WCIS: A Prototype for Detecting Zero-Day Attacks in Web Server Requests

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Presentation Outline

- Web Classifying Immune System (WCIS)
  - Traditional Artificial Immune System (AIS) features
  - Differences from traditional AIS
  - Classification Scheme
  - Web Server Request Model
  - Population Lifecycle
- Experimental Results
  - Accuracy at detect attacks in specific classifications
  - Detection of unknown attacks
- Future Research

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Web Classifying Immune System (WCIS)

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Artificial Immune System (AIS)

- Inspired by biological immune systems
  - Ability to adapt to variants and new pathogens
  - Pattern matching for “antibody” and “antigen” binding
- AIS tries to distinguish “self” from “non-self”
  - “Self” is “normal” traffic, “non-self” is “abnormal” traffic
- Uses several key biological features
  - Negative selection
  - Affinity maturation
  - Immunization
  - Peripheral tolerance

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Web Classifying Immune System (WCIS) Differences from Traditional AIS

- Add classifications to ‘non-self’ patterns
  - Enables specialization of sensors for specific areas
  - Enables “inoculation” for specific attack class(es)
  - Provides more information about zero-day attack than just “an attack has been detected”
- Separate evolutionary process from detection
  - Do costly processes “offline” on back-end system
  - Live traffic detection collects statistics to enable further refinement by back-end system

WCIS – Request Classifications

Class	Description
Info	Gather information about server
Traversal	Read-only directory traversal
SQL	SQL injection attack
Buffer	Buffer overflow attack
Script	Execute a script on the webserver
XSS	Cross-site scripting

WCIS – Request Fingerprint

Characteristics of Request	
HTTP Version	+
HTTP Command	..
Number of Variables	\
Length of URI	(or)
%	< or >
`	//

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

WCIS – Request Parsing

- Pattern/chromosome structure
  - Contains full set of request fingerprint features
  - Flags indicate active/inactive features for sensor
  - Each sensor has at least two active features
    - Example: Length of 50-75 characters and 5-10 + characters
- Pattern matching
  - Sensor compares active features to request
  - Detects request as attack when sensor matches
    - Must fall within range for ranged features
    - Must match set bit for bitmap features
    - Example: Length 65 with 7 + characters

WCIS – Sensor Population Lifecycle

- Random generation of sensors
  - Select features randomly & initialize with random values
- Iterative affinity maturation
  - Perform negative selection
  - Test against attacks in population's classification
  - Breed sensors with best affinity using genetic algorithm
    - Single point crossover and rank selection with elitism
    - Children feature selection based on union of parents' active features and random active features from each parent
  - Mutate subset of new sensors
    - Select random feature and alter it

WCIS – Sensor Population Lifecycle

- Deploy sensors on live environment
  - Currently just test sensors against unlabeled data
  - Record accuracy at detection and false positives
  - Compare classification decisions by sensor populations
- Refine sensors in response to live detection
  - Export statistical information to back-end system
  - Enter a modified affinity maturation loop
  - Code supports concept, but untested due to red tape
- Received clearance to test live deployment and refinement during this academic term

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Experimental Results

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Results – Experimental Setup

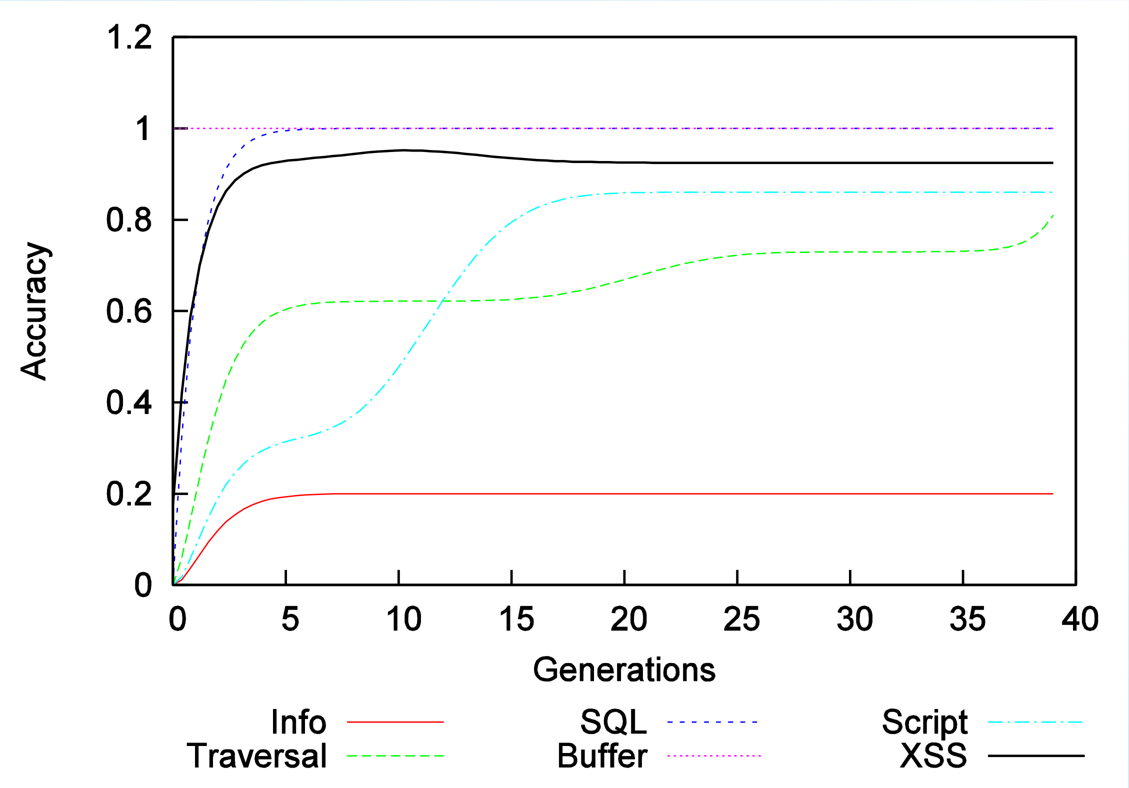
- "Normal" dataset – 52977 requests
  - Web server requests from DARPA Lincoln Labs logs
  - Verified normal requests from live web server logs
- "Attack" dataset – 179 attacks
  - Buqtraq proof of concepts
  - Verified attacks from live web server logs
  - Logs of tests run on isolated machine
- "Unknown" dataset – 11659 requests
  - Random entries from Apache access.log repository for the department web server

Results – Experimental Setup

Variable	Description
Pop	Population size for each classification
Gen	Max iterations for affinity maturation
Xover	Percent selected as parents by GA
Mut	Mutation rate for population
Thresh	Threshold affinity for negative select.
Agree	Attack alert agreement threshold

Results – Classification Accuracy

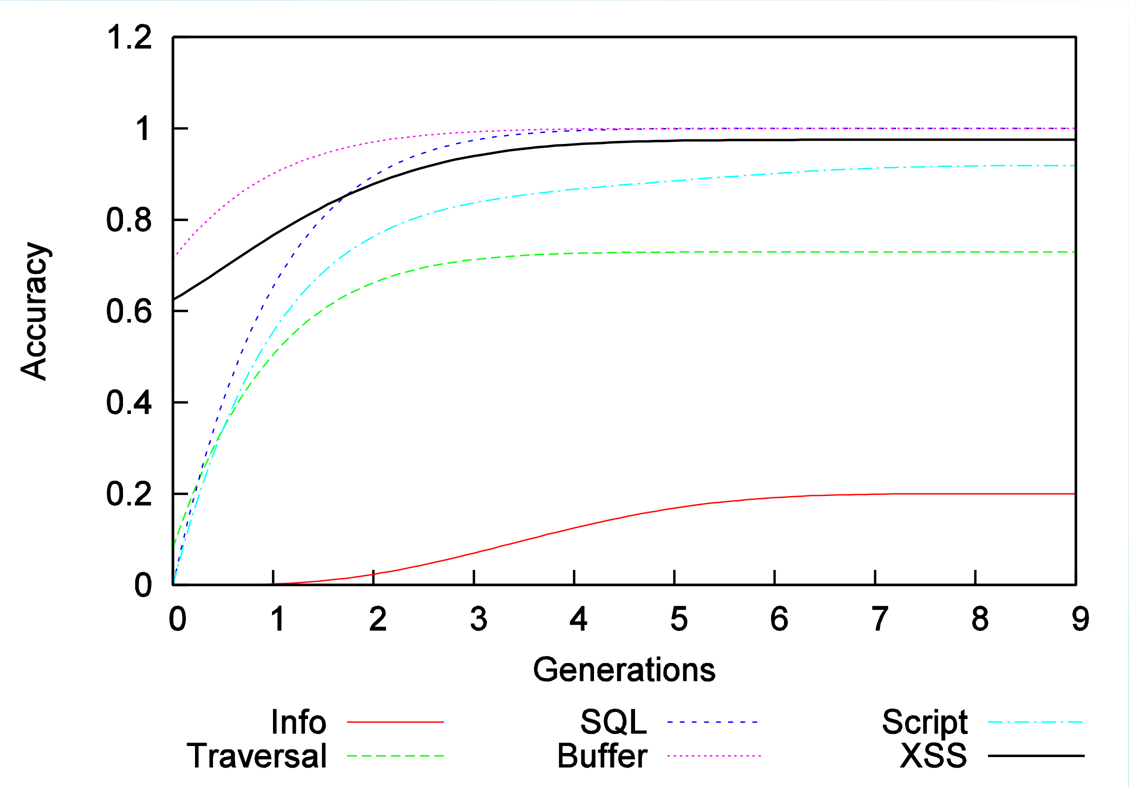
Pop=25 Gen=40 Mut=1%



Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Results – Classification Accuracy

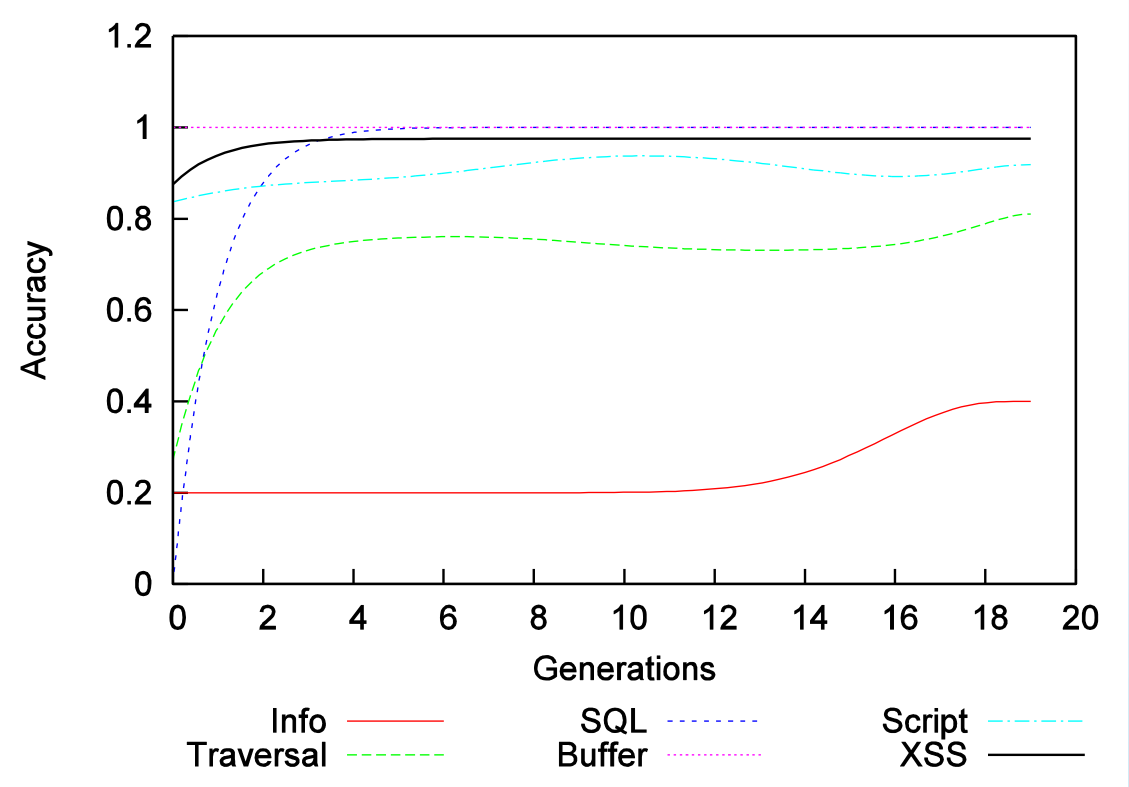
Pop=50 Gen=10 Mut=2.5%



Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Results – Classification Accuracy

Pop=75 Gen=20 Mut=5%



Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Results – Unknown Attacks Detected

Class	URI
Traversal	/.php?index=../../../../proc/self/environ%00
Script	/*.php?option=com_dump&controller=..//..//..//..//..//..///proc/self/environ%0000
Traversal	Same as previous line
Script	/faculty/interests/..\\index.html
Script	/cs150/index.php?p=../../
Script	/…/ports_labeled.jpg

Dr. Melissa Danforth
Dept. of Computer & Electrical Engineering & Computer Science
California State University, Bakersfield

Future Research

Future Research

- Detection against modeled data (real-time)
  - Isolated network is now functional
- Detection against live data – clearance received
- Expand fingerprint to include other parts of request
  - Attack data can be in other fields in request
- Explore other genetic algorithms
  - Single objective algorithm may not be best
  - Try multi-objective algorithms
  - Try variations on genetic algorithms
- Investigate other networking problem domains

Questions?