BAE SYSTEMS

Lessons from 8 Years of Government Experiments in Cyber Warfare Research and Development

Dr. Tiffany M. Frazier
BAE Systems, Arlington, VA

July 2-5, 2009

Approved for Public Release, Distribution Unlimited

Applied Research Cyber Experimentation

BAE SYSTEMS

- **Purpose**
  - Determine promise of current research direction
  - Inform determination of future direction of Government-funded research
  - Select and reject technologies for continued development and eventual transition to operational use
  - Convince operational Government partners to fund technology transfer
- **Features**
  - Multi-party: red (attack), blue (defend), white (test), & Government teams
  - Technology “bake-off”
    - Performers are either red or blue if research is offensive or defensive
  - Tests take place on a 3rd party test range
  - Test period tends to be one to six weeks in duration
    - There is always some kind of pre-test “shake out” period
  - Preparing for these tests involves an enormous amount of in-house experimentation with its own set of challenges
- **Experiences in support of this briefing**
  - 16 official tests as a performer (research team)

The Technologies

- **Computer Network Offense Technology**
  - *Ability to attack platforms without being detected*
  - I.e. despite presence of defensive technologies
- **Computer Network Defense Technology**
  - *Ability to stop (or limit impact of) unknown attacks*
  - Key component technologies:
    - *Detection*
    - *Estimation*
    - *Decision*
    - *Response*
    - *Recovery*
- **General nature of the applied research**
  - Leverage promising academic research
  - Conduct our own original research (usually very applied)
  - Build a prototype system to realize some integrated capability
  - Conduct extensive experimentation and analysis
  - Participate in 3rd party validation experiments or tests

BAE SYSTEMS

Experiments in a Nutshell

- **Hypothesis Set (per performer)**
  - Technology meets metric 1
  - Technology does not meet metric 1
  - Technology meets metric 2
  - Technology does not meet metric 2
  - …

PEL Model

- **If the technology meets all of the metrics then it is selectable and the overall hypothesis is true:**
  - A technology can be built to achieve certain new functionality
  - That technology can meet metrics 1 through N

- **Critical Assumptions**
  - These results are externally valid – i.e. they predictive of *operational performance*
  - The metrics measure whether the desired functionality has been successfully built

Example Metrics: DARPA's Dynamic Quarantine Program

BAE SYSTEMS

DYNAMIC QUARANTINE OF COMPUTER-BASED ATTACKS AGAINST MILITARY ENTERPRISE NETWORKS

Phase I Program Go/No-Go Milestones	**Passing Criteria**
Containment	Worms released on testbed must be contained to 10% of vulnerable machines by dynamic quarantine defenses.
False positive rate	False positive rate of detector components are not exceed 10 false alarms/day.
Time to recovery	The time to recovery for infected systems shall not exceed 60 minutes.

Phase II Program Go/No-Go Milestones	**End of Program Metric Goals**
Containment	Worms released on testbed must be contained to 1% of vulnerable machines by dynamic quarantine defenses.
False positive rate	False positive rate of detector components are not exceed 1 false alarm/day.
Time to recovery	The time to recovery for infected systems shall not exceed 6 minutes.

Defense Advanced Research Projects Agency

https://www.fbo.gov/index?s=opportunity&mode=form&tab=core&id=cd418979156bb07f104065613b5ade6c&_cview=1&cck=1&au=&ck=

Example Metrics: DARPA's DCAMANETS Program

BAE SYSTEMS

Defense Against Cyber Attacks on Mobile, Ad Hoc Network Systems (MANETS)

Phase I Program Go/No-Go Milestones	Passing Criteria
Containment	MANET-based system must be able to detect and self-reconfigure such that it contains worms released on MANET to 10% of all vulnerable nodes.
False reconfiguration	System does not reconfigure on more than 10% of normal sessions.
System throughput degradation	Good system throughput does not degrade more than 75% on average over the duration of the attack between any source-destination pairs.
Network Overhead	Network overhead generated by distributed detection mechanisms should not exceed 10% of baseline system throughput during normal conditions.

https://www.fbo.gov/index?s=opportunity&mode=form&tab=core&id=a272938c1bc54afde7c09d8ec76a0fb3&_cview=0

How does this tend to unfold? (For defensive technology R&D)

BAE SYSTEMS

- **Performer research, development, experimentation & analysis (blue)**
- **Metric development and refinement**
  - May be cooperative or less-than-cooperative
  - With multiple performers there is tension to make the metrics favor one party over another
- **Measurement infrastructure (white) and test attacks (red)**
  - Performer has to duplicate all of this in their lab (!) to prepare for testing
  - White and red teams also have a disadvantage in not being able to test their products against the technology prior to trial runs or even the test
- **Separate blue and red shake-out periods**
  - Unearth bugs in the infrastructure, performer technology, and attacks
- **Trial run**
  - For particularly complex tests → may use a single baseline test attack to shake out the experiment process and further bugs in the various systems
  - Frequently there are also tests to make sure the blue technology does not break the attacks simply by being present and running (but not effecting)
- **Test trials**
  - Cooperative: red, blue, & white teams run their systems and conduct analysis
  - Double blind: blue does not have access to red data, which, in practice, means red will have no meaningful access to blue data

My Rough Assessment

- **Experiment Design**
  - Mixed results
  - Best when all parties cooperate with as much disclosure as possible
- **Internal Validity**
  - Generally good *at the appropriate level of fidelity*
  - Be careful about drawing conclusions at the wrong level of fidelity
- **External Validity**
  - The most attention is placed here (still never enough)
  - One issue – perception of validity not always the same as reality of validity
- **Repeatability**
  - This is the first thing the teams get right
- **Reproducibility**
  - Complexity of experiments & technology → very hard for 3rd party to reproduce
  - An interesting and well-explored issue, though, is prepping for 3rd party tests – I.e. will my results be reproducible in someone else's target environment?
- **Analysis and Reporting**
  - 3rd party – generally very poor unless all parties cooperate
  - Internal – extensive internal analysis has been <u>the</u> driver of research progress

Internal Validity

- **Complex interacting systems**
  - Test measurement infrastructure and the test range
  - Traffic generation and host/user activity emulation
  - Movement scenario (for MANETs)
  - The attacks
  - The defensive technology
- **Alternative explanations for the outcome?**
- **If the technology meets the metrics…**
  - Were the tests “too easy”?
  - Did the performers have too much knowledge?
  - Was the target environment realistic enough?
- **If the test fails…**
  - Did the technology stop the attack or did the attack simply fail?
  - Are we even able to determine why the test failed?
- **Gaming the test**
  - Negotiating metrics to make them easier to pass (rare)
  - Outright cheating (really rare)

External Validity Challenges
Defensive Technology – Test Attacks

BAE SYSTEMS

- **Unknown attacks**
  - There are many challenges in "emulating" unknown attacks
    - *It is expensive* to develop and test attacks
    - The "good stuff" is just not going to be used
    - At least partly-shared code base (between attacks) is likely
    - Covering the attack space is infeasible
  - Pretending known attacks are unknown via Rules of Engagement and an Honor Code
  - Even then, any results involving repeated attacks (at some later date) are viewed with suspicion
- **Results of one experiment were completely dismissed**
  - Two different performers were able to defend against all test attacks
  - The test attacks were blamed (too easy and too narrow)
- **The next experiment (same performers)**
  - Good distribution of attacks
  - Internal validity / experiment control was poor (more later)

External Validity -- Other Realism Issues

BAE SYSTEMS

- **Representative populations**
  - Variability in platforms
    - Hardware, operating systems, applications
  - Variability in configurations
    - (Can't have just one systems administrator)
  - "Impossible" variability
    - Network infrastructure such as domain controllers
- **Platforms must be real**
  - Emulation *just does not work* at the pointy end of the spear
  - Fundamentally, attacks (and therefore defenses) are working around and not at interfaces, are exploiting bugs, etc.
  - Farther away from the pointy end emulation is okay
    - E.g. Emulating "the Internet Cloud"
  - This realism poses issues for conducting large scale experiments
- **Criticality of Background noise**
  - I.e. it is easy to defend if the only thing moving is the attack

External Validity -- Other Realism Issues

BAE SYSTEMS

- **Traffic generation and host/user activity emulation**
  - Again due to the need for realism, the only way to go is to script real applications to generate real traffic
- **MGEN (Multi-Generator)**
  - Open source software that provides the ability to perform IP network performance tests and measurements using UDP/IP traffic
  - Developed by the Naval Research Lab
  - MGEN emulates packet loss rates, communication delays and more
  - Essential for testing Mobile Ad Hoc Network-based technologies
- **LARIAT (Lincoln Adaptable Real-time Information Assurance Testbed)**
  - Comprehensive Enterprise network traffic and host/user activity system
  - Developed by MIT/Lincoln Labs
  - Not publicly available

External Validity --
Other Realism Issues

- **Traffic generation and host/user activity emulation, cont**
- **MGEN challenges for Network Defense experimentation**
  - Network flows can have realistic content but
  - Applications were simple loops
    - Trivialized host detection technologies
    - → Results were viewed with suspicion as a result
- **Subsequent experiments**
  - MGEN still used for network flows and radio emulation
  - *Extensive* effort put into scripting realistic video, voice, logistics and other applications

Repeatability – Some hard challenges well met

BAE SYSTEMS

- **Mobile Ad Hoc Networks involve special challenges**
  - A run is driven by a movement scenario for the “mobile” hosts
  - MGEN traffic generation and radio emulation
  - Real (heavily scripted) applications
  - Control Infrastructure
  - The attacks
  - And the defensive technology
- **An impressive amount of repeatability in this complex environment**
  - Remote repeatable control (scenario-applications-attacks)
  - Were able to runs dozens of trials
  - Up to 500 real hosts
- **Tension between realism and performance analysis**
  - Gap existed between a realistic movement scenario and ability to explore the performance envelopes of the defensive technology
  - Difficult to decide which corner cases are worth exploring

Full Disclosure – The Good

- **Best-value experimentation experience was when all parties worked closely together (red, blue, and white)**
  - Defensive technology test
  - Control infrastructure (known), attacks (unknown), movement scenarios (some unknown), target environment (known)
  - All parties get their software debugged and working
  - Critical in the MANET environment, for example, which has an extra level of complexity due to the use of movement models and the need to synchronize application execution
    - E.g. Packet loss can lead to dramatically different performance from one run to the next
- **Test runs**
  - Results were available to everyone to analyze
  - Once the runs began – all data became "known" in real-time
  - Some analysis could be performed in real-time as the runs unfolded
- **Got to test many aspects of the system and corner cases**
  - Depth of sensor suite
  - Distributed coordination algorithms

Full Disclosure – The Ugly (1 of 2)

- **Worst-value experimentation experience was double blind**
  - Defensive technology test
  - Control infrastructure (known), attacks (unknown), target environment (known)
  - Blue technology reported to Red/White data regarding any actions taken against detected attacks
  - Red team ran attacks, White team ran the infrastructure, and the Blue team ran the technology
  - No sharing of data…
    - The blue team didn't know if and when attacks were being run
    - The red team had no access to blue team GUI to understand what, if anything, the technology was doing in real-time
  - Other than the real-time Blue GUI
    - Blue team could collect any blue data desired, but only a day or more AFTER the run completed
  - Blue was able to get very limited "ground truth" from White/Red (a day later) – e.g. which boxes were successfully attacked and the launch point

Full Disclosure – The Ugly (2 of 2)

- **And chaos ensued…**
  - White/Red team did not know if their attacks did not work or if Blue had successfully stopped them
  - Blue could only verify if the system had taken any action or not
    - More analysis required access to Blue logs (which were delayed)
  - This actually led to the Test Director asking us to change our system configuration
    - Which we did … "blind" … based on verbal data from White/Red
  - And then the experiment schedule was not sufficiently altered to handle the two configurations
    - Blue Config 2 saw attacks Blue Config 1 had not and vice versa
  - All results were viewed with suspicion
- **At the post test runs hot wash**
  - Results were mostly empty – Red/White teams could not tell what had happened
  - Fortunately we could reverse engineer what really happened from our Blue data logs (back in our lab) with the limited "truth" data from White/Red

Reproducibility -- From Performer Test Range to 3rd Party Test Range

BAE SYSTEMS

- **Test range**
  - Hardware differences
  - Network infrastructure configurations – e.g. domain controllers
  - "Surprise" software – such as Microsoft's service load balancing
- **Test measurement and Experiment Control**
  - Always try to utilize white team's experiment control (though our own usually allows for much more efficient experimentation)
  - In one case we wrote line-for-line equivalent metric measurement and analysis code. This was essential for debugging the white team's code.
- **Target Machines**
  - System administrator differences
  - In the extreme a "recipe" and "gold disk" are used to build identical platforms
  - If the targets are supposed to be at least partly "unknown" then planning for last minute integration issues is necessary
- **Applications**
  - In some cases we never got these working in our lab
- **"Latest version" issues**

Experiment Design - Metrics

- **Secondary metrics**
  - At the wrong level of fidelity

Successes: Anomaly Detection Research

BAE SYSTEMS

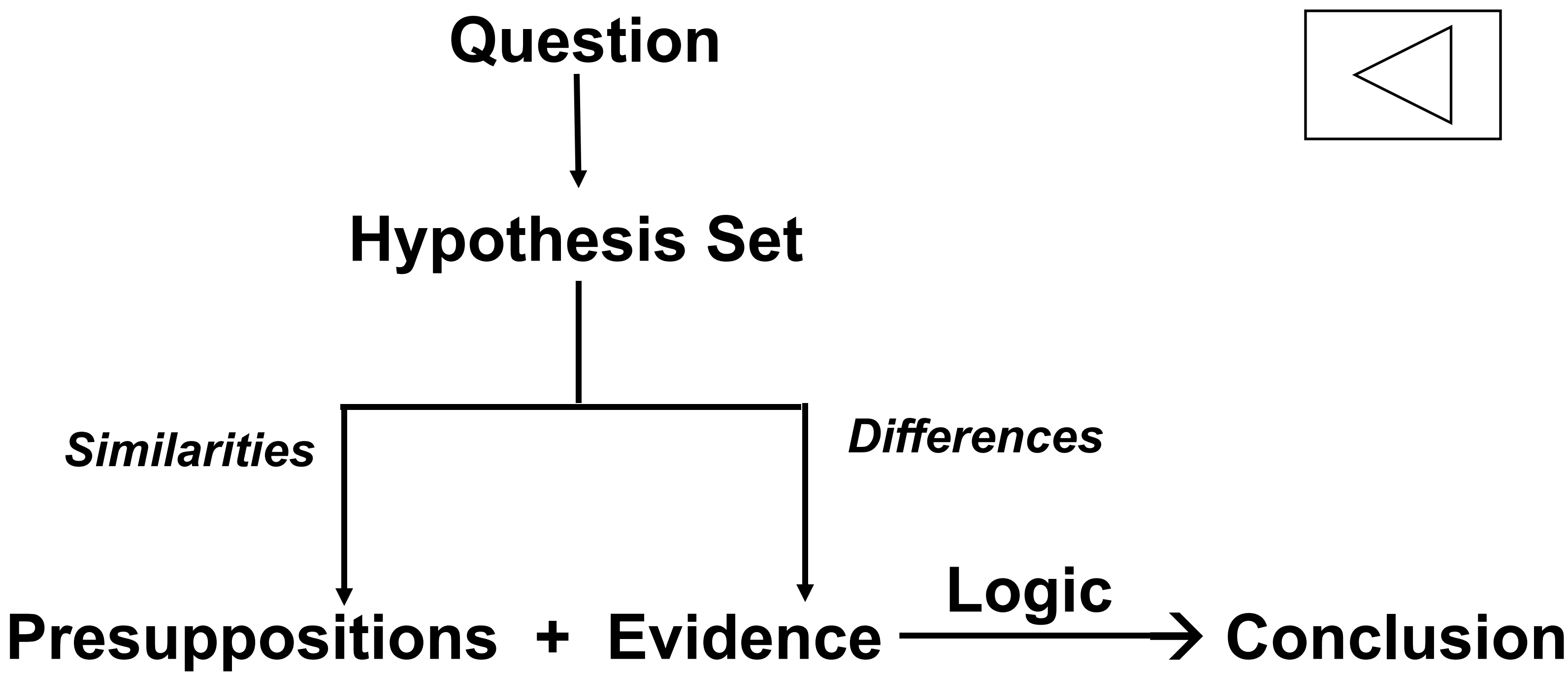
- **Started out with academic anomaly detection research**
  - Host- and network-based anomaly detection research
  - Train on normal (host or network) activity, detect attacks as non-normal anomalies
- **Collected and analyzed an enormous volume of results**
- **Key research areas**
  - When anomaly detectors fail they can fail spectacularly
  - In situ training
  - Model aggregation as a way to deal with differences between different network flows and hosts
  - Incremental updates to models as normal changes or new normal activity appears
  - Rate-based detectors do not work – models end up including all possible rates or they end up too narrow & ("boom")
  - Feature analysis – which features can be successfully abstracted across different hosts and which can not
  - "Big" models do not work; lots and lots of small, well-trained models work well
  - Breadth of anomaly detector suites
  - Scoring functions

Concluding Thoughts

BAE SYSTEMS

- **How experiments are conducted is incredibly important**
  - Methods used in the academic work that we leverage are lacking
  - Methods used in our applied research experimentation are “fragile”
    - Can easily go astray → wasted $ and frustrated scientists
    - What can we do to make this less likely to happen?
- **My top two wishes for academic research**
  - A methods section in every paper
  - That there was some “3rd party independence” in the experimentation
- **Internal Validity**
  - Must be careful to draw conclusions at *the appropriate level of fidelity*
- **Analysis and Reporting**
  - Cooperative analysis and full disclosure is powerful and essential
- **Experimentation areas that could use research**
  - Traffic and host/user activity generation
  - Testing against “the unknown”
  - External Validity: Need for realism versus need for confidence that the results are representative (statistically)
- **My worst fear**
  - Our “double blind” nightmare could easily happen again

PEL Model (Gauch, Jr) for Scientific Inquiry



[Archive]