Automated Attack Framework for Test & Evaluation (AAFT)

2019 International Test and Evaluation Association Cybersecurity Workshop

March 26-29, 2019

Mr. Andrew Shaffer

The Applied Research Laboratory
The Pennsylvania State University

Distribution Statement A. Approved for public release; distribution is unlimited.
Outline

- Test & Evaluation Needs
- AAFT Overview
- Development Approach
- New Technologies
- System Tools
- Potential Use Cases
- Development Status
- Summary & Future Work
Test & Evaluation Needs

• DoD 8500.01 requires Red Team cyber penetration testing
  - Must test IP & non-IP systems (DOT&E Cybersecurity Memos of 7/27/16 & 4/3/18)
  - Continuous and comprehensive monitoring recommended

• Red Teams lack availability for all required system testing
  - Limited availability delays system accreditation and increases risk
  - Tools to improve tester utilization and efficiency are needed

• Red Team training is time-consuming and highly specialized
  - Knowledge of one attack tool often does not translate to others
  - Decision aids are needed to improve and accelerate tester training

• Defender training and vulnerability validation require better support
  - Automated tools are needed to accelerate tasks and free up testing personnel

Automated tools are needed to improve tester utilization & efficiency
AAFT provides an integrated framework to automate cyber attacks

**AAFT Overview**

Currently, the few available skilled testers must:
- Emulate all threats
- Possess extensive IP and non-IP system knowledge
- Emulate adversaries for defender training
- Validate vulnerabilities

**Penetration Testing Techniques**

- **Basic**
- **Intermediate**
- **Advanced**

With AAFT, skilled testers can focus on advanced threats:
- AAFT emulates basic/intermediate threats
- AAFT facilitates interactions with IP and non-IP systems
- AAFT emulates adversaries
- AAFT validates vulnerabilities

**Provides automated basic and intermediate-level penetration testing**
- Enables comprehensive continuous monitoring of IP and non-IP systems
- Provides decision aid for attack selection
- Facilitates defender training and improves entry-level tester effectiveness
- Generates comprehensive attack logs to support post-attack forensic analysis

**Provides easy-to-use framework that facilitates new attack integration**
- Supports easy integration with new open-source and custom tools
- Similar to Metasploit but with support for multiple network types and attack tools

**Allows skilled testers to focus on advanced threats**
- Takes over less challenging penetration testing tasks
- Accelerates vulnerability validation
Development Approach

- Develop database of real-world attacks
  - Specify attack preconditions, postconditions, timing, risk of detection, and risk of damaging target
- Create modular attack scripts
  - Programmatically invoke attacks from command line
  - Capture, parse and store attack output
- Develop attack tree generation algorithms
  - Generate attack trees offline using threat representative attack scripts that meet timing and risk tolerances
  - Algorithms automatically reconfigure attack tree whenever new attack scripts become available
- Develop Autonomous Attack Engine & GUI
  - Create algorithms to optimize attack selection based on attack objective, timing, and risk
  - Support large attack tree visualization & navigation
- Develop non-IP network connectors
  - Develop hardware and software to automate connections to non-IP network interfaces

Incremental development ensures ongoing capability improvement

Distribution Statement A. Approved for public release; distribution is unlimited.
Attack scripts encode simple attack commands and parameters
- Specify preconditions (i.e. knowledge required) for attack to proceed
- Specify estimates of time required, risk of detection, and risk of damaging target
- Specify expected postconditions (i.e. knowledge gained) if attack is successful

Scripts capture attack tool output following attack execution
- Parse tool output and update list of conditions that have been fulfilled

Simple well-defined interfaces facilitate community creation of new attack scripts
- Community involvement supports broad coverage of current and new attack tools

Distribution Statement A. Approved for public release; distribution is unlimited.
• AAF attack trees encode attack sequences and target knowledge
  - Overall attack initially starts with minimal target knowledge
  - Each attack script attempted gains additional target knowledge
  - Attack tree nodes encode specific attack scripts and knowledge about target at specific points during attack

• Attack trees are generated offline using all available threat representative attack scripts that meet timing and risk tolerances
  - Attacks are added to the tree wherever their preconditions are fulfilled and the attack has the potential to gain additional target knowledge or access
  - Attacks are added to tree recursively until no further additions are possible
  - Branches not reaching the objective and non-optimal branches are pruned

Attack Trees are generated offline to emulate specific threat profiles
Autonomous Attack Engine provides high-level direction for AAFT attack

- **Autonomous Attack Engine** iteratively selects and executes optimal attack, updates target knowledge, and updates attack tree until attaining objective or exhausting all available attacks
  - Optimally selects next attack based on currently available information
  - Continually updates alternative attack options as target knowledge increases
  - Changes attack paths whenever a better attack option becomes available
- Can also recommend attacks as a decision aid instead of autonomously executing them
- Records complete transcript of attacks for forensic analysis

**Autonomous Attack Engine Processing Cycle**

Start: Full threat-representative attack tree w/minimal target knowledge

Finish: Attack transcript showing how objective achieved or determined unreachable

Select & Execute (or Recommend) Next Attack

Update Attack Tree to Reflect Latest Target Knowledge

Update Target Knowledge
**New Technologies**

**Non-IP Network Connectors**

- Non-IP network connectors automate connections to a wide variety of network types to dramatically accelerate non-IP system testing
  - MIL-STD-1553, ARINC-429, CAN Bus, Profibus, Modbus, NTDS, etc…

- Connectors streamline connection process and reduce test setup time
  - Provide AAFT with physical link to non-IP networks
  - Automatically detect and decode network protocols and parameters

Connectors enable AAFT automated testing of non-IP systems

Distribution Statement A. Approved for public release; distribution is unlimited.
Initial AAFT development is focused on common open-source tools:
- Nmap (network mapper)
- Metasploit (integrated attack framework)
- THC Hydra (online login cracker)
- John the Ripper (offline password cracker)

Attack scripts and target knowledge encoded using Extensible Markup Language (XML):
- XML is portable and highly extensible
- Open-source and custom attack tools use the same interfaces

Python facilitates rapid software development:
- Highly portable high-productivity language that is widely used

VirtualBox supports software portability and easy installation:
- AAFT system will ship as a ready-to-run virtual machine image
- Simplifies end user configuration and maintenance requirements

AAFT framework supports both open-source and custom tools.

Distribution Statement A. Approved for public release; distribution is unlimited.
Allow skilled Red Team testers to focus on advanced attacks
Perform basic and intermediate-level Red Team attacks using an innovative Autonomous Attack Engine
Accelerate vulnerability validation and remediation for test & evaluation
Support Network Defender training

AAFT supports Red Team and Network Defender cybersecurity efforts

Distribution Statement A. Approved for public release; distribution is unlimited.
Development Status

- Develop database of real-world attacks
  ✓ Internal and external data structures defined

- Create modular attack scripts
  ✓ Basic attack script structure developed
  ✓ Initial scripts populated (compromise 4 test systems)
    - Additional scripts currently in development

- Develop attack tree generation algorithms
  ✓ Attack tree generation and pruning algorithms defined
  ✓ Initial regression testing data sets developed
  ✓ Algorithms have already demonstrated attack synthesis
    - Scalability optimizations currently being applied

- Develop Autonomous Attack Engine & GUI
  ✓ Shortest-path selection algorithm implemented
    - Additional selection algorithms, GUI, and tree visualization algorithms currently in development

- Develop non-IP network connectors
  - Prioritization of non-IP networks currently in progress

✓ = Completed
Blue = In Progress

AAFT system development is well underway

Distribution Statement A. Approved for public release; distribution is unlimited.
AAFT advances state-of-the-art in Red Team cybersecurity testing

• Automatic attack tree generation and Autonomous Attack Engine are new technologies that advance cybersecurity test and evaluation
  - Applying autonomy research to cybersecurity improves state-of-the-art in cybersecurity penetration testing

• AAFT addresses critical cybersecurity Test and Evaluation needs
  - Enables better utilization and efficiency of available Red Team testers
  - Increases scope of cybersecurity testing that can be performed
  - Supports efficient cybersecurity testing of IP and non-IP systems and networks

• System design and development is ongoing as ARL internal research
  - Created attack tool test lab and system prototype development lab
  - Developed initial attack scripts and algorithms for attack tree generation
  - Began initial development of Automated Attack Engine

AAFT advances state-of-the-art in Red Team cybersecurity testing
Future Work

• Incrementally increase tool and attack sophistication
  - Add support for concurrent, persistent, and social engineering attacks
  - Survey MITRE Adversarial Tactics, Techniques, & Common Knowledge (ATT&CK) Matrix and add attacks to ensure that AAFT supports full spectrum of threats

• Add support for automated analysis of harvested target information
  - Apply automated text analysis and natural language processing to identify key IP addresses, port numbers, and user information that may enable greater penetration

• Improve attack tree visualization and GUI
  - Leverage gaming techniques for attack navigation and full attack tree visualization

• Improve support for non-IP network testing
  - Integrate hardware and automated network detection/protocol decoding algorithms

Seeking transition sponsors & funding to support further maturation

Distribution Statement A. Approved for public release; distribution is unlimited.
Points of Contact

Andrew Shaffer
Penn State Applied Research Laboratory
814-863-0312
aps148@arl.psu.edu

Bruce Einfalt
Penn State Applied Research Laboratory
814-863-4142
bte2@arl.psu.edu
Backups

• AAFT Quad Chart
• Attack Synthesis Illustration
Expanded cyber penetration testing is urgently needed to detect emergent system vulnerabilities – Continuous comprehensive cyber penetration testing needed for IP & non-IP systems to improve security and reduce cost.

Red Team shortage delays tests & increases risk – Tools needed to improve Red Team utilization and efficiency.

Red Team tester training is time-intensive – Current tools are highly specialized and not well integrated – Decision aids and integrated tools accepting high-level attack directives are needed to reduce tester training time.

Vulnerability validation & tester training shortfalls – Tools needed for training and rapid vulnerability validation.

Contact preconditions, postconditions, timing requirements, risk of detection, and risk of damaging target.

Create modular attack scripts – Programmatically invoke attacks from command line – Capture, parse and store attack output.

Develop attack tree generation algorithms – Automatically generate threat representative attack trees – Support automatic attack tree reconfiguration for new attacks.

Develop novel Autonomous Attack Engine & GUI – Create new algorithms to optimize attack selections – Develop techniques to visualize extremely large attack trees.

Develop non-IP network connectors – Create hardware and software to streamline non-IP tests.

Key Features

- Develop repository of real-world attacks
- Specify attack preconditions, postconditions, timing requirements, risk of detection, and risk of damaging target
- Create modular attack scripts
- Programmatically invoke attacks from command line
- Capture, parse and store attack output
- Develop attack tree generation algorithms
- Automatically generate threat representative attack trees
- Support automatic attack tree reconfiguration for new attacks
- Develop novel Autonomous Attack Engine & GUI
- Create new algorithms to optimize attack selections
- Develop techniques to visualize extremely large attack trees
- Develop non-IP network connectors
- Create hardware and software to streamline non-IP tests

Use Cases

- Skilled Red Team focuses on advanced attacks
- AAAF facilitates and automates basic and intermediate attacks
- AAAF-enabled training supports entry-level Red Team tester skill development
- Attacks blocked by accelerated remediation
- AAAF-enabled training improves accelerated Network Defender efficiency
- New vulnerabilities discovered and remediated

AAAF automates cybersecurity Red Team tasks

T&E Gap

- Expanded cyber penetration testing is urgently needed to detect emergent system vulnerabilities
- Continuous comprehensive cyber penetration testing needed for IP & non-IP systems to improve security and reduce cost
- Red Team shortage delays tests & increases risk – Tools needed to improve Red Team utilization and efficiency
- Red Team tester training is time-intensive – Current tools are highly specialized and not well integrated – Decision aids and integrated tools accepting high-level attack directives are needed to reduce tester training time
- Vulnerability validation & tester training shortfalls – Tools needed for training and rapid vulnerability validation

New Science and Technology

- Attack scripts, attack tree generation algorithm & Autonomous Attack Engine advance cybersecurity – Identify & exhaustively execute threat representative attacks – Support continuous comprehensive monitoring – Increase testing rigor & reduce reauthorization costs
- Automation & decision aids improve tests & training – Allow skilled Red Teams to focus on most challenging tasks – Record complete attack transcripts to support forensic analysis – Improve Network Defender vulnerability validation & training – Increase capabilities of entry-level cyber testing personnel
- Framework enables easy future tool integration – Supports open-source, custom, and proprietary attack tools – Automatically integrates with newly created attack tools

Project Tasks

- Develop repository of real-world attacks
- Specify attack preconditions, postconditions, timing requirements, risk of detection, and risk of damaging target
- Create modular attack scripts
- Programmatically invoke attacks from command line
- Capture, parse and store attack output
- Develop attack tree generation algorithms
- Automatically generate threat representative attack trees
- Support automatic attack tree reconfiguration for new attacks
- Develop novel Autonomous Attack Engine & GUI
- Create new algorithms to optimize attack selections
- Develop techniques to visualize extremely large attack trees
- Develop non-IP network connectors
- Create hardware and software to streamline non-IP tests

AAAF automates cybersecurity Red Team tasks

AAAF automates cybersecurity Red Team tasks

Distribution Statement A: Approved for public release; distribution is unlimited.
AAFT attack tree generation identifies all possible combinations of available attack tools that can achieve a specified attack objective, identifying emergent attacks before adversaries do.

- **Synthesized Attacks**
- **Manually Encoded Attacks**

AAFT has synthesized emergent attacks using as few as 15 scripts.