

Vulnerabilities and Threats in Distributed Systems

Bharat Bhargava (PI) – <http://www.cs.purdue.edu/people/faculty/bb/>
Leszek Lilien (co-PI) – <http://www.cs.wmich.edu/~llilien/>



Research Program

1. Vulnerability Models
2. Threat Models
3. Mechanisms to Reduce Vulnerabilities & Threats
 - 3.1. Applying Reliability and Fault Tolerance Principles to Security Research
 - 3.2. Privacy-preserving Data Dissemination

Models of Threats

- Threats in security domain – like errors in reliability domain
 - Entities that can intentionally exploit or inadvertently trigger specific system vulnerabilities to cause security breaches
 - Attacks or accidents materialize threats
- Threat **classifications**:
 - Based on *actions*, we have: threats of illegal access / threats of destruction / threats of modification / threats of emulation
 - Based on *consequences*, we have: threats of disclosure / threats of (illegal) execution / threats of misrepresentation / threats of repudiation

Models for Vulnerabilities

- A vulnerability in security domain – like a fault in reliability domain
- **Modeling** vulnerabilities
 - Features of vulnerabilities
 - Classifying vulnerabilities
 - Taxonomies of vulnerabilities
 - Providing formalized models of vulnerabilities
 - Analysis of four common computer vulnerabilities
 - Vulnerability lifecycle model applied to three case studies
- **Model-based analysis** to identify configuration vulnerabilities
- Some **conclusions**:
 - Not all vulnerabilities can be removed, and some vulnerabilities shouldn't be removed
 - Need threat assessment to decide which vulnerabilities to remove first
 - System design should not let an adversary know vulnerabilities unknown to the system owner

Threat Research Issues

- Analysis of known threats in context
 - Identify known threats for the context
 - Find salient features of these threats and associations between them
 - Build a threat taxonomy for the considered context
 - Propose qualitative and quantitative models of threats in context
 - Define measures to determine threat levels
- Develop quantitative threat models using analogies to reliability models
- Propose evaluation methods for threat impacts
- Invent algorithms, methods, and design guidelines to reduce number and severity of threats
- Study threat detection

Vulnerability Research Issues

- Analyze severity of a vulnerability and its potential **impact on an application**
- Provide procedures and methods for efficient extraction of **characteristics and properties** of known vulnerabilities
- Construct comprehensive **taxonomies** of vulnerabilities for different application areas
 - Good taxonomies facilitate both prevention and elimination of vulnerabilities
- Enhance structure of metabases of vulnerabilities and incidents
- Provide models for vulnerabilities and their **contexts**
- Devise quantitative **lifecycle vulnerability models** for a given type of application or system
 - The lifecycle models helps solving some vulnerability problems

Industry Research Programs, Products, and Services

- Some **research programs**
Microsoft Trustworthy Computing (Security, Privacy, Reliability, Business Integrity), IBM Almaden (information security), IBM Zurich (information security, privacy, and cryptography),...
- **Commercial and free products & services**
- Some vulnerability & incident metabases
CVE (Mitre), ICAT (NIST), OSVDB (osvdb.com), Apache Web Server (Red Hat), Cisco Secure Encyclopedia (Cisco), DOVESCComputer Security Laboratory (UC Davis), DragonSoft Vulnerability Database (DragonSoft Security Associates), Secunia Security Advisories (Secunia), SecurityFocus Vulnerability Database (Symantec), SIOS (Yokogawa Electric Corp.), Verletzbarkeits-Datenbank (scip AG), Vigil@nce AQL (Alliance Qualit e Logiciel), ...
- Some vulnerability **notification systems**
CERT (SEI-CMU), Cassandra (CERIAS-Purdue), ALTAIR (esCERT-UPC), DeepSight Alert Services (Symantec), Mandrake Linux Security Advisories (MandrakeSoft), ...
- Some **other tools**
 - Vulnerability Assessment Tools
AppDetective (Application Security), NeoScanner@ESM (Inzen), AuditPro for SQL Server (Network Intelligence India Pvt. Ltd.), eTrust Policy Compliance (Computer Associates), Foresight (Cubico Solutions CC), IBM Tivoli Risk Manager (IBM), STAT-Scanner (Harris Corp.), StillSecure VAM (StillSecure), Symantec Vulner. Assessment (Symantec), ...
 - Automated Scanning Tools, Vulnerability Scanners
Automated Scanning (Beyond Security Ltd.), iPLegion/intraLegion (E*MAZE Networks), Managed Vulnerability Assessment (LURHQ Corp.), Nessus Security Scanner (The Nessus Project), NeVO (Tenable Network Security), ...
 - Vulnerability and Penetration Testing
Attack Tool Kit (Compute.ch), CORE IMPACT (Core Security Technologies), LANPATROL (Network Security Syst.), ...
 - Intrusion Detection/Prevention Systems
Cisco Secure IDS (Cisco), Cybervision Intrusion Detection System (Venus Information Technology), Dragon Sensor (Enterasys Networks), McAfee IntraShield (IDS/McAfee), NetScreen-IDP (NetScreen Technologies), Network Box Internet Threat Protection Device (Network Box Corp.), ...
 - Threat Management Systems
Symantec ManHunt (Symantec), ...
- Some **services**
 - Vulnerability Scanning Services
Netcraft Network Examination Service (Netcraft Ltd.), ...
 - Vulnerability Assessment & Risk Analysis Services
ActiveSentry (Intranode), Risk Analysis Subscription Service (Strongbox Security), SecuritySpace Security Audits (E-Soft), Westpoint Enterprise Scan (Westpoint Ltd.), ...
 - Threat Notification
TruSecure IntelliSHIELD Alert Manager (TruSecure Corp.), ...
 - Patches
Software Security Updates (Microsoft), ...

Applying Reliability Principles to Security Research

- Apply Reliability ideas/solutions to Security
- Analogies in **basic notions**
 - Fault \leftrightarrow vulnerability
 - Error \leftrightarrow threat (enabled by a fault / enabled by a vulnerability)
 - Failure/crash \leftrightarrow Security breach (materializes a fault, consequence of an error / mater. a vulnerability, conseq. of a threat)
- **Time - effort** analogies: time-to-failure distribution for accidental failures \leftrightarrow expended effort-to-breach distrib. for intentional sec. breaches
 - Not a "direct" analogy: it considers important differences between Reliability and Security
 - Most important difference: intentional human factors in Security
- Analogies to **fault avoidance/tolerance**
 - Fault avoidance - threat avoidance
 - Fault tolerance - threat tolerance (gracefully adapts to materialized threats)
 - Maybe *threat* avoidance/tolerance should be named: *vulnerability* avoidance/tolerance (to be consistent with the vulnerability - fault analogy)
- Analogy: **Fault-tolerant syst.** deal with failures Build **vulnerability-tolerant systems** to deal with security breaches
- Caveat: **Reliability analogies not always helpful**
 - Differences between seemingly identical notions
 - No simple analogies exist for intentional security breaches arising from planted malicious faults
 - No simple analogies exist when attack efforts are concentrated in time

Privacy-Preserving Data Dissemination

- Data dissemination **problem**
 - Owners entrust sensitive data to *guardians*
 - Guardians allowed / required to disseminate data with owner's explicit consent, or w/o it
 - Owner's privacy preferences might be lost
Risk grows with length of the guardian chain, and milieu fallibility & hostility
If preferences lost, even honest receiving guardian unable to honor them
 - Sensitive data disseminated, accidentally or intentionally, to entities that should not see them
- Proposed **solution**:
 - Use **bundles** to make data and metadata inseparable
bundle = self-descriptive private data + its metadata
 - Each bundle includes mechanism for **apoptosis**
apoptosis = clean self-destruction
 - Bundle chooses apoptosis when threatened with a successful hostile attack
 - Develop distance-based bundle **evaporation**
Distance metrics not just in geographical terms (e.g., one's insurer is "more distant" than doctor)
The more "distant" from its owner is a bundle, the more it evaporates (becoming more distorted)

Conclusions

- 20 years of research in Reliability can form a basis for vulnerability and threat studies in Security
- Need to quantify threats, risks, and potential impacts on distributed applications
- Adapt and use resources to deal with different threat levels
- Government, industry, and the public are interested in progress in this research

Selected Publications

- A. Bhargava and B. Bhargava, "Applying fault-tolerance principles to security research," *IEEE Symp. on Reliable Distributed Systems*, New Orleans, Oct. 2001.
- B. Bhargava, "Security in Mobile Networks," *NSF Workshop on Context-Aware Mobile Database Management (CAMM)*, Brown University, Jan. 2002.
- B. Bhargava, "Vulnerabilities and Fraud in Computing Systems," *Intl. Conf. IPSI*, Sv. Stefan, Serbia and Montenegro, Oct. 2003.
- B. Bhargava, S. Kamisetty and S. Madria, "Fault-tolerant authentication and group key management in mobile computing," *Intl. Conf. on Internet Computing*, Las Vegas, June 2000.
- B. Bhargava and L. Lilien, "Vulnerabilities and Threats in Distributed Systems," *Intl. Conf. on Distributed Computing & Internet Technology (ICDCIT 2004)*, Bhubaneswar, India, Dec. 2004, pp. 146-157.
- B. Bhargava, Y. Zhong, and Y. Lu, "Fraud Formalization and Detection," *Intl. Conf. on Data Warehousing & Knowledge Discovery DaWaK-2003*, Prague, Czechia, Sep. 2003.
- L. Lilien and A. Bhargava, "From Vulnerabilities to Trust: A Road to Trusted Computing," *Intl. Conf. on Advances in Internet, Processing, Systems, and Interdisciplinary Research (IPSI-2003)*, Sv. Stefan, Serbia and Montenegro, Oct. 2003.
- L. Lilien and B. Bhargava, "A scheme for privacy-preserving data dissemination," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 36(3), May 2006, pp. 503-506.
- Y. Zhong, Y. Lu, and B. Bhargava, "Dynamic Trust Production Based on Interaction Sequence," Tech. Rep. CSD-TR 03-006, Dept. Comp. Sciences, Purdue Univ., Mar. 2003.

NSF Cyber Trust Principal Investigators Meeting
January 28-30, 2007
Atlanta, Georgia