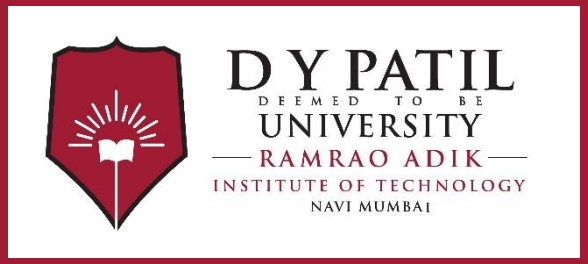



# Information Security

# Module: 4

•Faculty: Mrs. Bhavana Alte
Mr. Prathmesh Gunjgur

# Contents

**Module 4:**

# Lecture 22: Confinement problem

# Confinement Problem

- The confinement problem deals with preventing a process from taking disallowed actions. Consider a client/server situation: the client sends a data request to the server; the server uses the data, performs some function, and sends the results (data) back to the client. In this case the confinement problem deals with preventing a server from leaking information that the user of that service considers confidential.

Access control affects the function of the server in 2 ways

- Goal of service provider: The server must ensure that the resources it accesses on behalf of the client include only those resources that the client is authorized to access.
- Goal of the service user :The server must ensure that it does not reveal the client's data to any other entity not authorized to see the client's data.
- Thus, the confinement problem is the problem of preventing a server from leaking information that the user of the service considers confidential.

# Confinement Problem

- A process that does not store information cannot leak it.This implies that the process cannot do any computations because an analyst could observe the flow of control and deduce information about the inputs.
- A process that cannot be observed and cannot communicate with other processes cannot leak information. This is called total isolation.
- Total Isolation is not practical because processes share CPU, networks, disk storage. Unconfined processes can transmit information over shared resources
- A covert channel is a path of communication that was not designed to be used for communication e.g.

  above until p creates a file called end when q knows that the message isprocess p is confined and cannot communicate with q
  p & q share a file system
  both have read, create, delete privilege to the same directory
  p creates file of length 0 or 1 bit
  q reads the length and then deletes it
  continue the process sent.

# Confinement Problem

- The rule of transitive confinement states that if a confined process invokes a second process, the second process must be as confined as the caller.
- Confinement is a mechanism for enforcing the principle of least privilege. The problem is that the confined process needs to transmit data to another process. The confinement needs to be on the transmission, not on the data access. The confinement mechanism must distinguish between transmission of authorized data and the transmission of unauthorized data. This presents a dilemma in that modern computers are designed to share resources and yet by the act of sharing they create channels of communications along which information can be leaked.
- Even time can be used to transmit information.

  e.g.

  One process can read the time by checking the system clock or counting the number of instructions
  executed.

  A second process can write the time by executing a set number of instructions and stopping

**Module 4:**

# Lecture 23: Formal methods

# Formal methods

- In computer science, formal methods are mathematically rigorous techniques for the specification, development, analysis, and verification of software and hardware systems. The use of formal methods for software and hardware design is motivated by the expectation that, as in other engineering disciplines, performing appropriate mathematical analysis can contribute to the reliability and robustness of a design.
- Formal methods can be used at a number of levels:
  - Level 0: Formal specification may be undertaken and then a program developed from this informally. This has been dubbed formal methods lite. This may be the most cost-effective option in many cases.
  - Level 1: Formal development and formal verification may be used to produce a program in a more formal manner. For example, proofs of properties or refinement from the specification to a program may be undertaken. This may be most appropriate in high-integrity systems involving safety or security.
  - Level 2: Theorem provers may be used to undertake fully formal machine-checked proofs. Despite improving tools and declining costs, this can be very expensive and is only practically worthwhile if the cost of mistakes is very high

# Formal methods

- Dual use of Formal Methods (logical system: syntax, deduction models, semantics)
- build reliable software: give trust and certify software
- Analyse and understand existing software or processes: provide explainability and conformity to expectations
- Chasing bugs in software : bugs are source of vulnerabilities
- • Modeling environment (conforming, adversarial, uncertain, unknown) and expected (mis-)behaviour
- Risk analysis : accidental or intentional threats, faults or attacks
- Recovery mechanisms:
- Certification
- (Safety/Security /Privacy) by design

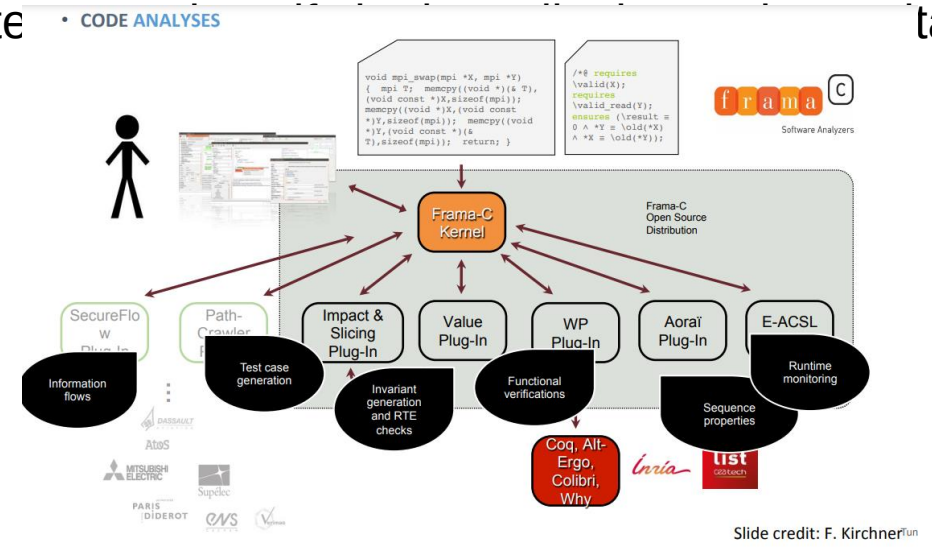
# Formal methods

- with impact on user's trust and empowerment:
- Safety
- Security
- Privacy
- Transparency
- Accountability
- Certification
- Ethics

# Formal methods

Safety

- Formal modeling and verification methods have been successful in improving the safety and security of software in areas such as aeronautics.
- It is essential to improve on the foundations and interconnection of tools and formalisms for interactive and automated program verification.The automation and the expressivity of these tools must be improved so that they can scale to the verification of larger software syste tative properties.



# Formal methods

Security

- Cryptography: confidentiality, integrity, availability, authenticity
- Security Policy: set of rules that specify how sensitive and critical resources are protected
- Prevention: to early detect vulnerabilities
- Cyber-resilience :capacity to tolerate attacks (hardware and software), to detect malware
- Security by design
- cryptographic primitives:mathematical proofs, use theorem proving and program verification to achieve computer checked proofs.
- automated verification tools to analyze the protocol specifications andfind vulnerabilities in the protocol logic producing verified implementations

# Formal methods

Privacy

- Data anonymization: detect private information that can be inferred about individuals using the anonymized data with prior (orbackground) knowledge

- Differential privacy : protect an individual's data while publishing aggregate information. Capability for users to obfuscate their personal data, adding noise by themselves
- Empowering users with personal clouds : individualized management and control over one's personal data. Ensure security and extensibility
- Privacy preserving protocols and communication technologies : Homomorphic and functional encryption to operate on encrypted data.

# Formal methods

Deep Neural Network(example of Transperancy)

- identify weaknesses of deep learning strategies
- Analyse what kinds of attacks are possible and propose mechanisms for protecting convolutional neural networks against adversarial attacks.

- Monitor the internal activations that flow between the layers of a deep network
- make the overall process more robust

Accountability

- Principle which requires that organizations put in place appropriate technical and organizational measures and are able to demonstrate their compliance with the regulation.
- Example and challenges of blockchain

**Module 4:**

# Lecture 24: Evaluating systems

# Evaluating Systems

Evaluation is a process in which the evidence for assurance is gathered and analyzed against criteria for functionality and assurance. The results can be a measure of trust that indicates how well a system meets particular criteria.

Goals of Formal Evaluation

Perfect security is an ultimate, but unachievable, goal for computer security. A trusted system is one that has been shown to meet specific security requirements under specific conditions. This trust is based on assurance evidence.

# Evaluating Systems

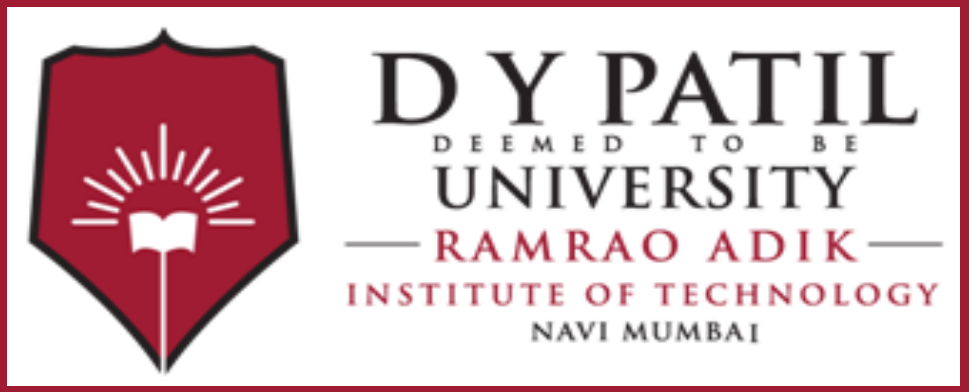
Formal security evaluation techniques were created to facilitate the development of trusted systems. Typically, an evaluation methodology provides the following features:

1. A set of requirements defining the security functionality for the system or product.
2. A set of assurance requirements that delineate the steps for establishing that the system or product meets its functional requirements. The requirements usually include specific evidence of assurance.
3. A methodology for determining that the product or system meets the functional requirements based on analysis of the assurance evidence.
4. A measure of the evaluation result, called a level of trust, that indicates how trustworthy the product or system is with respect to the security functional requirements defined for it.

# Evaluating Systems

3 Standards

- TCSEC – Trusted Computer System Evaluation Criteria
  - first evaluation system
  - composed of a Trusted Computing Base (TCB)
  - government and military driven
  - embraced by business
  - most experience with this method
- ITSEC – Information Technology Security Evaluation Criteria
  - European effort for crypto equipment
  - lead to FIPS 140-1 & 140-2 requirements
  - German Green Book
  - British Criteria
- CC – Common Criteria
  - international effort that built on many previous criteria
  - current evaluation method


D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

# Thank You