Vulnerability Discovery with Attack Injection

IEEE Transactions on Software Engineering (2010)

Joa˜o Antunes, Nuno Neves, Miguel Correia, Paulo Verissimo,and Rui Neves

Park, Ji Hun

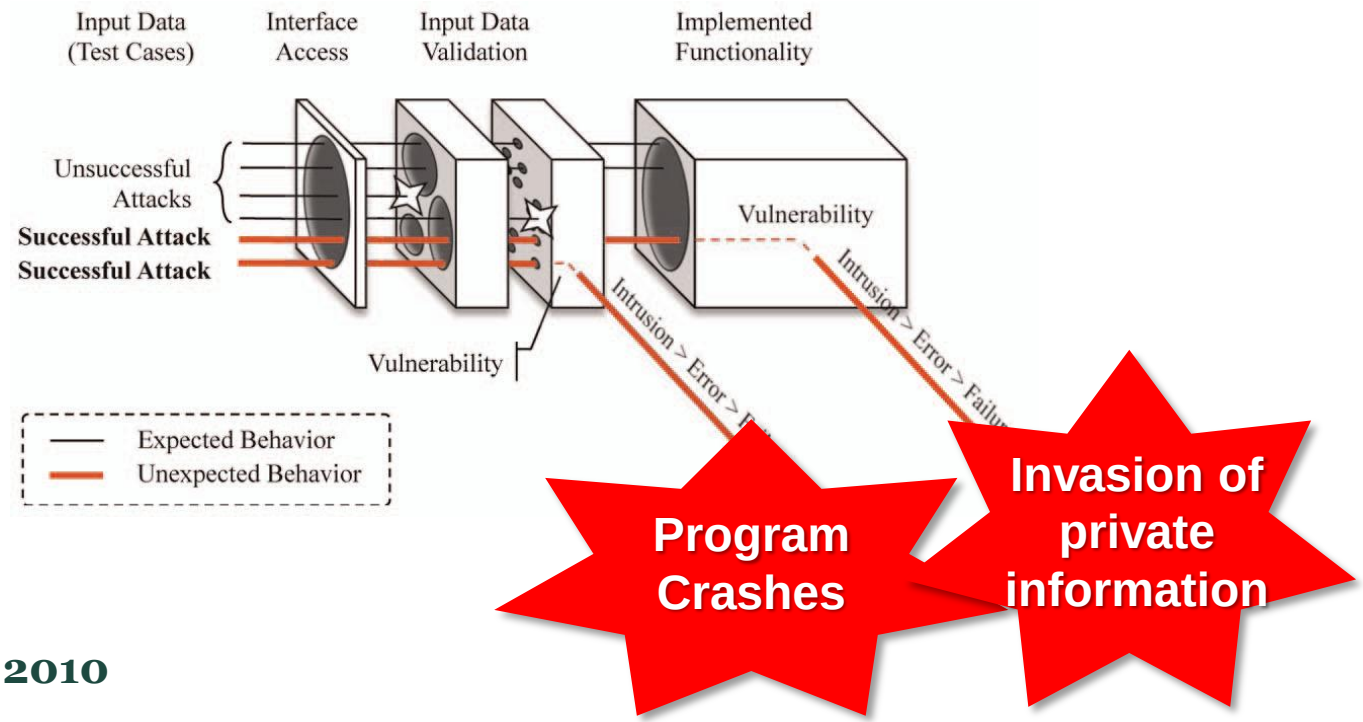
2010.08.17

KAIST SE LAB
KAIST Software Engineering Laboratory

Contents

Introduction (1/2)

- Vulnerability
  - Faults caused by wrong design, implementation mistakes, which is easy to being exploited by an attack
    - Attack means that malicious input performs some unintended and usually illegal activity



Introduction (2/2)

- Motivation
  - Software dependability is getting more importance over the years especially in networked computer systems
  - Network-connected servers should consider about new threats and forms of attacks
    - Sustain long periods of uninterrupted operation
    - Corruption on the server may threaten client's privacy

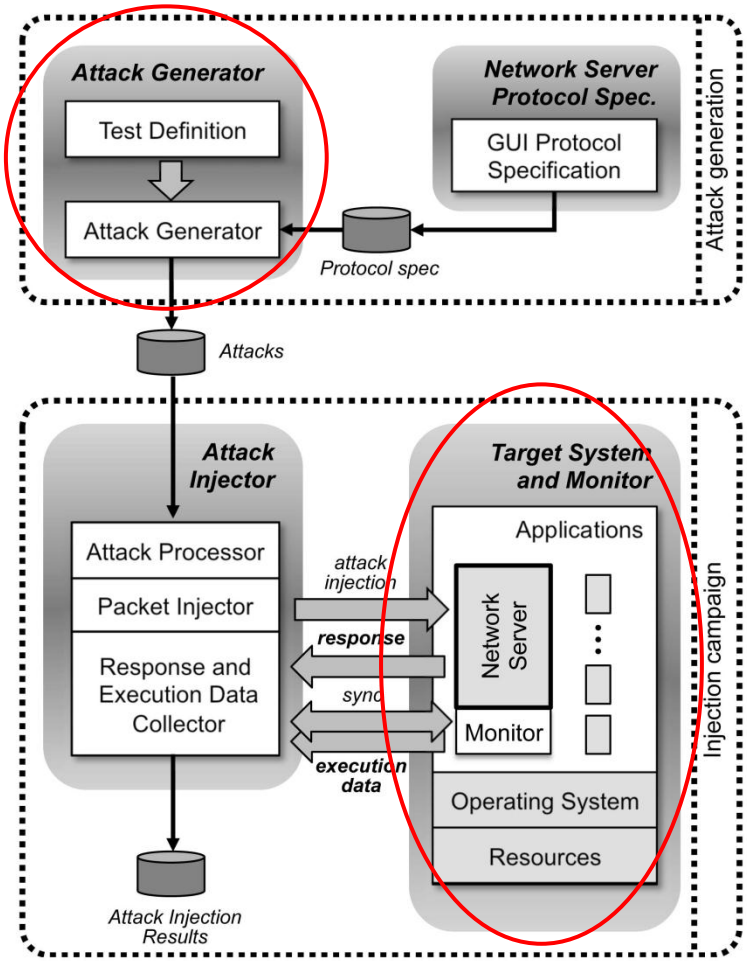
- Goal of this paper
  - Suggest an attack injection tool(AJECT) for the automatic discovery of vulnerabilities in software

Attack injection tool (1/8)

❖ Overview of tool AJECT

Attack generation: Automatically generate attacks in 4 ways

Input : Protocol specification



Injection phase: Execute previously generated test cases(attacks)

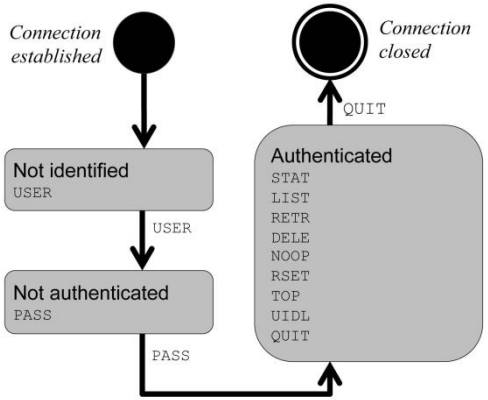
Monitoring: Monitor target system's state while executing attacks in 3 ways

Attack injection tool (2/8)

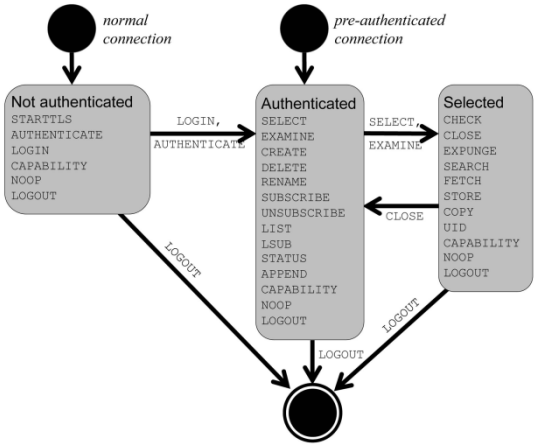
- Protocol as an input
  - Even though the source code of server system is not available, protocols tend to be well defined
  - AJECT offers a graphical user interface tool to input specification of protocol
  - The tool attacks servers by transmitting erroneous packets with the protocol

Attack injection tool (3/8)

- Network server protocols
  - 2 email protocols
    - Fully developed and commonly used
  - POP protocol (POP3)
    - Post Office Protocol
    - Three states
    - Interaction through text strings
  - IMAP protocol (IMAP4Rev1)
    - Internet Message Access Protocol
    - Much more complicated than POP
      - Wider functionality(ex. Create mailbox)
    - Interaction through text strings



State machine of POP protocol



State machine of IMAP protocol

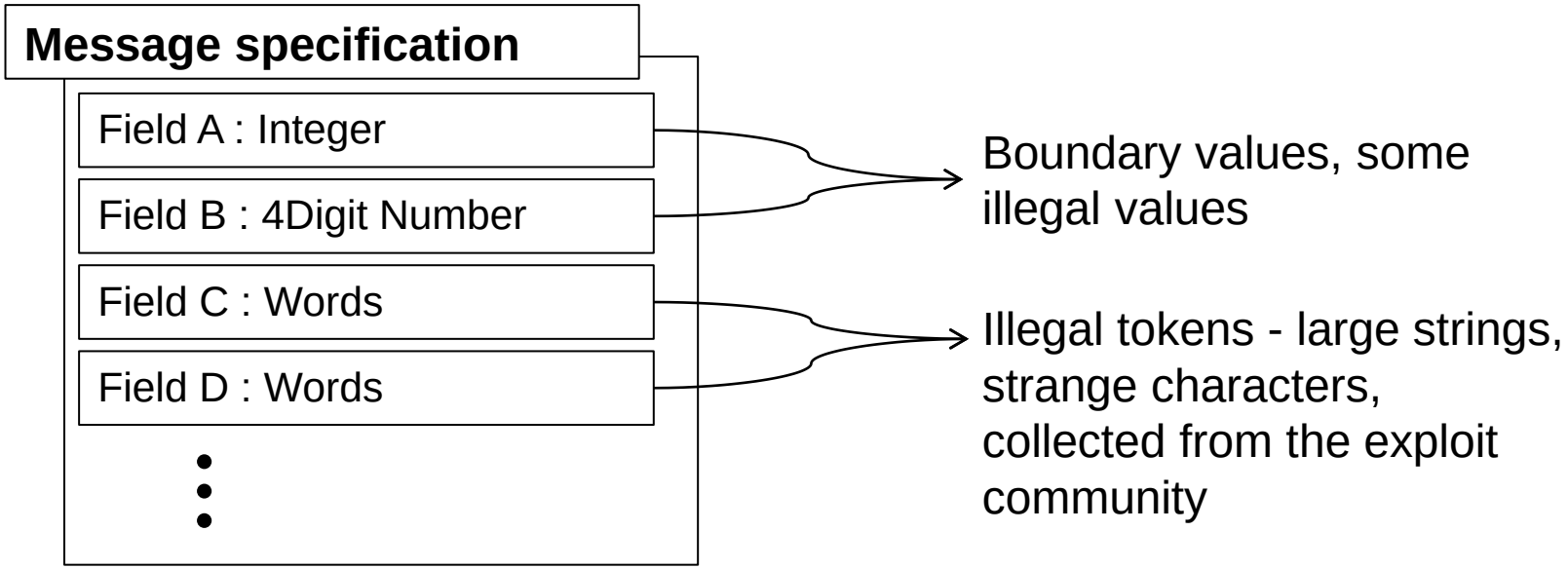
Attack injection tool (4/8)

- Attack generation phase
  - Delimiter test
    - Put illegal or missing delimiters
      - Double quotes as illegal delimiters
      - Omitting delimiter like ‘space’ character

  - Syntax test
    - Violate syntax
      - Addition, elimination, reordering of each field

Attack injection tool (5/8)

- Attack generation phase(cont'd)
  - Value test
    - Replace message specification fields with malicious tokens



Example

- AUTHENTICATE <A x 1296>
- < %s x 10 >;

Attack injection tool (6/8)

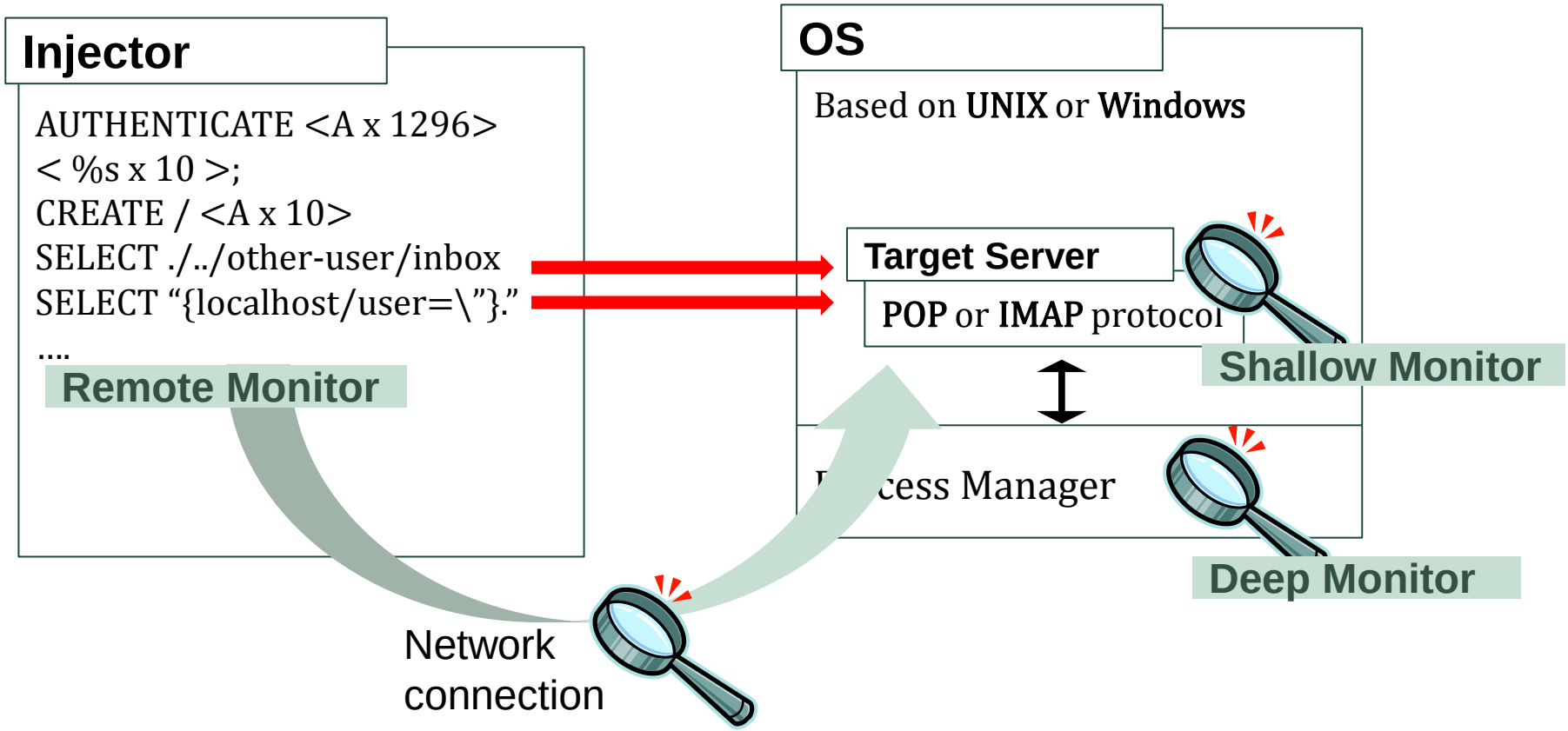
- Attack generation phase(cont'd)
  - Privileged access violation test
    - Determine if the server allows unauthorized accesses
      - Specialization of the value test
    - If server allows these attacks, revealing private information or granting access to some files could be possible

Example

- CREATE / <A x 10>
- SELECT ./../other-user/inbox
- SELECT "{localhost/user=\"}."

Attack injection tool (7/8)

- Injection campaign phase
  - Attack injector and three kind of monitors



Attack injection tool (8/8)

- Injection campaign phase(cont'd)
  - Monitor

Monitor	Property	Execute in	Access Target Server's OS
Deep Monitor	Trace server's main process and children in great detail	Target server machine	O
Shallow Monitor	Monitor server's termination code	Target server machine	O
Remote Monitor	Infer server state by network connection	Injector machine	X

Case study (1/2)

- Experimental results
  - Target network servers

- Total up-to-date 16 servers
- Every servers support POP and IMAP both
- 8 Window servers
- 3 Unix servers
- 5 U/W servers
- 12 commercial servers
- 4 open source servers
- 3 D/S/R monitors
- 13 Remote monitors

E-mail Servers (POP3/IMAP4)	OS	Version	Build Date	Monitor
602LAN Suite *(602 Software)*	W	5.0.08.0403	4/8/2008	R
Citadel*	U	7.32	2/17/2008	R
dovecot*	U	1.1.rc3	3/9/2008	D/S/R
Hexamail Server Corporate	U/W	3.1.0.002	-	R
hMailServer	W	4.4.1	3/9/2008	R
IMail Server *(Ipswitch)*	W	2006,23	12/5/2007	R
Kerio MailServer *(Kerio Tech.)*	U/W	6.5.1	5/12/2008	R
Mailtraq *(Fastraq)*	W	2.12.1.2364	5/8/2008	R
Mdaemon *(Alt-N Technologies)*	W	9.6.5	6/19/2007	R
Merak Mail Server *(IceWarp)*	U/W	9.1.0	9/17/2007	R
NoticeWare Email Server NG	W	4.6.2	4/3/2008	R
Softalk Mail Server Corp.	W	8.5.1.431	10/30/2007	R
SurgeMail Mail Server *(NetWin)*	U/W	3.9e	4/10/2008	R
uw-imap* *(Univ. of Washington)*	U	2007b	6/4/2008	D/S/R
WinGate Email Server *(Qbik)*	W	6.2.2	7/12/2008	R
xmail*	U/W	1.25	1/3/2008	D/S/R

* - Open source; U - Unix/Linux; W - Windows
D - Deep monitor; S - Shallow Monitor; R - Remote Monitor

Case study (2/2)

- Experimental results(cont'd)
  - Results
    - AJECT found vulnerabilities in five servers out of 16 servers
      - AJECT can be very useful in discovering vulnerabilities
    - Characteristic of vulnerable servers
      - All vulnerabilities are found with IMAP protocol
        - » Indicate more complex protocols lead to more error-prone
      - All of vulnerable servers are closed source commercial servers
        - » Larger and active community could make fewer flaws

Vulnerable Server	Attack	Observable Behavior
hMailServer(IMAP)	>20k CREATE and RENAME messages	Server becomes unresponsive until it crashes
NoticeWare(IMAP)	>40 A01 LOGIN Ax5000 password	Server crashes
Softtalk(IMAP)	>3k A01 APPEND messages	Server crashes after low memory
SurgeMail(IMAP)	A01 APPEND Ax5000	Server crashes
WinGate(IMAP)	A01 LIST Ax1000 *	Server deny all connections

Related works

- Fault injection
  - Inject faults to study behavior in the presence of faults
    - Because of simplicity of faults, it is difficult to apply complex faults, like vulnerabilities of network servers
- Fuzzers
  - Inject random sample inputs to discover vulnerability
    - Since it uses random samples, test cases are either too simplistic or specialized
    - Lack of monitoring mechanisms
- Vulnerability scanners
  - Inject faults from database of known vulnerabilities
    - Unable to uncover unknown vulnerabilities

Conclusion

- Contribution
  - Suggest automated methodology to discover vulnerabilities in software
  - Suggest attack injection mechanism which can be applied without target's source code
- Future work
  - Experiment with other protocols

Discussion

- Pros
  - The tool uses only well-defined part(protocols) of the program which could be poorly documented
  - Well-organized experiments and excellent results

- Cons
  - Simple idea, similar with existing test case generation and authors just added automatic monitoring
  - In attack generation phase, it also needs pre-defined or collected malicious tokens

Thank You

Q & A

KAIST SE LAB
KAIST Software Engineering Laboratory

About authors

- Joao Antunes
  - In PhD degree, University of Lisboa in Portugal
- Paulo Verissimo
  - Professor at the University of Lisboa in Protugal
  - Past associate editor IEEE Transaction on Dependable and Secure Computing
  - Past chair of IEEE Technical Committee on Fault-Tolerant Computing
  - Steering Committee of the DSN conference
  - Fellow of the IEEE and the ACM