Telecom Equipment Assurance Testing

T.V.Prabhakar,
Gopi Krishna S Garge,

Indian Institute of Science
Bangalore

Agenda

- Overview of the TETC
- Security Testing & requirements
- Security Standards?
- Is there a formalism to what we want?
- Can TTCN 3 help?
- Discussion

Our Mission

- Telecom space
  - Telecom includes data networking; focus on DN
  - Equipment acceptance tests
  - Security Evaluation
  - Safe-to-connect certification
  - Publish guidelines for procurers and OEMs

Objectives

- Set up an assurance test facility
- Tests include
  - Telecom Equipment (untrusted)
  - Detect hidden malicious code/systems within
  - Other h/w and s/w weaknesses that may exist
- Set up contractual terms for suppliers
- Review the requirements of such assurance facilities

Assurance Testing

- Product and System assurance
- Suite of tests
  - Vulnerability Analysis
  - Penetration Testing (BB and fuzzing)
  - Deep Inspection (source code, processes, etc.)
  - Non-functional tests, SVCT, MVCT, etc.

Assurance Framework

- Common criteria (adapted?)
  - Criteria, methodology and recognition for IT security evaluation
  - Protection Profiles
  - Security Targets
  - Testing and Evaluation
- Can we use TTCN 3 in such a context?

Security Evaluation

- Risk estimation and Deployment Targets
- What to protect?
- What protection to evaluate?
- Formal Representation? Grammar?
- Translate to a spec language?
- Derive test suites?
  - Code for execution
  - Code inclusion
  - Verdict/security level quantification

Security Tests

- Compliant vs Vulnerable
- Test Design
  - SUT
    - Load Conditions
    - Responses
    - Graceful degradation/recovery
  - Attack Parameters
    - Persistent vs non-persistent
    - low/med/high persistence
    - Single vs multiple attacks
    - Detection avoidance

TTCN-3 Applications

- Mobile communications
  - LTE, WiMAX, 3G, TETRA, GSM
- Broadband technologies
  - ATM, DSL
- Middleware platforms
  - WebServices, CORBA, CCM, EJB
- Internet protocols
  - SIP, IMS, IPv6 and SIGTRAN
- Smart Cards
- Automotive
  - AUTOSAR, MOST, CAN

Security Standards

- ETSI and the eEurope programme – 2005
- STF 356 – Making better security standards
- 4th ETSI Security Workshop
  - EG 202 387, Common Criteria
  - ES 202 382, Protection Profile
  - ES 202 383, Security Target

Security Standards

- Any requirement should be testable
- Any security requirement must be testable AND must achieve its security objective
- Open development of crypto has been the norm for a number of years (AES for example)
- Security systems need to be open to examination
- Assurance evaluation schemes fit the model
- Designing in anticipation of assurance evaluation is good practice

Security Standards

- Risk analysis is still top of the process tree
- Objectives still have to be established before requirements
- Crypto based solutions by themselves don’t provide security

Security Testing

- Telecom equipment security testing means:
  - Equipment is free from vulnerabilities
    - DOS, Buffer overflow, Remote Code Execution, Format string, Malloc bombs, ..
  - Equipment is free from virus and malware
  - Equipment is recommended for “safe to connect”

Security testing approaches

- Several approaches are possible:
  - Attack the equipment and observe its capability to withstand or mitigate the attack
    - Attack heuristics can be developed
  - Perform a black box robustness testing and look for implementation level security
    - Design test cases
  - Complete coverage of the input space
  - Monitor traffic with a sniffer and analyze the data with appropriate filters
  - Monitor a deviation from the baseline – anomaly detection?

Security testing

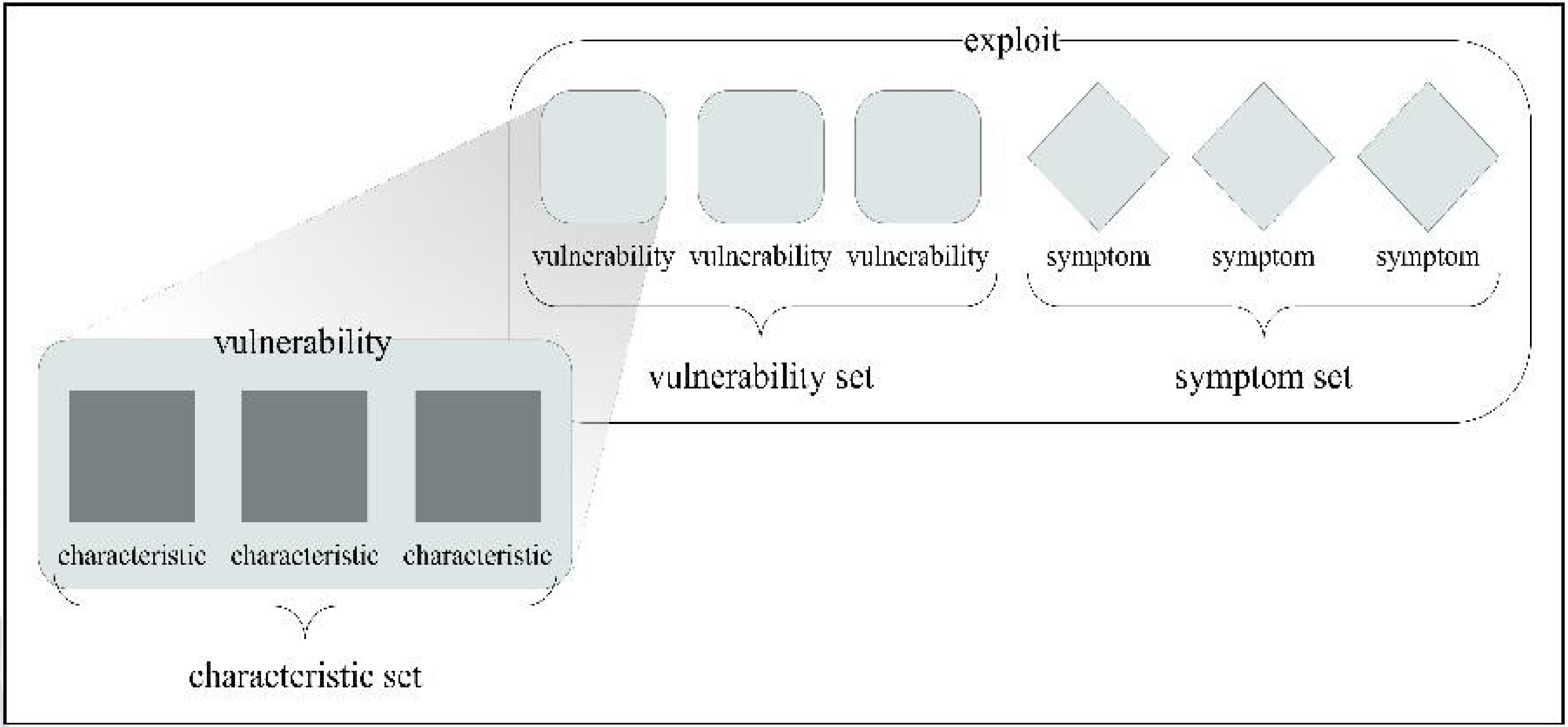
- TTCN-3 based security test suites, when done, have to be made publicly available
- Threat and Risk perceptions master script
  - Recommends the actual scripts that are required to be run
- Certification scripts adhering to security standards are urgently required
  - Common Criteria based Protection profiles will be invaluable
- Client-Server/ Peer scripts to maintain security assurance of production equipment
  - Eg: Impact of opening a firewall’s port on core router

Using TTCN 3

- Grammar for expressing network policy violations
- Representation of exploits as action sequence trees
- Compliance vs Vulnerability
- Stateful protocols
- Synchronization

The formalism available: An Illustration

Terminology



Characteristic

Characteristic Grammar Definitions

MIS-CONFIGURATION	Device or software configuration violates network policy
MIS-SPECIFICATION	Intended semantics not captured by protocol specification, example: mandating broken encryption schemes
FLAW	An unintended condition present in a system
SOFTWARE FLAW	Application or operating system violates protocol design
HARDWARE FLAW	Hardware violates protocol design
AUTHENTICATION FLAW	Protocol verifies identities insufficiently or not at all
AUTHORIZATION FLAW	Protocol verifies permissions insufficiently or not at all
NONCE FLAW	Protocol binds data to session non-uniquely or not at all
STATE MACHINE FLAW	Protocol specification tracks internal state insufficiently or not at all, enabling flooding and brute forcing

Characteristic Grammar

<characteristic> ::= MIS-CONFIGURATION | MIS-SPECIFICATION | SOFTWARE FLAW
| HARDWARE FLAW | AUTHENTICATION FLAW | AUTHORIZATION FLAW
| NONCE FLAW | STATE MACHINE FLAW

Symptoms

Symptom Grammar Definitions

USER CREDENTIALS	User data used to create logical host connections
USER DATA	Information created by a user
HOST CREDENTIALS	Host data used to create logical host connections
HOST DATA	Information created by a user or host
HOST SERVICE	A user or operating system process on a host
CONNECTION	Logical or physical connection for data exchange between hosts
BANDWIDTH	Channel capacity of logical or physical connection
DIVERT	To change ownership, possibly transparently
DISABLE	To make unavailable, partially or completely
SNIFFING	Acquiring broadcast or non-broadcast user data
IMPERSONATING	Using user or host credentials of others
CONNECTION HIJACKING	Taking a connection and associated credentials from their owner
MAN IN THE MIDDLE	Intercepting connections between other hosts
DENIAL OF SERVICE	Disabling a host service or connection
FLOOD	Impeding or disabling a connection by saturating bandwidth

Symptom Definitions

Symptom Grammar

<symptom>	::=	DIVERT <resource> | DISABLE <resource>
<resource>	::=	<user resource> | <host resource> | <network resource>
<user resource>	::=	USER CREDENTIALS | USER DATA
<host resource>	::=	HOST CREDENTIALS | HOST DATA | HOST SERVICE
<network resource>	::=	CONNECTION | BANDWIDTH
<sniffing>	::=	DIVERT USER DATA
<impersonating>	::=	DIVERT USER CREDENTIALS | DIVERT HOST CREDENTIALS
<connection hijacking>	::=	DIVERT CONNECTION $\wedge$ DIVERT HOST CREDENTIALS
<man in the middle>	::=	<connection hijacking>$^{+}$
<flood>	::=	DISABLE BANDWIDTH
<denial of service>	::=	DISABLE HOST SERVICE | <man in the middle> | <flood>

Vulnerability

Vulnerability Grammar

<vulnerability> ::= MIS-CONFIGURATION$^+$ | <implementation>$^+$ | <design>$^+$

<implementation> ::= SOFTWARE FLAW | HARDWARE FLAW

<design> ::= MIS-SPECIFICATION | STATE MACHINE FLAW | <forgery>

<forgery> ::= AUTHENTICATION FLAW | AUTHORIZATION FLAW | NONCE FLAW

Exploit

Exploit Grammar

<exploit> ::= <vulnerability>$^+$ <symptom>$^+$

Algorithm

$\textsc{Is-Secure}\,(\text{Protocol } P,\ \text{Implementation } I,\ \text{Policy } \rho)$

$secure \leftarrow \neg\textsc{Is-MisConfigured}\,(I, \rho)$

$secure \leftarrow secure \wedge \neg\textsc{Is-MisSpecified}\,(P)$

$secure \leftarrow secure \wedge \neg\textsc{Has-Implementation-Flaw}\,(I)$

$secure \leftarrow secure \wedge \neg\textsc{Has-State-Machine-Flaw}\,(P)$

for each packet type $p \in P$

$\quad secure_p \leftarrow \neg\textsc{Has-Authentication-Flaw}\,(p)$

$\quad secure_p \leftarrow secure_p \wedge \neg\textsc{Has-Authorization-Flaw}\,(p)$

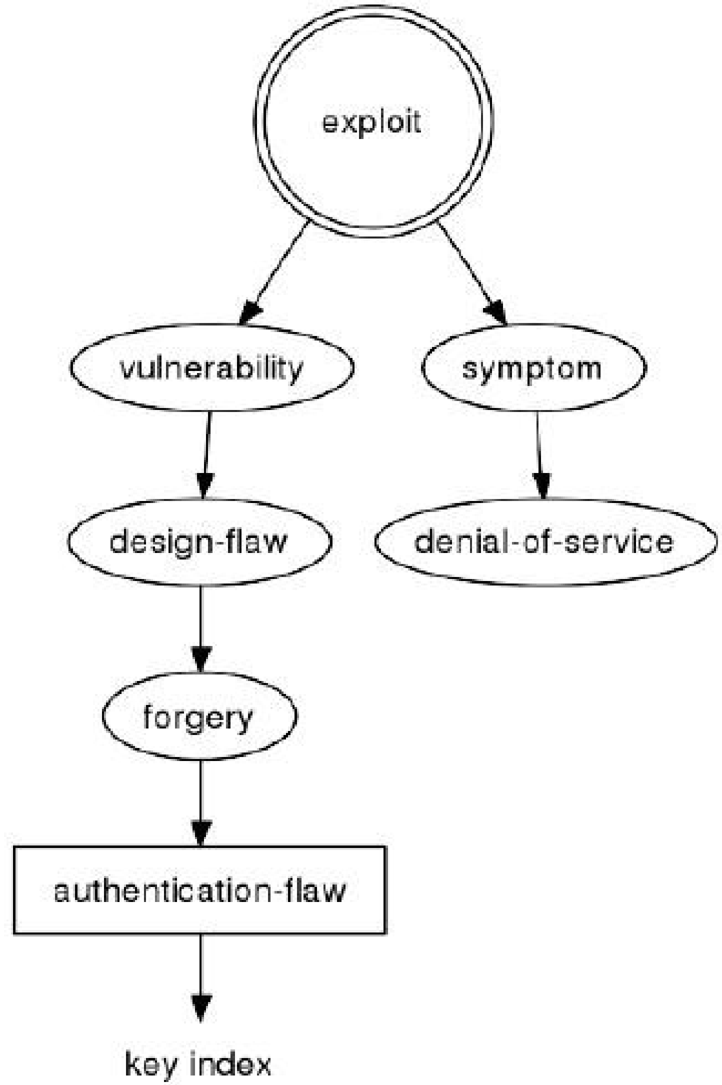
$\quad secure_p \leftarrow secure_p \wedge \neg\textsc{Has-Nonce-Flaw}\,(p)$

$\quad secure_p \leftarrow secure_p \vee \textsc{Allows-Forgery}\,(p)$

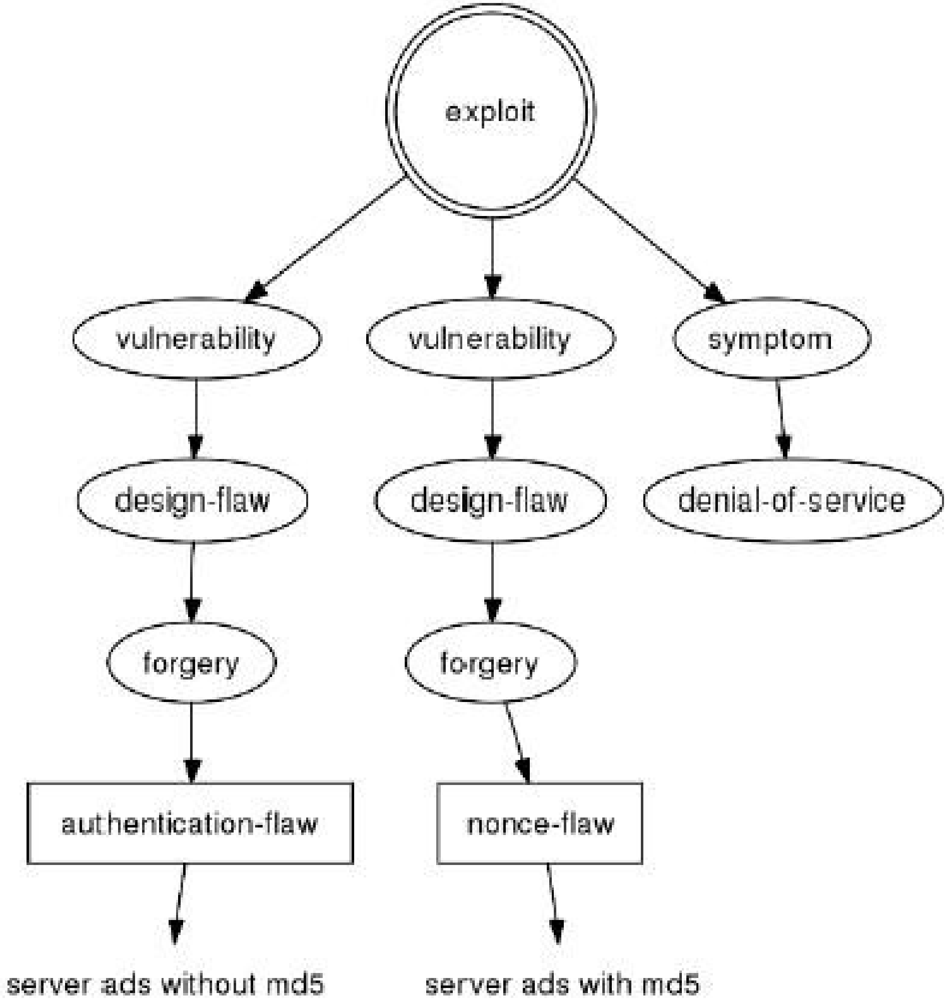
$\quad secure \leftarrow secure \wedge secure_p$

return $secure$

`ntp` Vulnerability



VLAN Vulnerability



So,

- Is this formalism helpful?
- What is required in terms of functions and libraries?
- Use the IPv6 core and common libraries to generate prototype test suites?
- Follow up with a similar approach for layer 4 protocols? Is this feasible?
- Known effort - TTCN 3 and Security – T3FAH

Thank You