Detecting and Preventing Attacks with Vulnerability Signatures

David Albrecht, University of Illinois at Urbana-Champaign
ADVISOR: Nikita Borisov

Background
Exploit-generic vulnerability signatures use application and protocol semantics to precisely identify vulnerability vectors, rather than attack. We aim to build a fast implementation of vulnerability signatures to be detected in the network or on the NIC.

Research Plan
• Create vulnerability signatures for bugs.
• Classify vulnerability by complexity.
  • Can a simple regular expression describe all malicious inputs known to trigger the vulnerability?
  • If regular expressions are inadequate, detector may have to perform application-like parsing on input.
• Develop hardware-based detector implementation.
  • Is the Intel IXP platform appropriate?

Fundamental Questions/Challenges
• Computational complexity
  • Higher-level analysis requires data stream reassembly up to OSI Layer 7
  • HTTP supports 3 parameter encodings, requires detector normalization
  • Vulnerability language must be minimal, yet highly expressive
    • Bounded resource usage (e.g., no dynamic memory allocation)
    • Must perform at line rate

Example Bug: Common Unix Printing System Content-Length
• Negative HTTP Content-Length parameter causes denial of service in CUPS
  Content-Length: -1
• Detectable with regular expression:
  /Content-Length:\.*-/i

Example Bug: CUPS Attribute Length
• CUPS uses static buffer to hold variable-length attribute in binary data stream
  unsigned char input[8192];
• Attribute specified as length-prefix string using 16-bit unsigned big-endian format
  23 28 <String data comes next>
• Difficult pattern for regular expressions
  • Any value over 8192 is malicious
  • Data-dependent string offset
• Turing-complete parser can emulate CUPS parser; is this acceptable for line-rate performance?

Related Work/Interaction with Other Projects
• Planned collaboration with Georgia Tech’s Center for Experimental Research in Computer Systems and School of Electrical and Computer Engineering, with Herbert Bos’s research group at the Vrije Universiteit Amsterdam, and with Li Xu at the Universiteit van Amsterdam