

## Formal Assessment Instrument for Ensuring the Security of NASA's Networks, Systems and Software

**David P. Gilliam**

*Jet Propulsion Laboratory,  
California Institute of Technology  
[David.P.Gilliam@jpl.nasa.gov](mailto:David.P.Gilliam@jpl.nasa.gov)*

**John D. Powell**

*Jet Propulsion Laboratory,  
California Institute of Technology  
[John.Powell@jpl.nasa.gov](mailto:John.Powell@jpl.nasa.gov)*

**Josef Sherif**

*Jet Propulsion Laboratory,  
California Institute of Technology  
[Josef.Sherif@jpl.nasa.gov](mailto:Josef.Sherif@jpl.nasa.gov)*

### **Abstract**

*Today there are a multitude of applications that are network aware and which often provide services including login credentials. A key issue is the security of these applications. Do they inadvertently allow compromise of the system on which they reside or the data on those systems? What insures that the applications are not vulnerable to common problems such as race conditions or buffer overflows, and other potential security problems?*

*Security vulnerabilities in software on networked systems provide attackers an avenue to penetrate those systems. The source of these security weaknesses are usually traced to poor software development practices, non-secure links between computing systems and applications, and mis-configurations. An otherwise secure system can be compromised easily if a system or application software on it, or on a linked system, has vulnerabilities. Currently, there is a lack of security assessment tools for use in the software development and maintenance life cycle to mitigate these vulnerabilities.*

*To address this problem, the National Aeronautics and Space Administration (NASA) has funded the Jet Propulsion Lab in conjunction with the University of California at Davis (UC Davis) to begin work on developing a software security assessment instrument for use in the software development and maintenance life cycle, which is composed of 5 activities:*

- *A Vulnerability Matrix (VMatrix)*
- *Security Assessment Tools (SATs)*
- *Property Based Tester (PBT)*
- *Model Based Verification (MBV) with the use of a Flexible Modeling Framework (FMF)*
- *Software Security Checklist (SSC)*

*To date 4 critical activities have been delivered and are being made available to the internet community: 1) VMatrix, 2) SATs, 3) PBT, and 4) A Report on MBV methodology for security.*

*The vulnerability database is an outgrowth of a vulnerability matrix developed for NASA. Its purpose is to provide information about various vulnerabilities including the exploit used to gain access, how to protect against the exploit and the Common Exposures and Enumeratives (CVE) listing. The information is being transferred to the UC Davis Database Of Vulnerabilities, Exploits, and Signatures (DOVES) where it will be maintained and updated as new exploits are discovered. This information is used to extract properties and requirements that express potential network vulnerabilities. These properties can then be utilized by the PBT tool and the FMF.*

*The SATs are a collection and an assessment of publicly available software security code checking tools available on the Internet that can be used to test for potential weaknesses of software code. This list includes a description of each of the tools and their uses. It will be updated as additional tools become available.*

*The PBT tool performs formal verification of properties, including those obtained from the vulnerability matrix, at the code level. Properties are verified by slicing the code in search of the specific vulnerability properties in question.*

*Like the PBT tool, the FMF aids in formally verification of properties over an abstract model of the system. The FMF performs this function at the abstract level before code exists.*

*The SSC has two foci: 1) a checklist to verify that software released by NASA does not provide users backdoors or encrypted channels into NASA networks or provide other information about NASA's systems or networks (such as IP Address space); 2) a potential checklist for code developers to write secure code for network aware applications, such as use of network ports, protocols, authentication, privileges, etc.*

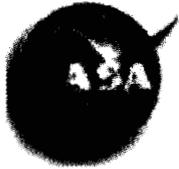
*The assessment instrument is a comprehensive set of tools that can be used individually or together to ensure the security of network aware software application and systems. Using the various tools together provide a distinct advantage. Each tool's resulting output provides feedback into the other tools. Thus, more comprehensive assessment results are attained though the leverage each tool provides to the other when they are employed in concert.*

*An ongoing effort is underway with the Multi-Mission Encryption Communication System (MECS) to pilot the usage of this security assessment instrument.*

### **Acknowledgement**

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The work was sponsored by the NASA Office of Safety and Mission Assurance under the Software Assurance Research Program lead by the NASA Software IV&V Facility. This activity is managed locally at JPL through the Assurance and Technology Program Office.

*For further information about this ongoing research, refer to <http://rssl.jpl.nasa.gov>*



# **Integrated Approach to Assuring Software Security**

**David Gilliam, John Powell, Josef Sherif**  
**California Institute of Technology,**  
**Jet Propulsion Laboratory**

**Matt Bishop**  
**University of California at Davis**

**California Institute of Technology, Jet Propulsion Lab** **JPL**



# NASA RTOP: Reducing Software Security Risk

---

- **NOTE:**
  - **This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.**
  - **The work was sponsored by the NASA Office of Safety and Mission Assurance under the Software Assurance Research Program lead by the NASA Software IV&V Facility.**
  - **This activity is managed locally at JPL through the Assurance and Technology Program Office.**



# Collaborators

---

- **David Gilliam – Principle Investigator  
Network and Computer Security, JPL**
- **John Powell – Research Engineer  
Quality Assurance, JPL**
- **Josef Sherif – Research Engineer  
Network and Computer Security, JPL**
- **Matt Bishop – Associate Professor of Computer  
Science, University of California at Davis**

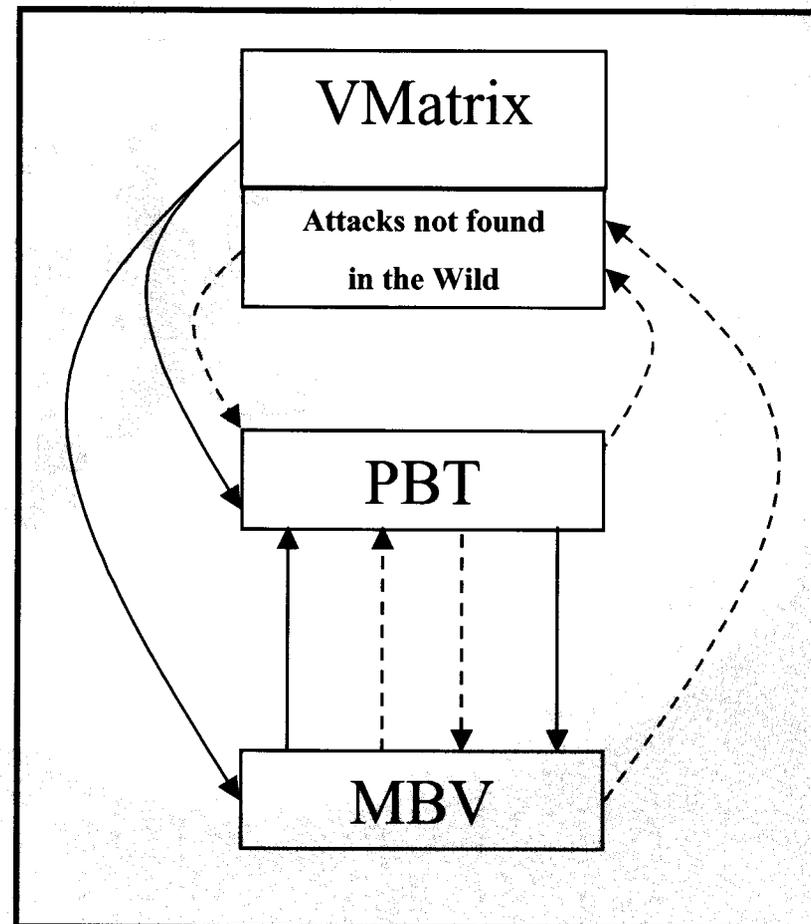




# An Integrated Approach

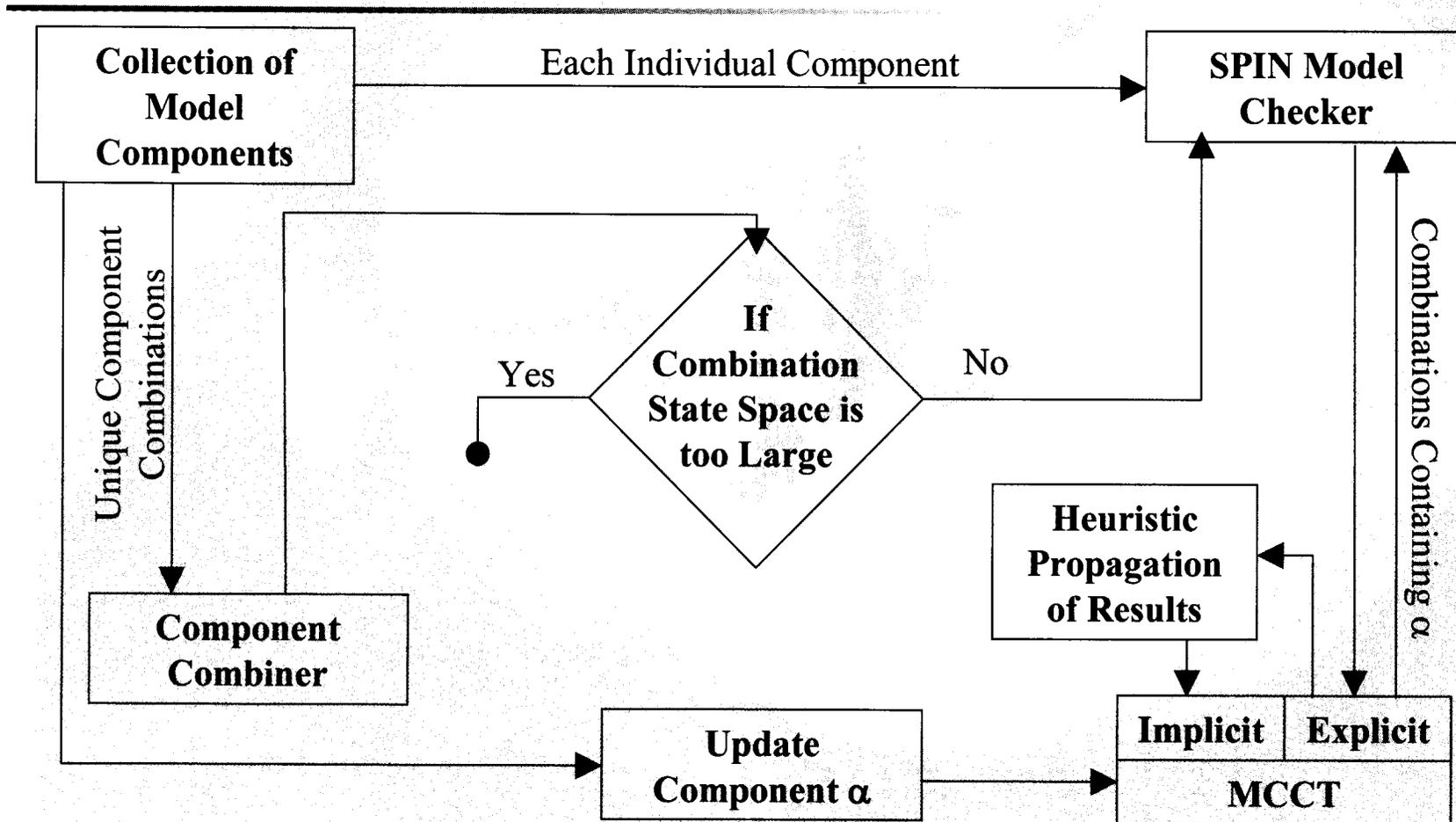
Each part of the instrument supports the other parts

- VMatrix provides known vulnerability properties to PBT and MBV
- PBT provides newly discovered code vulnerabilities to the Vmatrix property set and code level verification feedback to MBV
- MBV provides newly discovered vulnerabilities scenarios to the Vmatrix property set and early lifecycle verification results to PBT for tracability purposes





# Flexible Modeling Framework

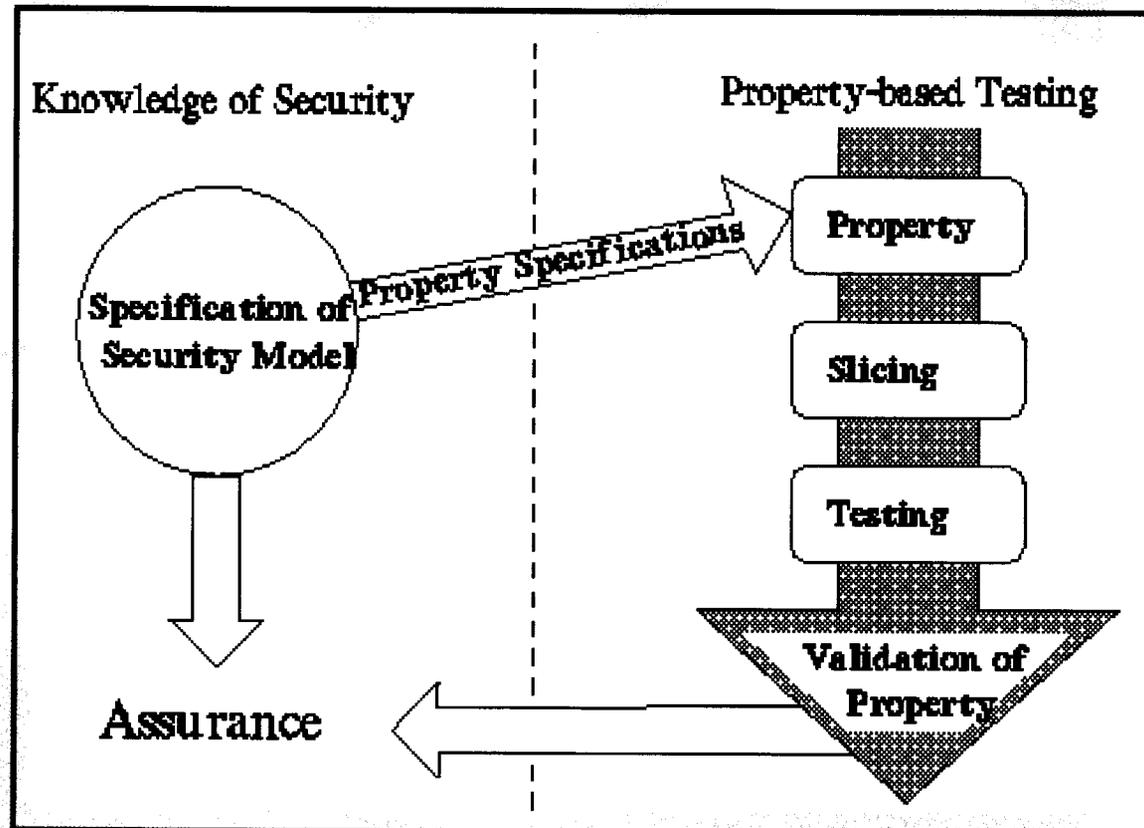




# PBT - How It Works

## Compare program actions with specifications

- Create low-level specifications
- Instrument program to check that these hold
- Run program under run-time monitor
- Report deviations





# Vulnerability Matrix

---

- **Vulnerability matrix to assist security experts and programmers where best to expend their efforts**
  - **DOVES database:**  
<http://seclab.cs.ucdavis.edu/projects/vulnerabilities/>
  - **Uses the Common Vulnerabilities and Exposures (CVE) dictionary ID**
  - **Ranks vulnerabilities based on severity versus frequency**
  - **Contains signatures used to exploit the vulnerability – signatures used with the Tester's Assistant**
  - **Maintained by UC Davis and SANS**



# Software Security Checklist (SSC)

---

- **The Software Security Checklist (SSC) will:**
  - **Provide software code developers with an instrument for writing secure code for network aware applications, such as the use of network ports, protocols, authentication, privileges, etc...**
  - **Ensure that released software does not provide backdoors into or information about an organization's systems or networks**



# Security Assessment Tools (SATs) and Checklist

---

- **Security Assessment Tools (SATs):**
  - **Collection of tools for assuring the security of the systems and software on them**
  - **Property Based Testing Tool (PBT) – UC Davis (Matt Bishop)**
    - **Code insertion for testing properties**
    - **Verify the properties are not violated after compiled**
    - **JAVA based – look to port to C, C++, Perl, Mobile Code**
  - **Tool list maintained by UC Davis**