APPLYING THE DEPENDABILITY PARADIGM TO COMPUTER SECURITY: THEN AND NOW

Catherine Meadows
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375
meadows@itd.nrl.navy.mil

BACKGROUND OF THIS TALK

- **In 1995, gave a talk on applying the dependability paradigm to security at the New Security Paradigms Workshop**
- **Went through the IFIP WG 10.4 dependability taxonomy and and each point asked the questions:**
  - **What is the security community doing that is relevant to this?**
  - **Could the security community be doing something relevant to this?**
  - **Should the security community be doing something relevant to this?**
- **Pointed out some holes in computer security research**
- **Purpose of this talk: to revisit these points**
  - **Find out what has changed**
  - **Find out what still needs to be done**

HOW DEPENDABILITY IS GUARANTEED

- **A fault is a condition in a system that can lead to failure**
- **To assure dependability:**
  - **Identify the types of failures you are worried about**
  - **Identify the faults that can lead to these failures**
  - **Do some combination of the following**
    - **Fault prevention: Prevent faults from occurring in the first place**
    - **Fault removal: Identify and remove faults after they occur**
    - **Fault tolerance: Build systems tolerant of faults**
    - **Fault forecasting: Estimate incidence of present and future faults**

WHERE WE WERE IN 1995

- **Research in security had concentrated on only part of this approach**
- **Fault prevention**
  - **Use of formal methods, good software engineering practices, testing, etc.**
- **Beginning to see fault identification and removal**
  - **E.g., intrusion detection**
- **Little on fault tolerance or forecast**
  - **Usually limited to worst-case assumptions -- what can go wrong will!**

THESIS OF MY 1995 TALK

- **Concentration on worst-case assumptions a paradigm that is becoming obsolete**
- **Need to develop more sophisticated fault model that can be used to help in**
  - **Containing (tolerating) faults**
  - **Predicting and measuring faults**
- **Three issues:**
  - **Maturing of the field**
  - **Changing emphasis of security research from secrecy to other considerations**
  - **Growing complexity and interconnectivity of computer systems**
- **All three still hold today**

MATURITY

- **Concentration on worst-case assumptions characteristic of a developing field**
  - **Test limits of theory by applying to worst-case assumptions**
  - **Worst-case assumptions simplest to develop and formulate**
- **Limitations appear as theory matures**
  - **Worst-case solutions often impractical to apply**
  - **Infinite extension of worst-case assumptions**

EXAMPLE: INFORMATION FLOW AND COVERT CHANNEL ANALYSIS

- **In a multilevel system, actions of high untrusted processes should be invisible to low processes**
  - **Any way of high affecting low could be exploitable as an illicit (covert) channel**
- **Information flow theories developed to specify systems invulnerable to this kind of attack**
- **History of information flow up to 1995 (greatly condensed)**
  - **Deterministic**
  - **Nondeterministic**
  - **Probabilistic**
- **What's needed: realistic "fault models" of covert channels and methods for evaluating theories in terms of those models**
  - **Example: work now in approximate non-interference**
    - **Measuring difference between noninterfering system and interfering one**

CHANGING EMPHASIS OF COMPUTER SECURITY RESEARCH

- **Early research in computer security concentrated on secrecy**
- **Model used: trusted mechanism controlling access of untrusted subjects to other parts of the system**
- **In theory, secrecy could be obtained in this model, even if untrusted part of system completely hostile, as long as**
  - **Access controls implemented soundly**
  - **Access controls not bypassable**
  - **All covert channels eliminated**

ACCESS CONTROL MODEL NOT AS HELPFUL FOR OTHER PROPERTIES

- **Integrity**
  - **Access control can determine what processes write what data**
  - **Can’t control what is written**
- **Denial of service**
  - **Access control of only limited use in denial of service**
    - **Problem is often in identifying the attacker in the first place**
- **What’s needed**
  - **Ability to recover from and fend off attacks (fault tolerance)**
  - **Ability to predict behavior of attackers and likely attacks (fault prevention)**

GROWING COMPLEXITY AND INTERCONNECTION

- **Systems don’t exist in isolation**
- **In many ways a system will be connected to and rely upon services of other systems less than completely trustworthy**
  - **But not completely untrustworthy, either**
- **Need ways of identifying way in which components of a large distributed system can fail**

OUTLINE OF A FAULT MODEL FOR SECURITY

- Faults in the security mechanism
- Hostile attacks on a system
- Misuse of a system, e.g.
  - Bad choice of passwords
  - Incorrect setting of security parameters
  - Opening attachments on email from unknown sources
  - Entrenchment of systems with known security problems
  - Etc.

SECURITY FAILURE CAN BE THE RESULT OF INTERACTION OF A NUMBER OF SYSTEM FAULTS

- **Computers without proper access controls (system fault)**
- **Users who open attachments on email from unfamiliar sources (human misuse)**
- **Writers of hostile self-replicating code (hostile attack)**

Adds up to the virus problem

Fault Forecast and Security

- **Faults in the security mechanisms**
  - **Likelyhood that a fault will exist**
  - **Difficulty of taking advantage of a fault**
  - **Second is better understood than the first**
    - **Examples**
      - **Capacity of a covert channel**
      - **Amount of effort involved in breaking a cryptosystem**
- **Human misuse**
  - **Can perform studies that will get this information**
- **Hostile attack**
  - **Data much harder to get, although information available on types of attacks that have occurred in the past**
  - **Parameters include: resources available, willingness to expend resources, goals of attacker**

FAULT TOLERANCE AND SECURITY

- **Fault tolerance permeates security**
  - Multilevel secure systems tolerate Trojan Horses
  - Key distribution protocols tolerate hostile intruders with complete control of network
  - Secret sharing schemes tolerate dishonest trustees
  - Secure DBMSs tolerate those trying to infer sensitive data
- **In most cases**
  - Faults tolerated limited to hostile attack
  - Concentrated on worst-case scenarios
  - Includes well-delineated boundary that can’t be crossed

OTHER POSSIBILITIES FOR FAULT TOLERANCE AND SECURITY

- **Tolerance of misuse**
  - **Protocols to mitigate bad effects of choosing weak passwords**
  - **Heuristics for cryptographic algorithms making them easier to implement and use**
- **Tolerance of “ankle-biter” attacker**
  - **Use of honeypots to distract intruders**
- **Tolerance of failure of mechanisms**
  - **Use of multiple encryption algorithms**

Open Questions

- **What do you do with faults you can’t forecast reliably?**
- **How does including security affect the dependability paradigm?**
- **How do we take into account changing abilities and goals of attackers?**

BACK TO THE 21st CENTURY

SOME NEW PARADIGMS

- **Intrusion Tolerance**
  - **Treat intrusion as a fault**
  - **Take similar architectural approach as in classical fault tolerance**
    - **Distribute information over different components of a system**
    - **Intruder may be able to access or damage a component of the system, but this will not allow it to access sensitive data**
- **Survivability**
  - **Define mission of a system**
  - **Concentrate on fulfilling mission even in presence of failure of system components**
    - **Failures may have different causes such as attack, accident, etc.**
  - **Note that mission fulfillment not the same as correct operation**
    - **Need to separate critical from non-critical requirements**

WHERE THIS LEAVES US

- **Fault-tolerance now added to fault prevention and removal in the computer security toolbox**
- **Comes in two flavors**
  - **Maintaining normal operation in face of attack**
    - **Example: Web-based service maintaining normal operations in face of denial of service attacks**
  - **Maintaining critical functions in face of effort to destroy or hobble system**
    - **Example: maintaining the ability to perform funds transfer in face of attempt to shut down the nationwide banking network**

BUT WHAT ABOUT FAULT FORECAST?

- **Still a hard problem**
- **Still not much on predicting security flaws or human misuse**
- **Predicting intrusions even harder**
- **One approach: rely on information from previous attacks**
  - **Approach of pattern-based intrusion detection**
- **Some open problems in fault forecast for security**
  - **Predicting human misuse**
  - **Predicting nature of attacks based on system assets and mission**
  - **Using fault forecast to help in formulating security strategy**
    - **Identify parts of system likely to be come under attack**
    - **Concentrate resources on protecting them**
  - **Determining the nature of an attack in its early stages**
    - **Is it an attack or not?**
    - **What are its goals?**
    - **How severe is the attack?**

CONCLUSION

- **Security getting closer to exploiting options offered by full dependability paradigm**
  - **Seeds for much of this already present in early work**
- **More than one way of applying dependability paradigm, depending on the nature of the problem**
- **Fault forecast still an open problem**