-compliance detection mechanism on each edge node, rogue server can store, process, or disclose unauthorized user data or fail to anonymize data before transmission.
- **Policy conflicts**. Edge devices and servers are deployed in countless locations across globe, where each country or organization has its own policy requirements.
  - As data is transmitted across network, it can move across continents and arrive at destination with policies conflicting with its origin.
  - For example, certain information may be considered publicly accessible, or duration of data storage may have longer time frame.

User and Identity and Access Management 1/7

- Identity and Access Management (IAM) ensures that same identity is managed for all service interactions while simultaneously ensuring security.
- Used to authenticate entity and grant or deny accesses to data and other system resources.
- Large-scale system or service does not maintain its own identity store or authentication mechanism to authenticate.
- IAM makes identity management simpler for large-scale distributed systems.
- Mainly deals with identifying entities and managing access to resources based on pre-established policies.
- Many organizations offer IAM systems, including SailPoint, IBM, Oracle, RSA, and Core security.

User and Identity and Access Management 2/7

- There are number of components related to IAM, including
1. identity management and provisioning,
2. authentication management,
3. federated identity management, and
4. authorization management.
- Those components collaboratively ensure that authorized users are securely and effectively incorporated into cloud.

User and Identity and Access Management 3/7

1. *Diversity of identity information and APIs in edge computing*. Identity management is highly related to security issues of identity management in edge computing.
   - Due to diversity of devices, it is challenging to properly collaborate with all different types of interfaces, including proprietary ones.
   - Competition among major vendors further poses obstacles for uniformed management system.

User and Identity and Access Management 4/7

2. *Forgeable identity of edge devices.* As more devices are out in field and provide more direct public access, verification of submitted identity information becomes critical to correctly enforce further operations of IAM system.
   - For example, preventing identity spoofing becomes much more challenging with edge computing, where mechanism to construct unforgeable identity characteristics is under urgent call.
   - Unlike users and general-purpose computers that share more similar set of identity characteristics (e.g., passwords and fingerprints for human beings, motherboards and OS versions for general-purpose computers), each edge device embeds unique identity characteristics depending on vendor and hardware, including customized and proprietary components.
   - It requires tremendous efforts from both manufacturer and services provider to build chain of trust together for edge device identities.

User and Identity and Access Management 5/7

3. *Rogue identity providers in edge computing.* Just like threats from rogue Certificate Authorities (CAs) in PKIs, similar problems exist in identity providers.
   - Delegating identity management task to trusted third party also means that authenticity of identity and all of credentials are in others' hands, and one can only hope provider remains trusted and benign.
   - In case of rogue identity providers, chain of identity will completely break under IAM.
   - Such threat will even be more daunting and influential under edge computing, if there is any identity service for edge devices, due to uncountable volume of entities that will get affected.

User and Identity and Access Management 6/7

4. *Lack of suitable access control models for numerous heterogeneous devices.* Access control models designed for general-purpose computers and cloud computing follow more uniform pattern thanks to consistency of architecture and form factors.
   - No general model would be satisfying in complex edge computing infrastructure due to diversified systems and their enabled applications, which calls for fine-grained access control that can handle countless scenarios.
   - Not to mention, exhaustively enumerating and making sense of all possible threats itself is an open research question.

User and Identity and Access Management 7/7

5. *Loose network access control.* Often, organizations adopt threat models with weak adversarial assumptions.
   - For instance, they may assume that adversary may not be able to compromise end devices.
   - Access control is very loosely defined within network while enforcing more access control on network perimeters (i.e., firewalls).
   - This becomes major issue once presumably trusted devices mis-behave.
   - For example, compromised IoT device may start probing resources on network or try to spread infection further in network. Such an attack is feasible since access control is non-existent or very loosely defined.

Regulatory Compliance

- Emergence of edge computing and other data-driven technologies (IoT, big data, and cloud platform services) sparked initiatives to regulate and protect end-user cyber rights.
- Vast amount of data are produced by end-users and devices (e.g., cameras, sensors, and smartwatches) and transmitted/processed over edge network.
- These data include personally identifiable information, user actions, habits, health information, etc.
- In edge computing, these data are transmitted to edge devices for fast processing and response, but they might as well be sent further into network (edge servers, cloud).

Regulatory Compliance

- Network nodes could be geographically spread or even on different continents.
- Different privacy regulations may apply depending on where data is transmitted, processed, and stored.
- For instance, European countries have adopted General Data Protection Regulation (GDPR), while state of California in United States has enforced California Consumer Privacy Act (CCPA).
- Other regulations exist to protect sensitive data, such as healthcare data (HIPAA).

Regulatory Compliance

- Edge computing deployments must be compliant with such regulations.
- Given size and distributed nature of infrastructure, this could be challenge.
- Each edge node in infrastructure must only receive, process, and maintain data needed to perform its operations successfully (aka data minimization).
- Data might need to be erased or kept for specific amount of time, depending on effective regulation.
- Data anonymization may need to be applied before transmitting data from one place to another.

Regulatory Compliance

- One must ensure that edge infrastructure and operations comply with any enforced regulations.
- Considering size and geographical distribution of nodes, this could be very challenging:

1. *Regulation Applicability*. Administrators need to identify regulations that pertain to type of data processed in each edge computing node.
2. *Compliance of technical operations*. It requires comparing regulatory requirements with technical device operations.
   - Such operations need to be identified systematically to evaluate compliance accurately and fairly across edge network.

Regulatory Compliance

3. *Compatibility issues*. Regulations may pose compatibility problems across edge network, such as scenario in which two edge devices are in different geographical locations where different laws apply, and both devices have same purpose.
   - Assume that processing requires access to historical healthcare data.
   - Regulation applying to one of edge devices is GDPR, while other device is HIPAA.
   - GDPR protects right to be forgotten and disallows storage of historical information.

Regulatory Compliance

3. *Compatibility issues*.
   - HIPAA does not grant he right to be forgotten; hence historical data are available on device.
   - This is simple scenario that raises compatibility issues.
   - Although edge devices have same purpose, their underlying functionality changes as one device has access to historical data while other is not.

- Ensuring compliance on edge is not trivial. Edge network administrators need to be equipped with tools to aid compliance process or maintain compliance.

Regulatory Compliance

- Need systematic approach for formalizing each device's functionality on edge network.
- Need to map functionality to regulatory requirements affecting different subnetworks of edge network and pinpoint where issues are and how to fix them.