GARNET

Graphical **A**ttack graph and **R**eachability **N**etwork **E**valuation **T**ool*

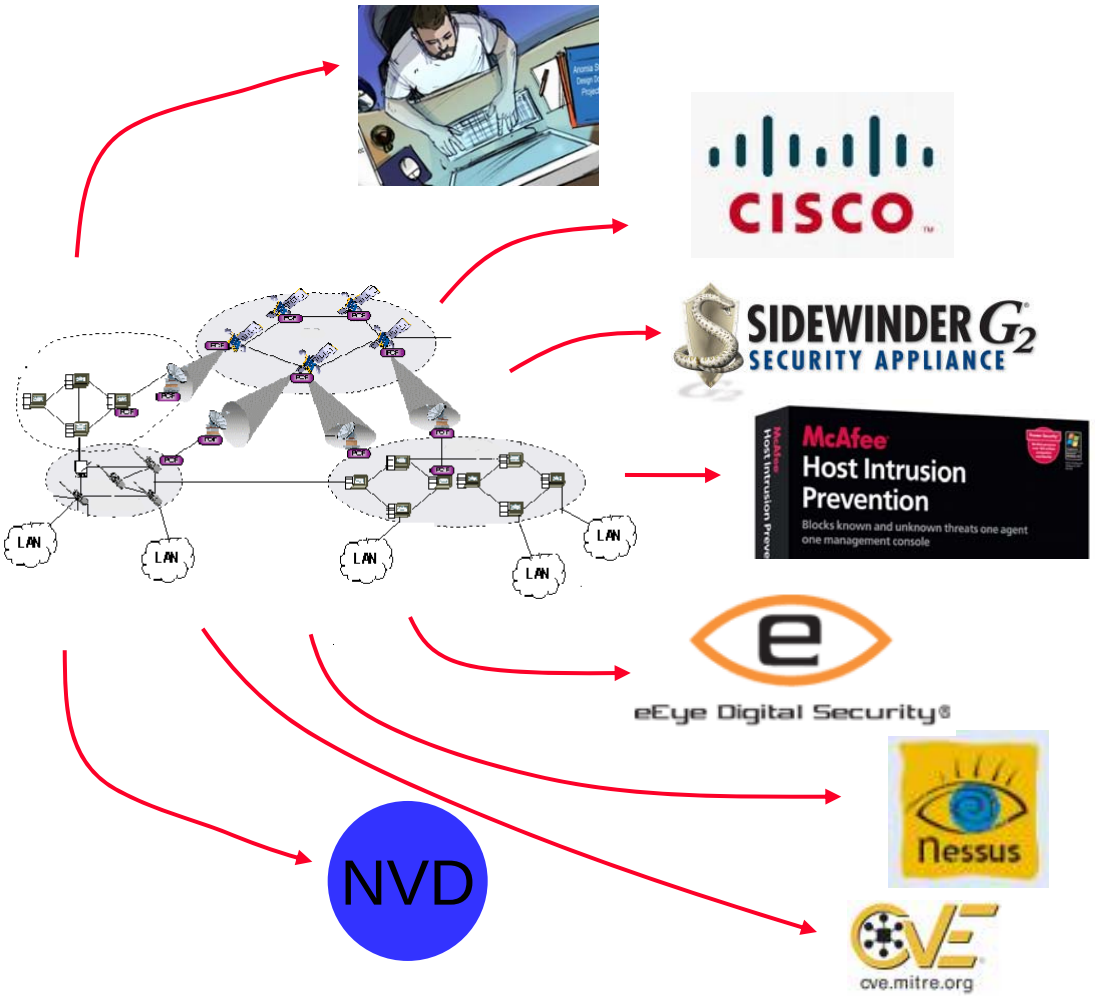
Leevar Williams, Richard Lippmann, Kyle Ingols

MIT Lincoln Laboratory

15 September 2008

*This work is sponsored by the United States Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government

A Defender's Primary Advantage is Detailed Network Knowledge – This Needs to Be Used Effectively!



Specify Asset Values and Adversary

Define Network Topology and Filtering Rules

Discover Vulnerabilities

Define Vulnerability Requirements/Effects

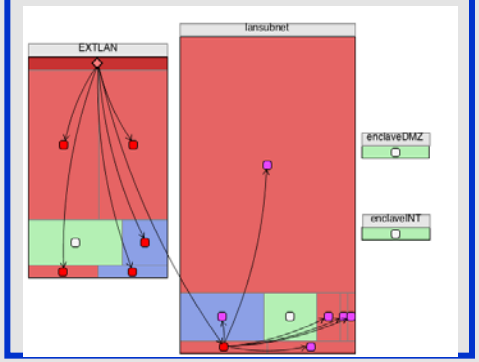
A Tool Named NetSPA Integrates This Data and Supports "What If" Experiments

Infer Network Topology

Construct Network Model

Compute Host-to-Host Reachability

Attack Graph



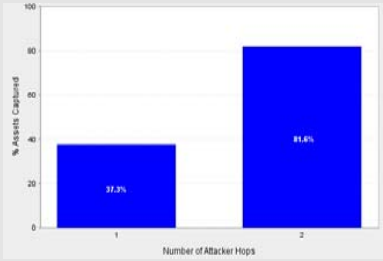
What If ?

Adversary

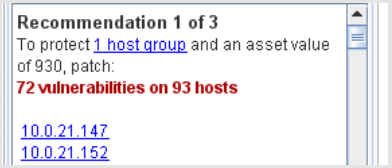
Defenses

Host Asset Values

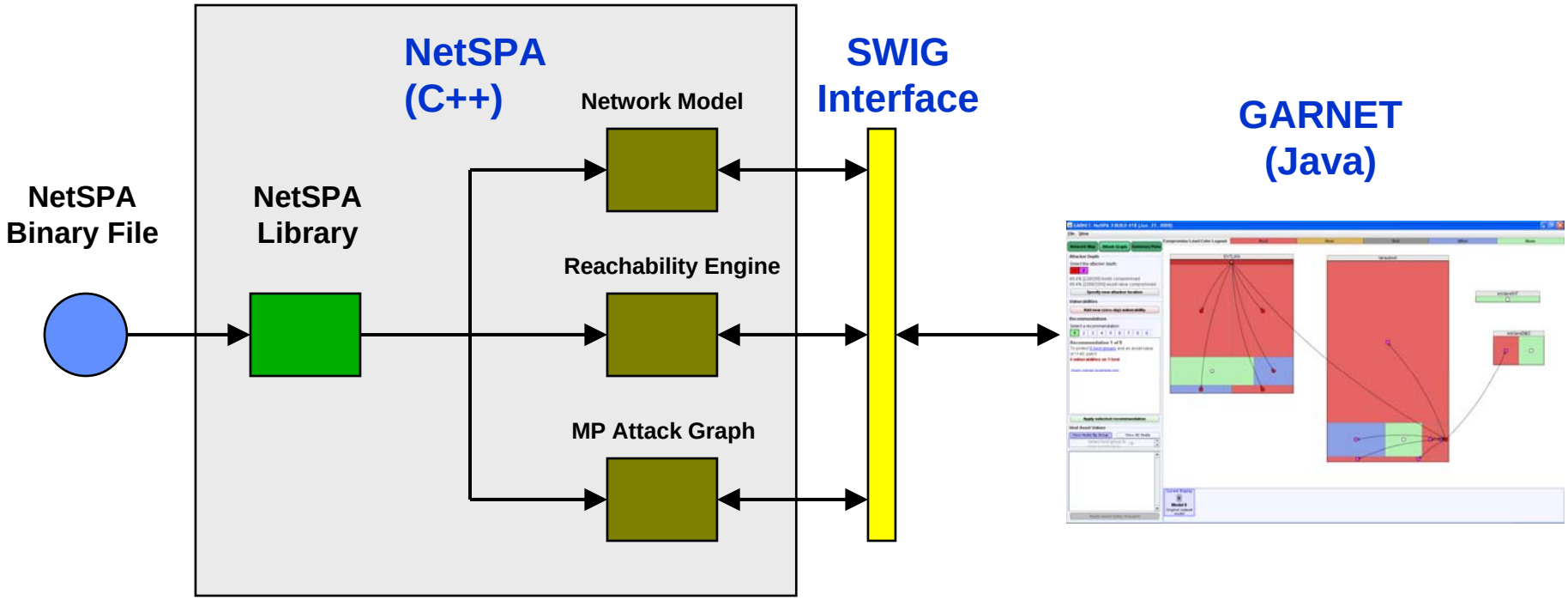
Metrics



Recommendations



GARNET Uses NetSPA to Provide Rapid Interactive Response

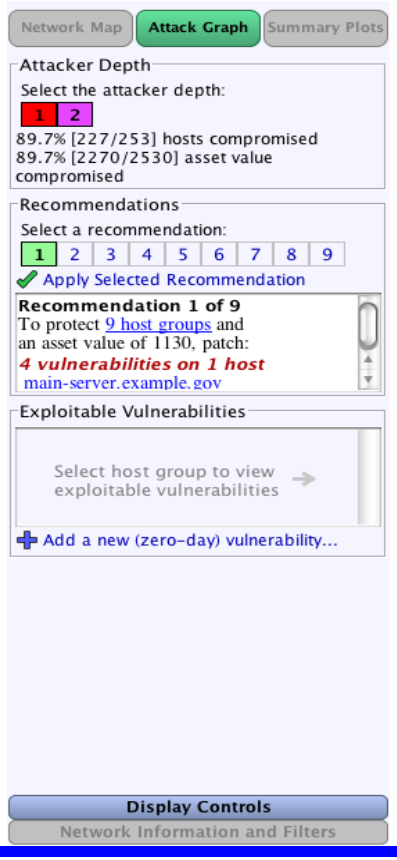


- GARNET is implemented in Java using Swing
- NetSPA loads binary network model and produces C objects for data access
- Java and C code communicate using SWIG Toolkit

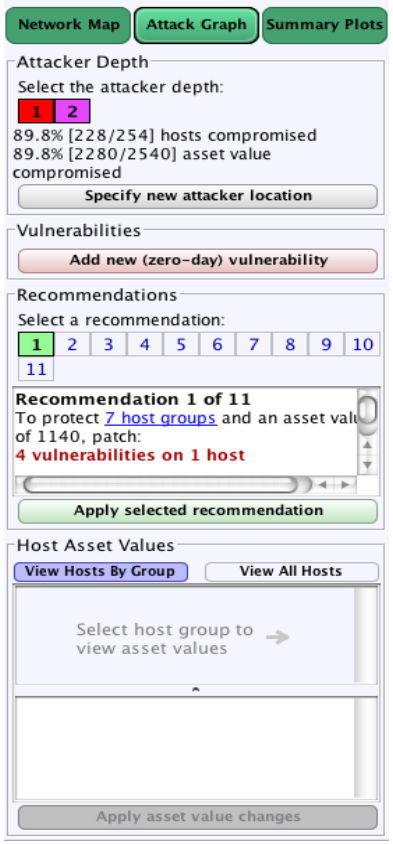
A Heuristic Evaluation Greatly Improved GARNET's Ease of Use

- **Five participants evaluated 9 networks with the initial GUI and provided recommendations using protocols developed by (Nielsen and Molich, 1990; Nielsen 1994)**

Original



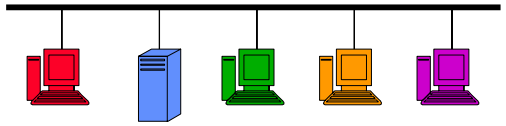
Revised



- **20 Major Changes**
- **All "What-If" controls unified in Attack Graph Panel**
- **All network details placed in Network Map Panel**

Security Compared by Determining Adversary Cost to Achieve Goals

- **Emulate MORDA (Mission Oriented Risk and Design Analysis) procedures**
- **Need to model**
  - **System (Network)**

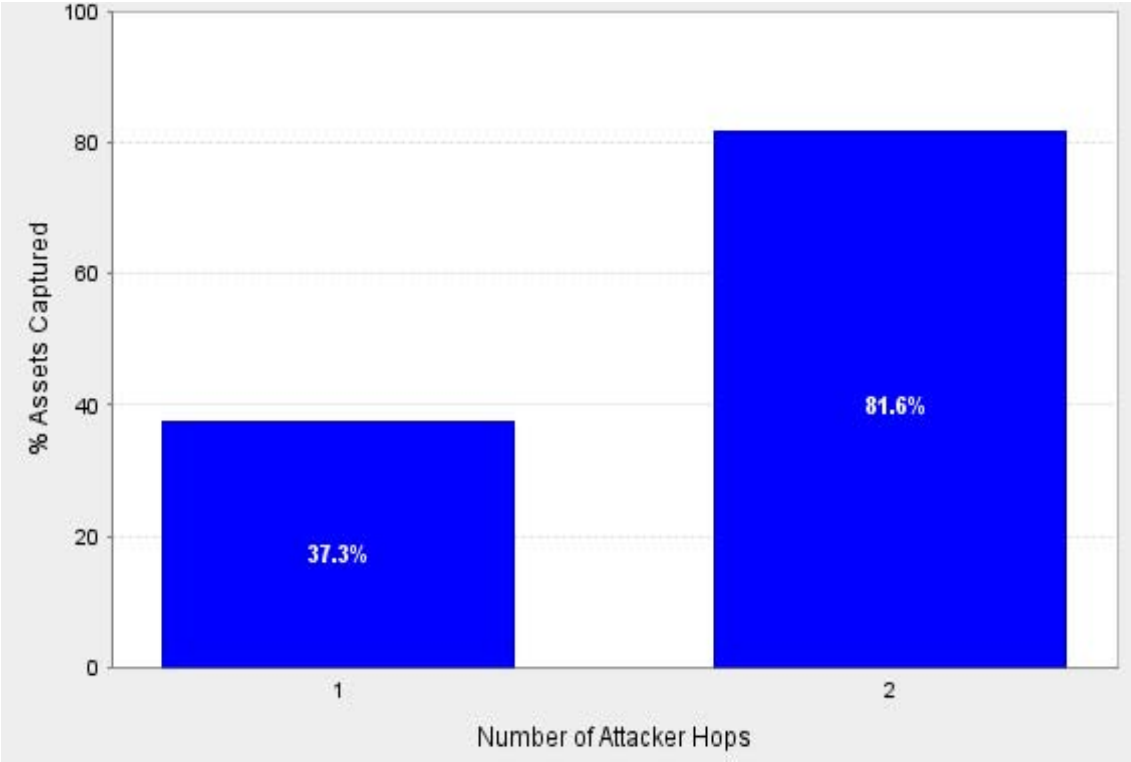


  - **Adversary goal**

  - **Defenses**

  - **Adversary**

We Currently Assess Adversary Cost Using Three Different Metrics



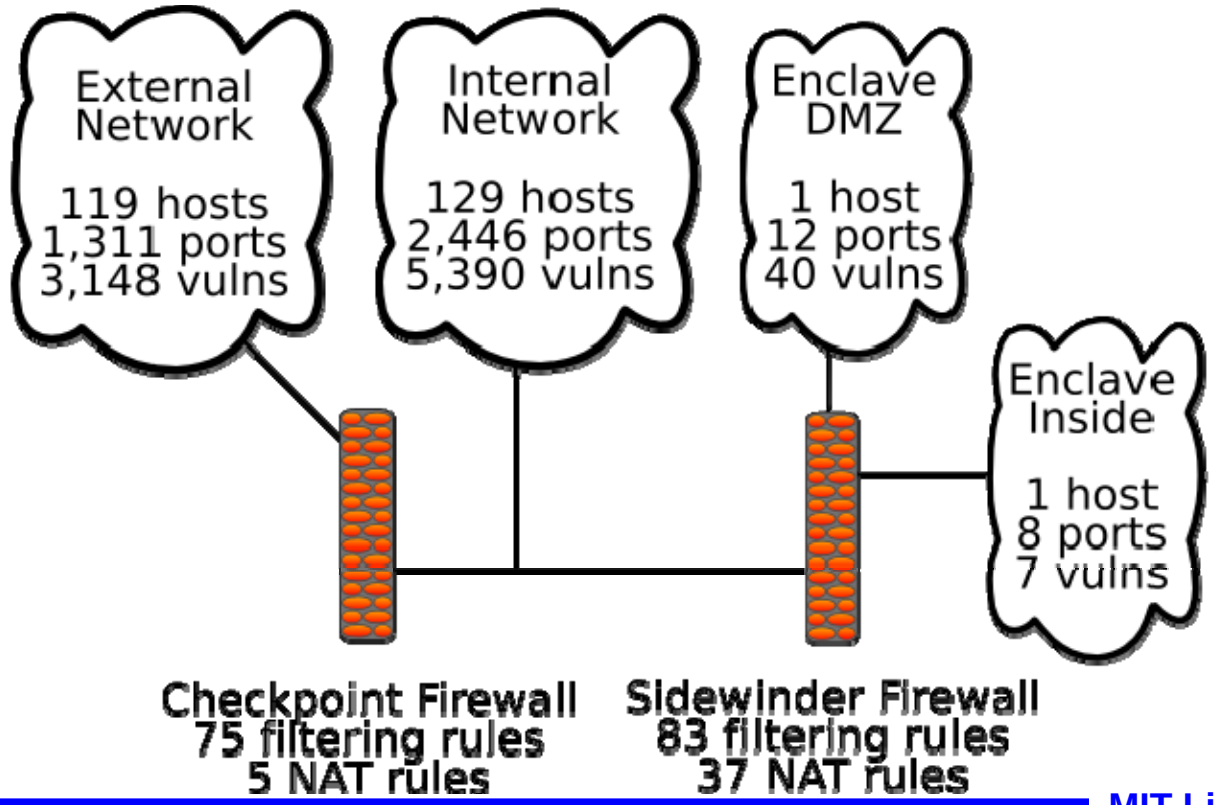
- Number of hops to reach assets
- Number of unique exploits that are required
- Cumulative CVSS attack complexity

Analyze Network Security Using Escalating Adversary Models

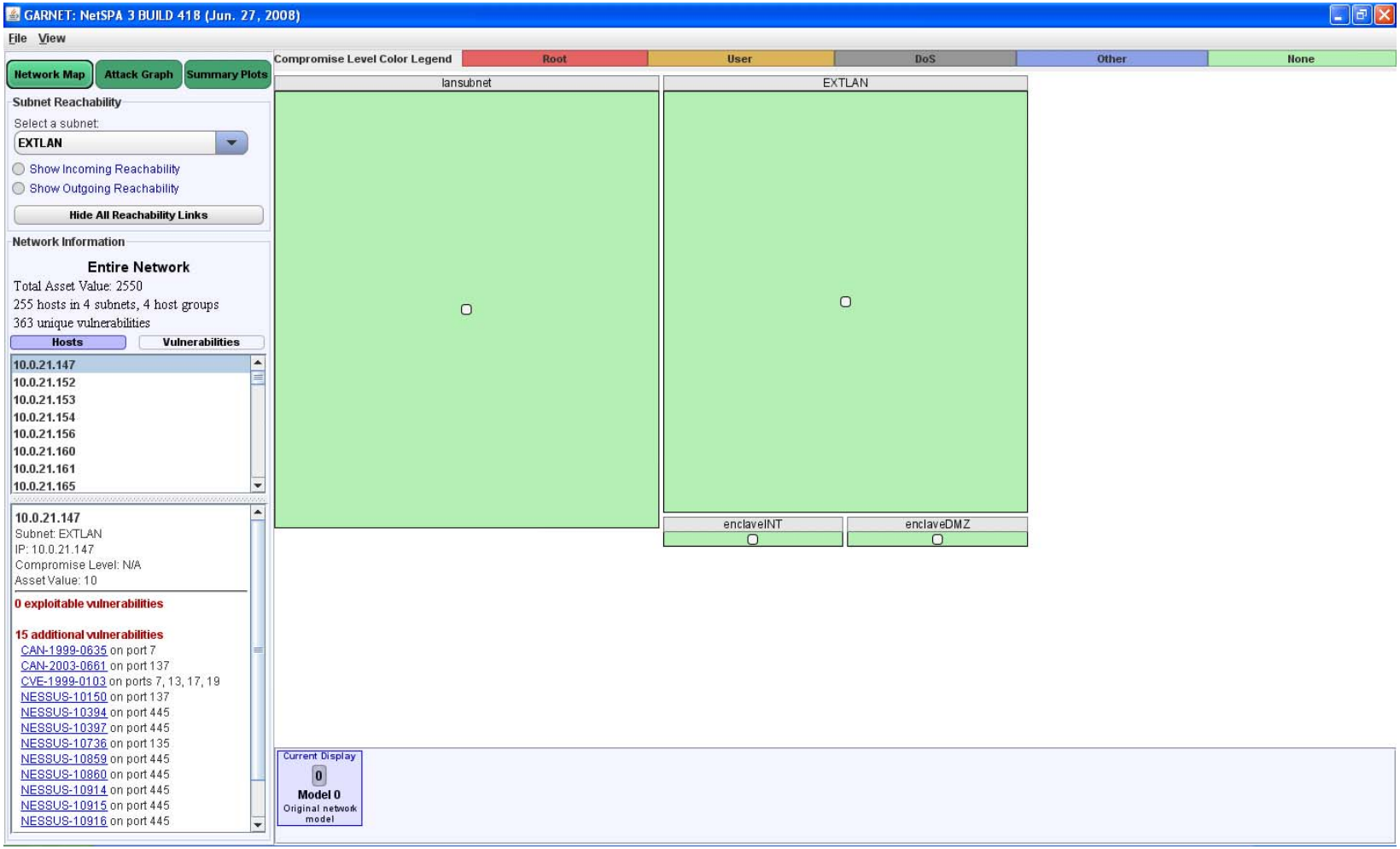
1. **Script User**
   - Has an exploit for all known vulnerabilities
2. **Single Zero-Day**
   - Able to create a zero-day exploit for the one application server on this network that provides the most access
3. **Comprehensive-Zero Day**
   - Able to create a zero-day exploit for all application servers on this network
   - Any host that can be reached can be compromised

Example Network

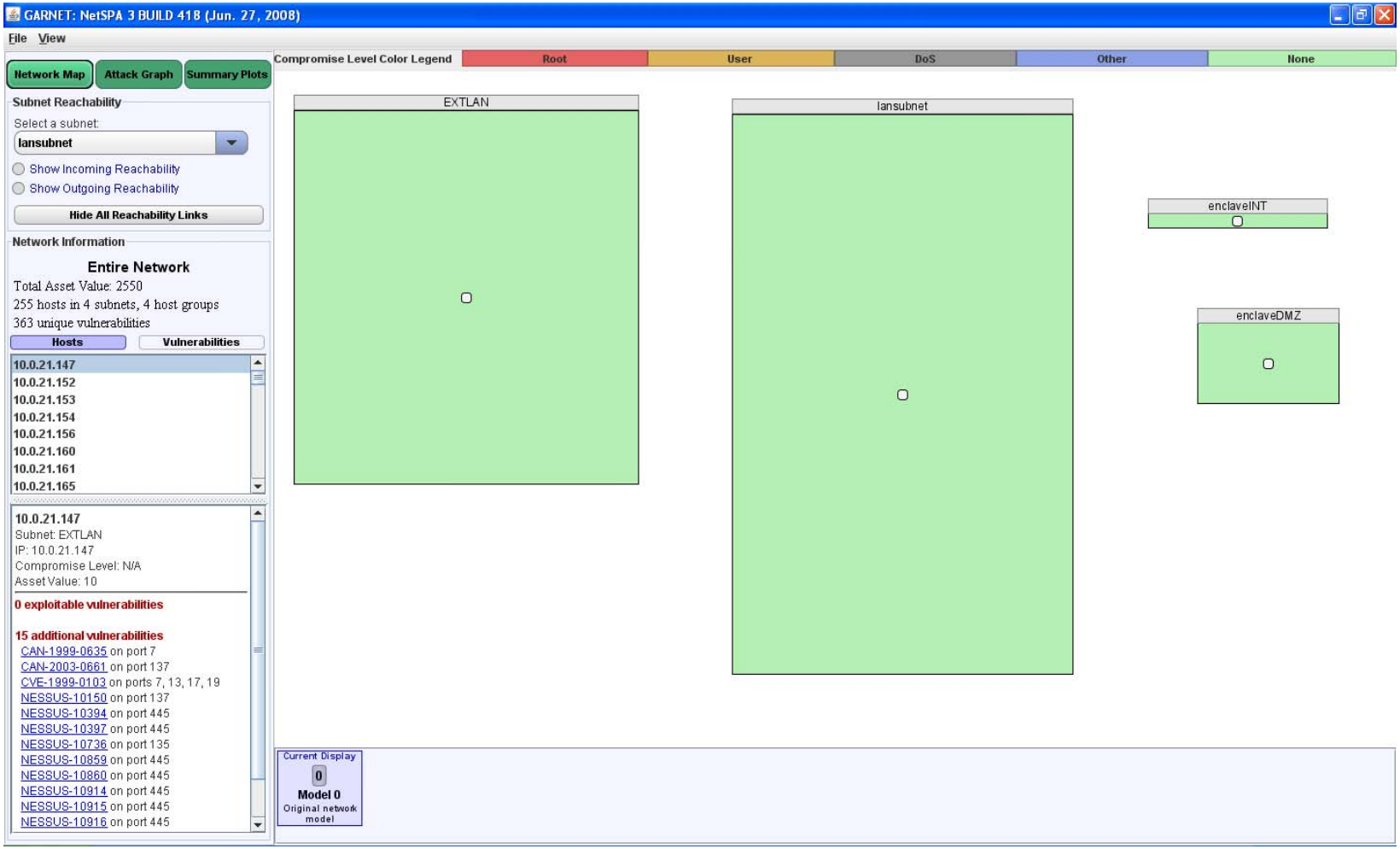
- **Anonymized data from real, field test network**



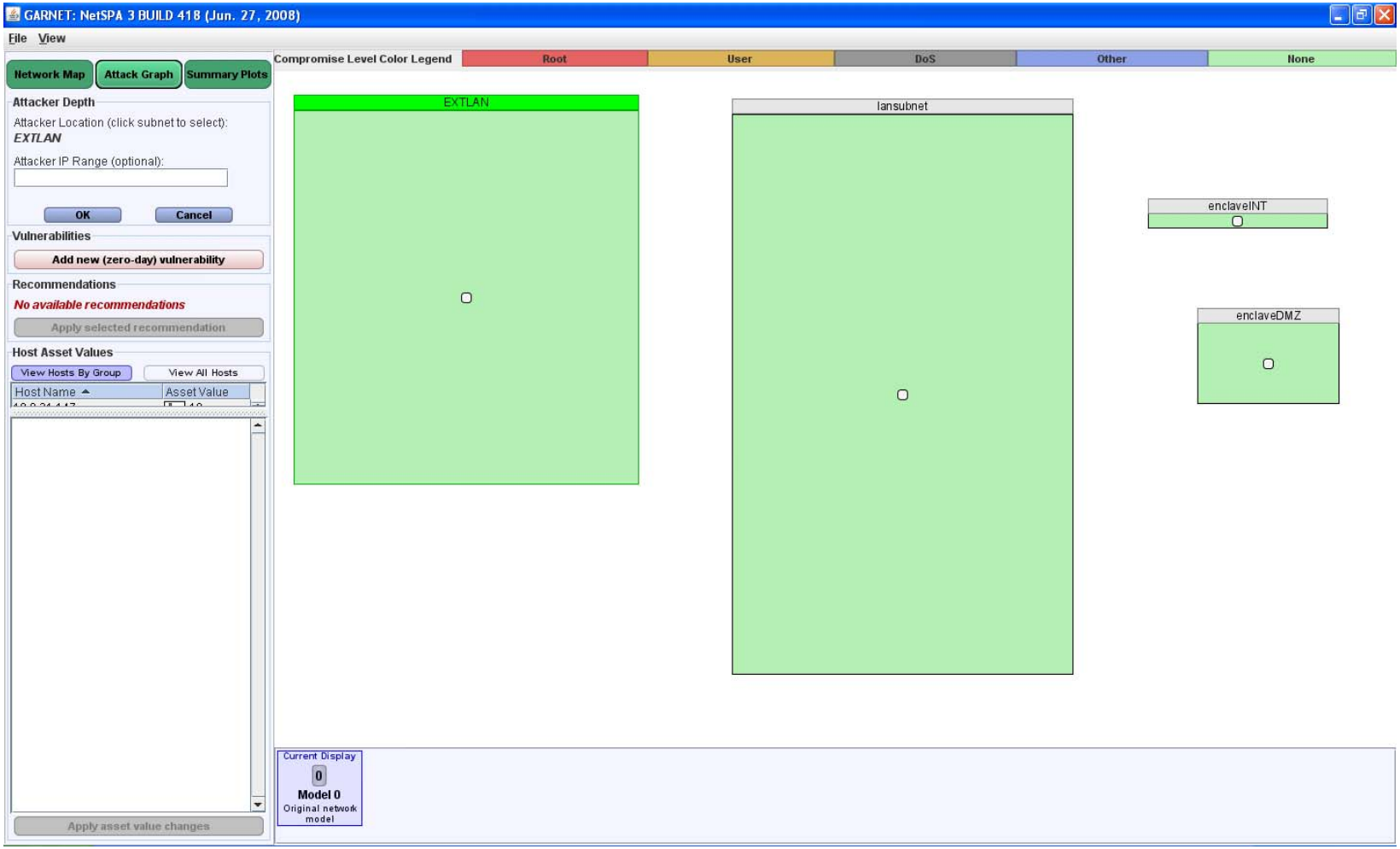
Demonstration: Loaded Network



Demonstration: Rearranged Network

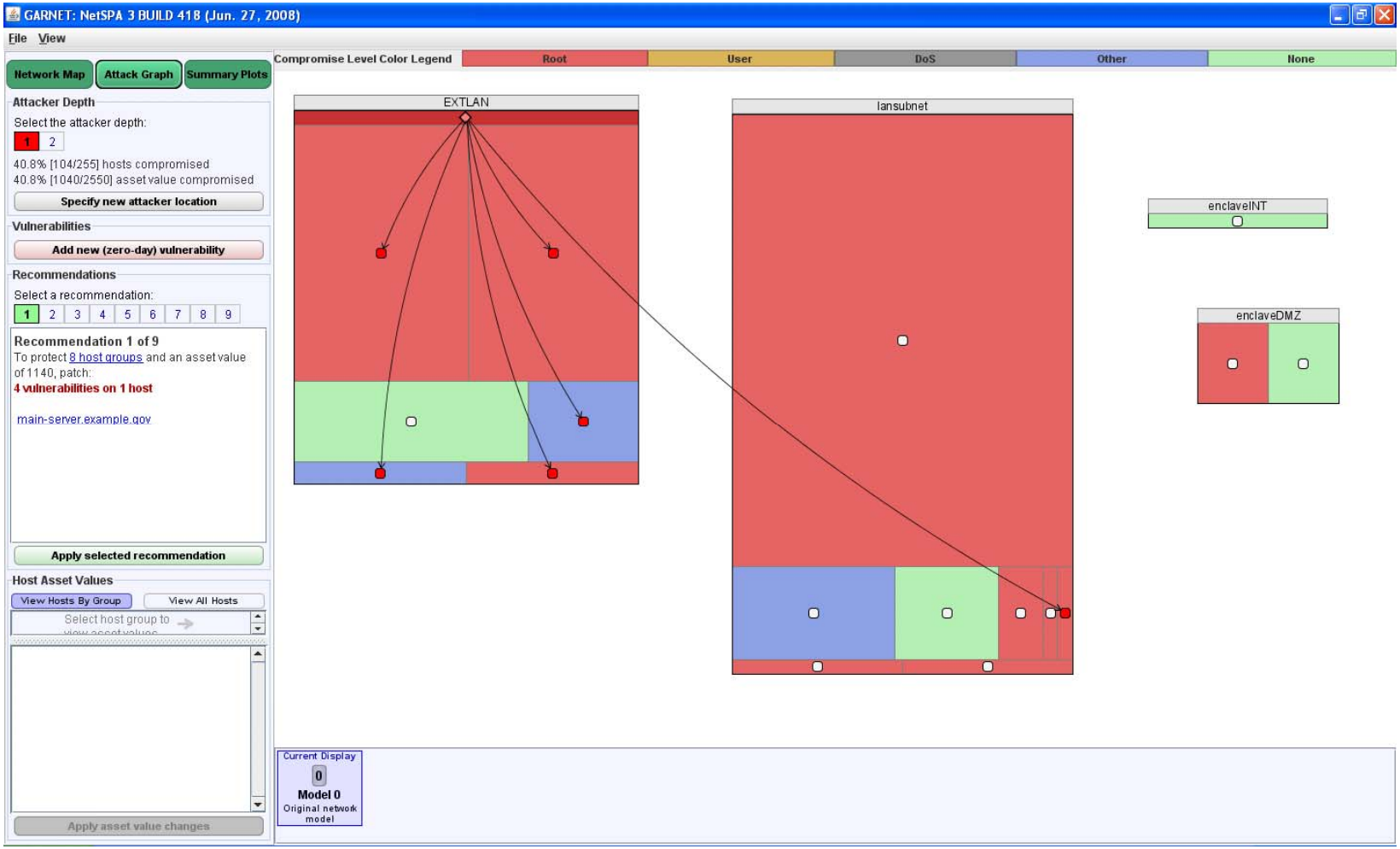


Demonstration: Attacker Start Select

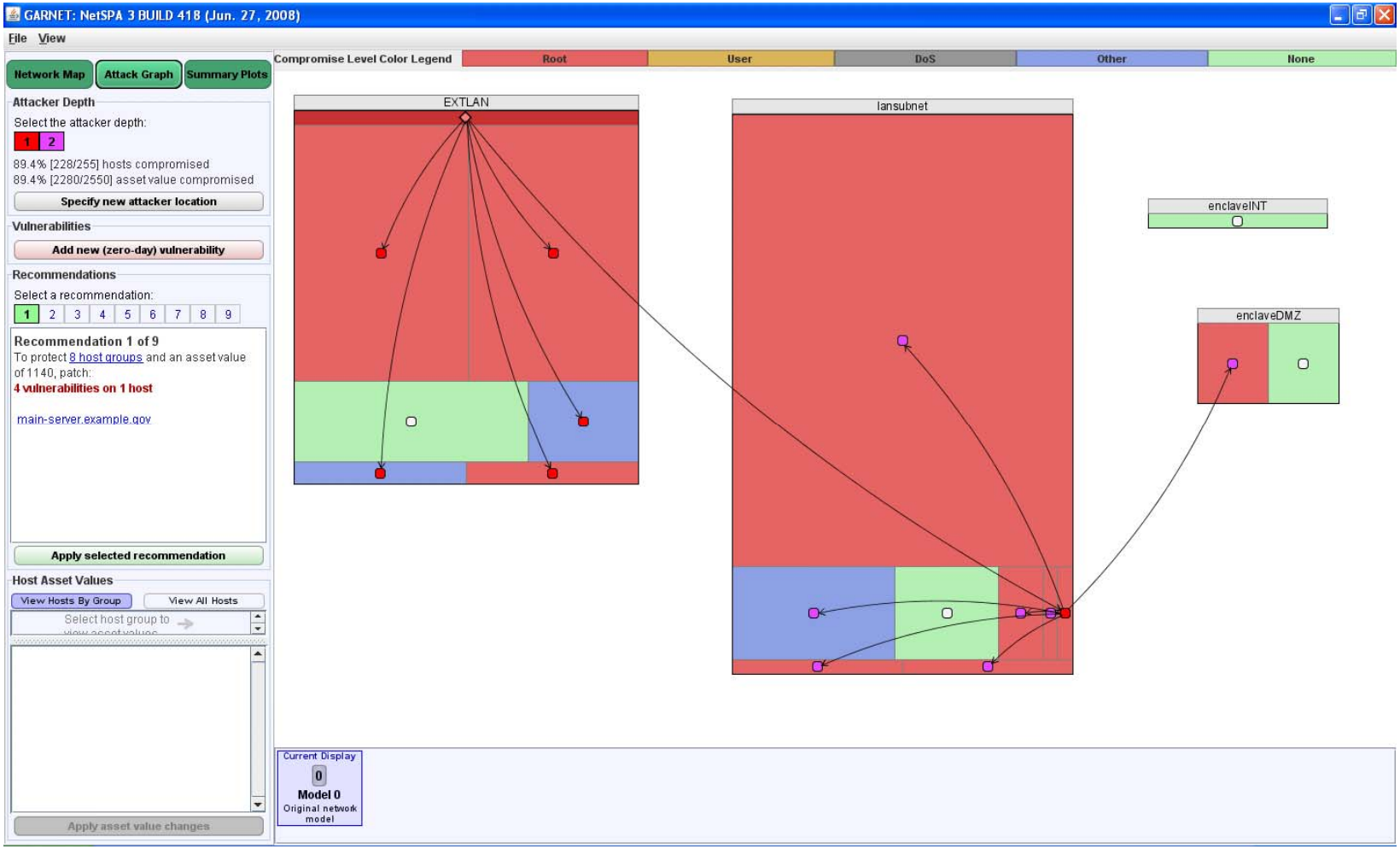


Demonstration: Attack Result

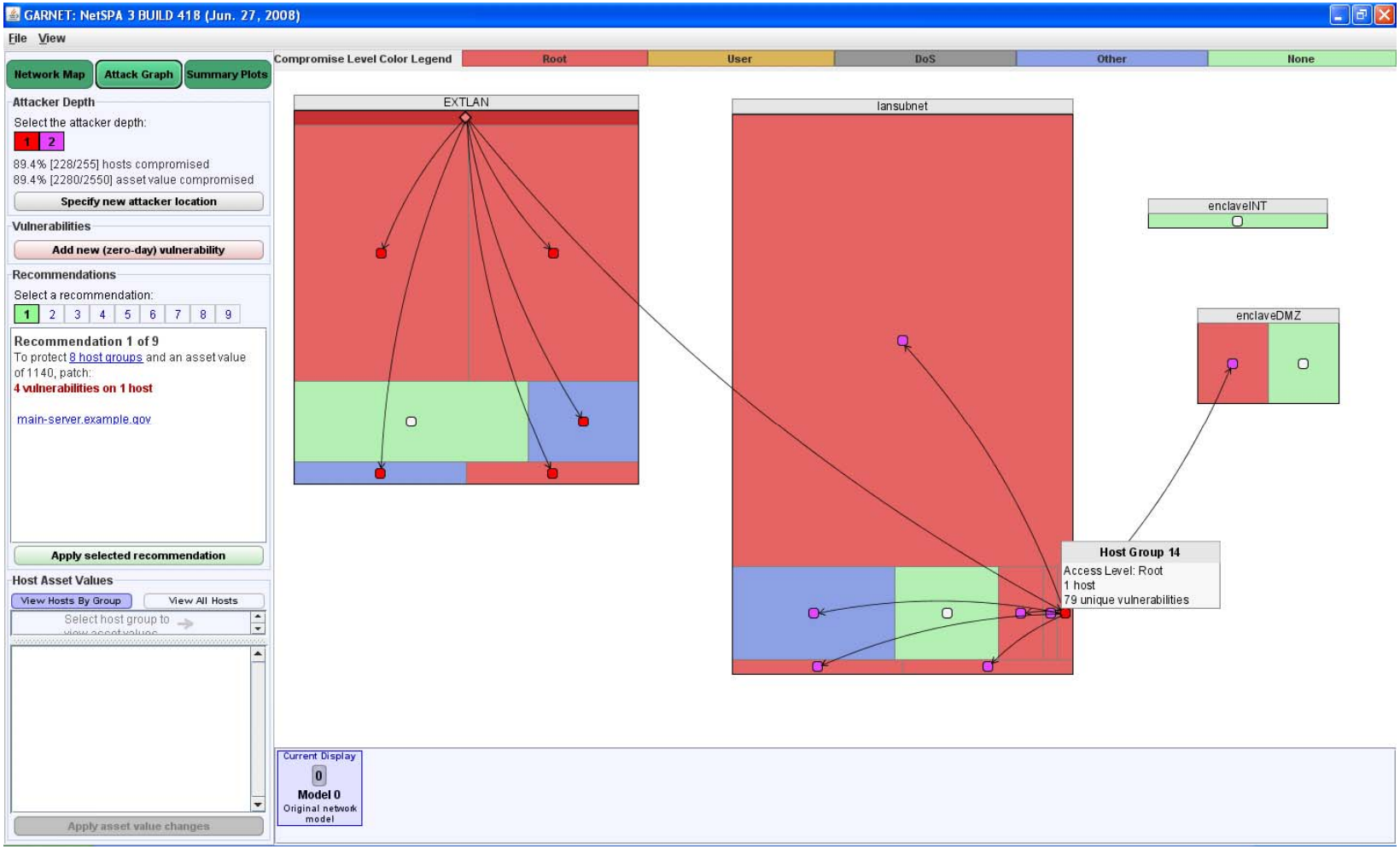
Demonstration: Attack, Step One



Demonstration: Attack, Step Two

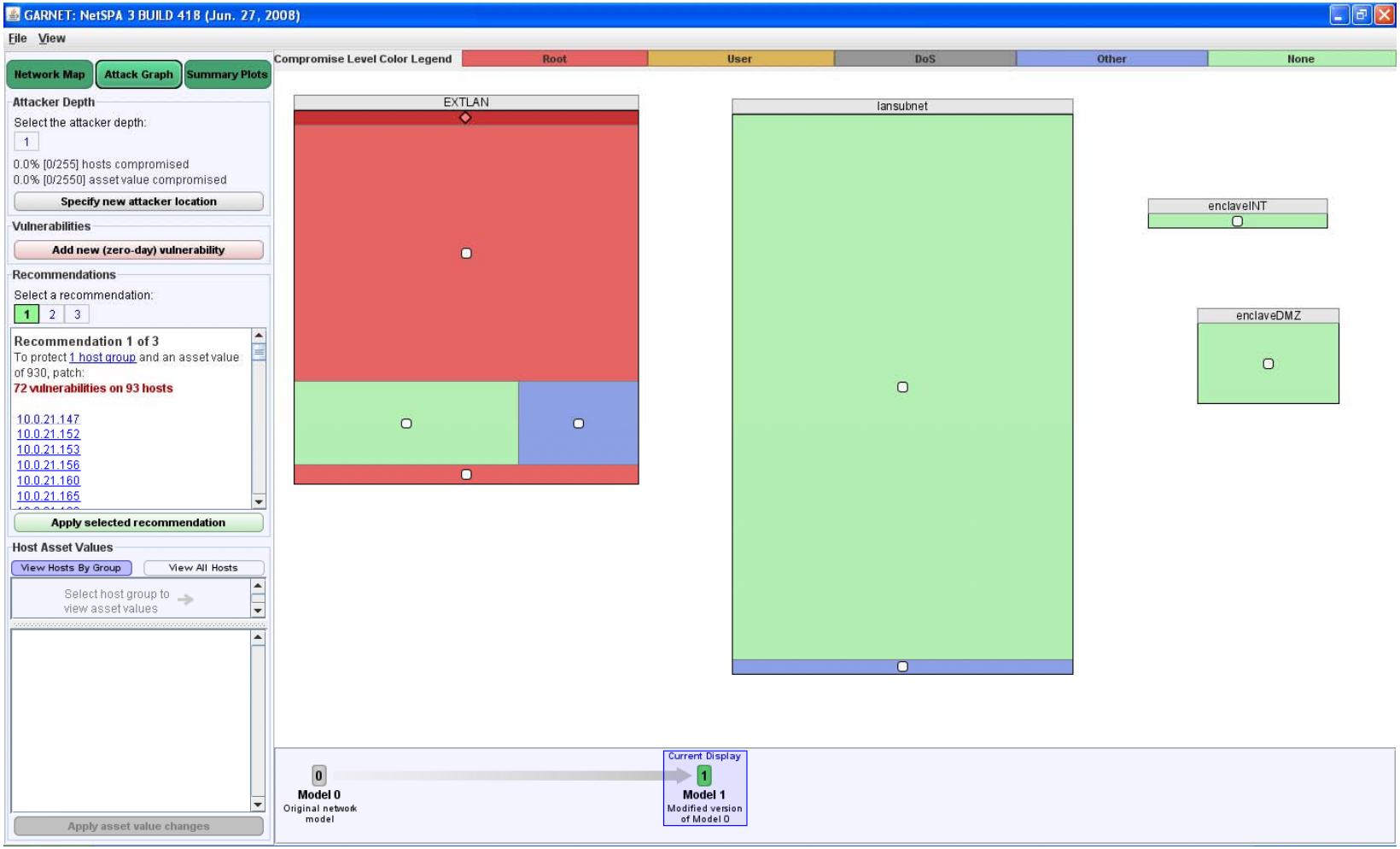


Demonstration: Stepping-Stone

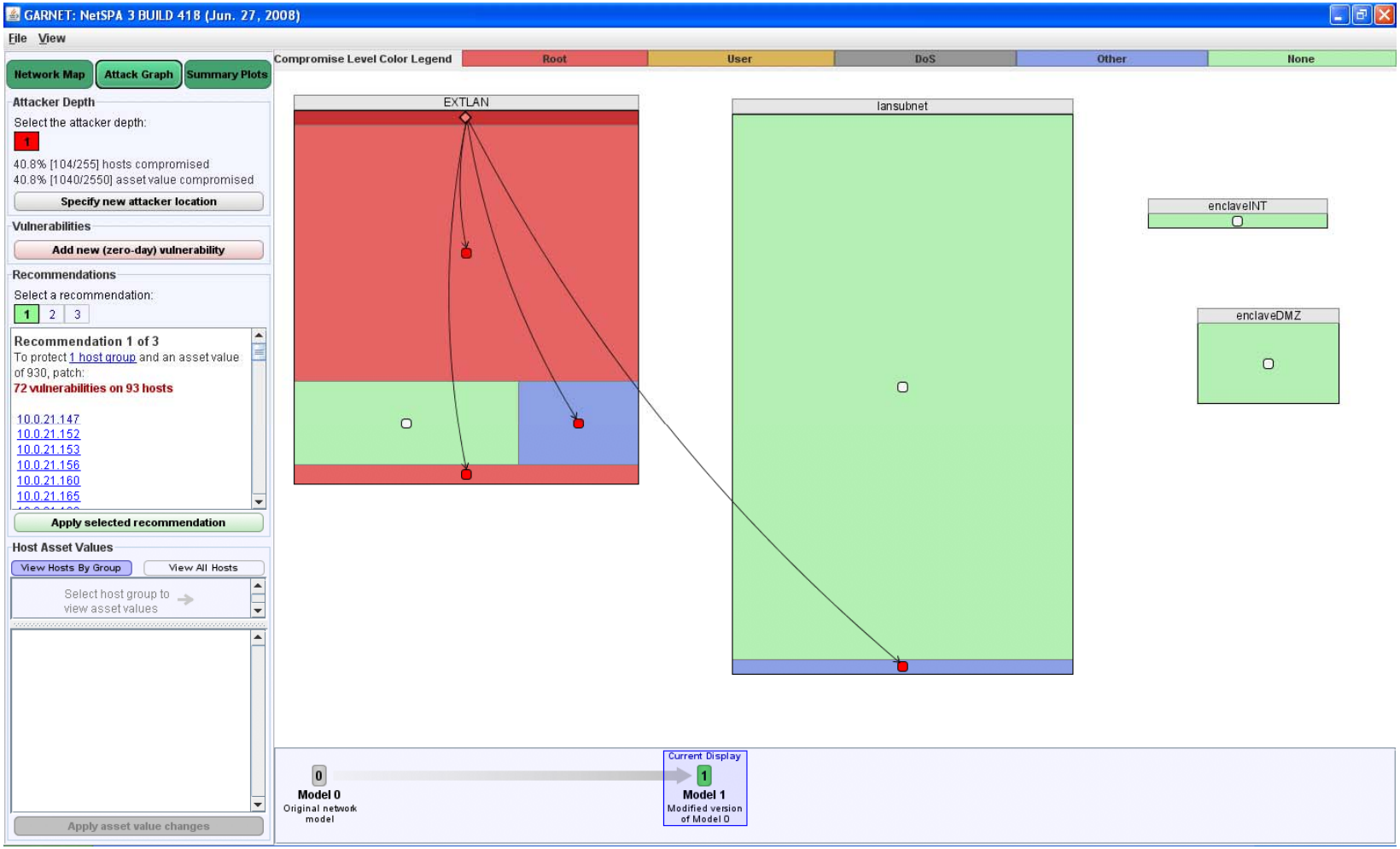


Demonstration: Reachability Trace

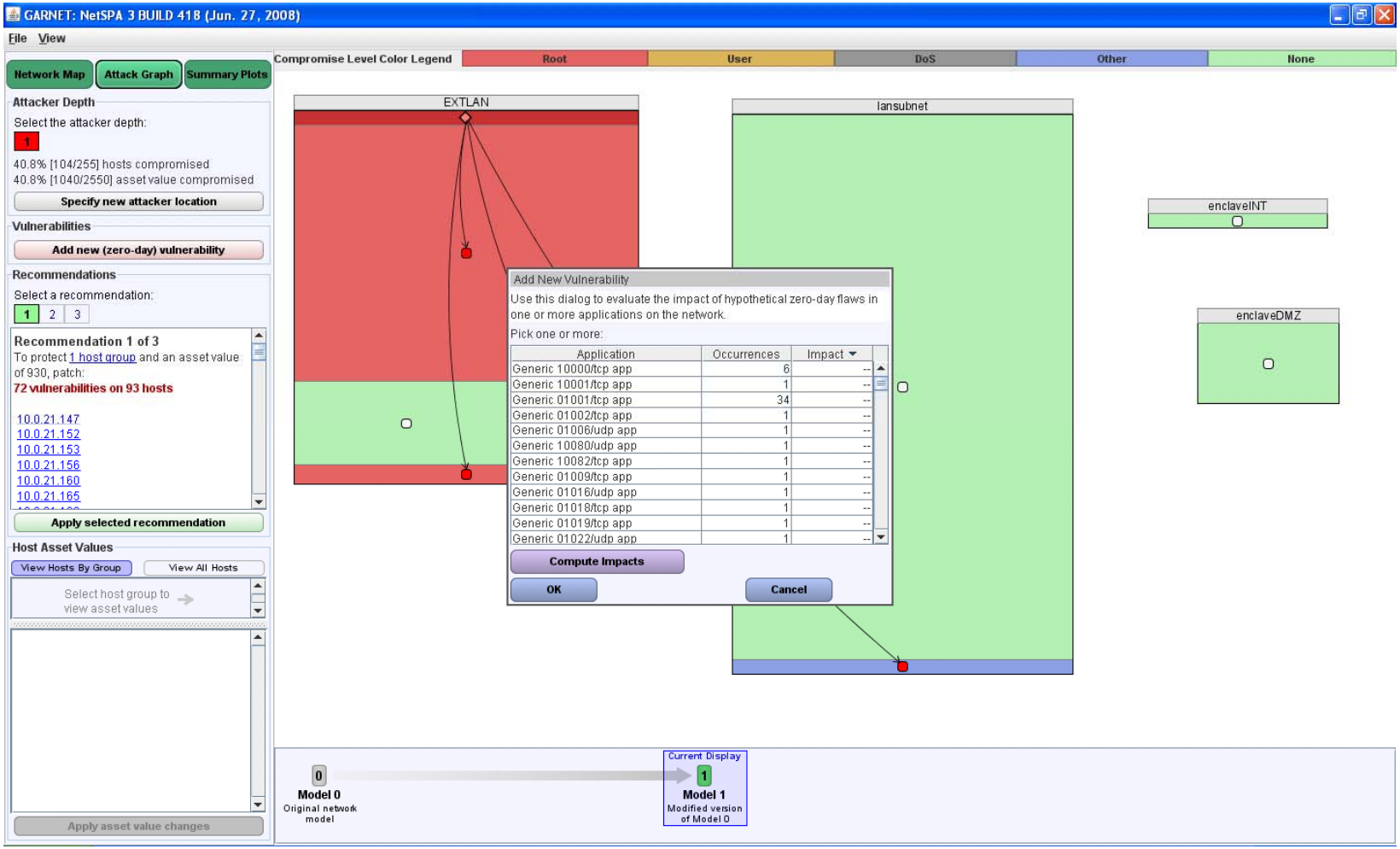
Demonstration: Recommendation Used



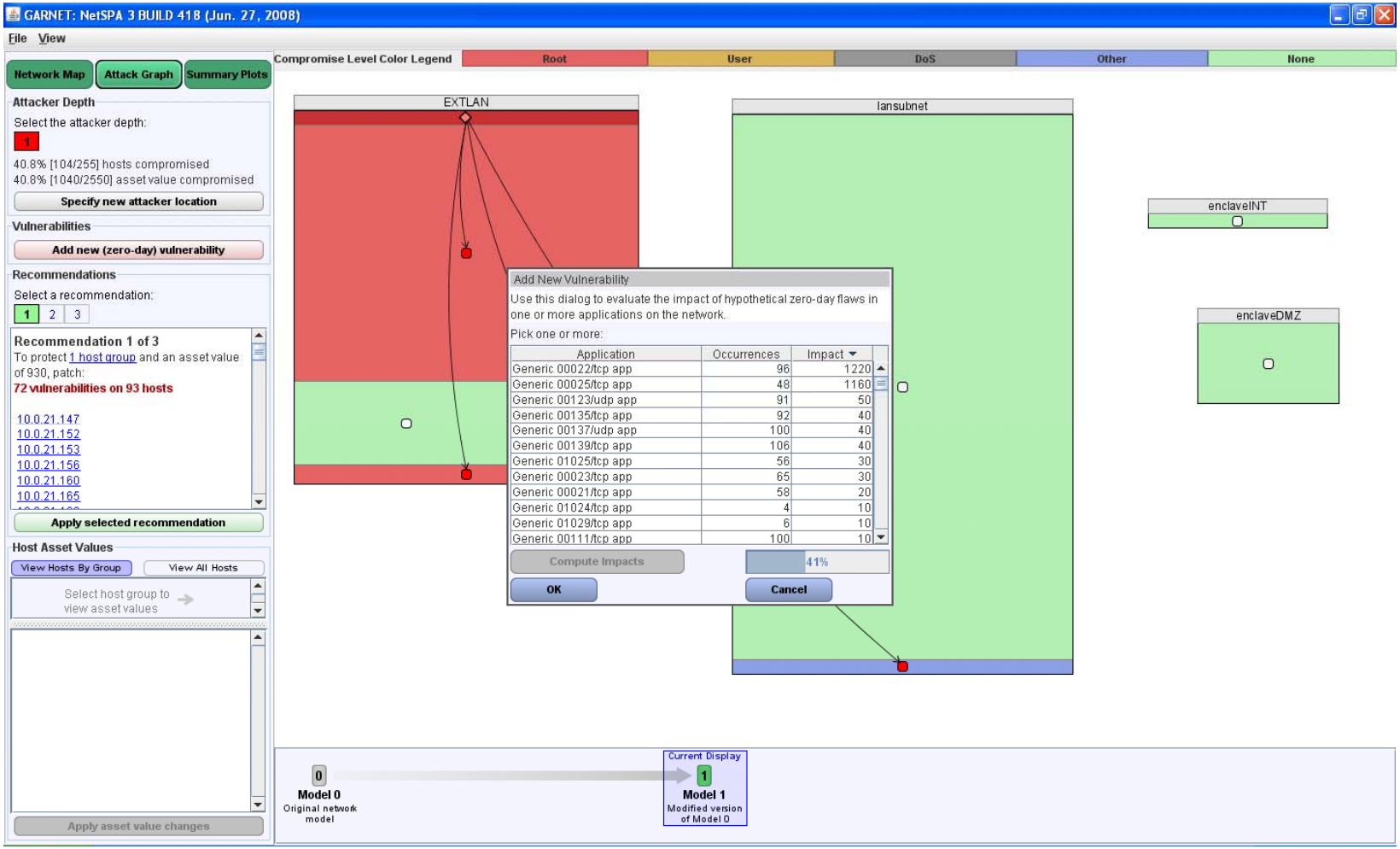
Demonstration: Recommendation Used



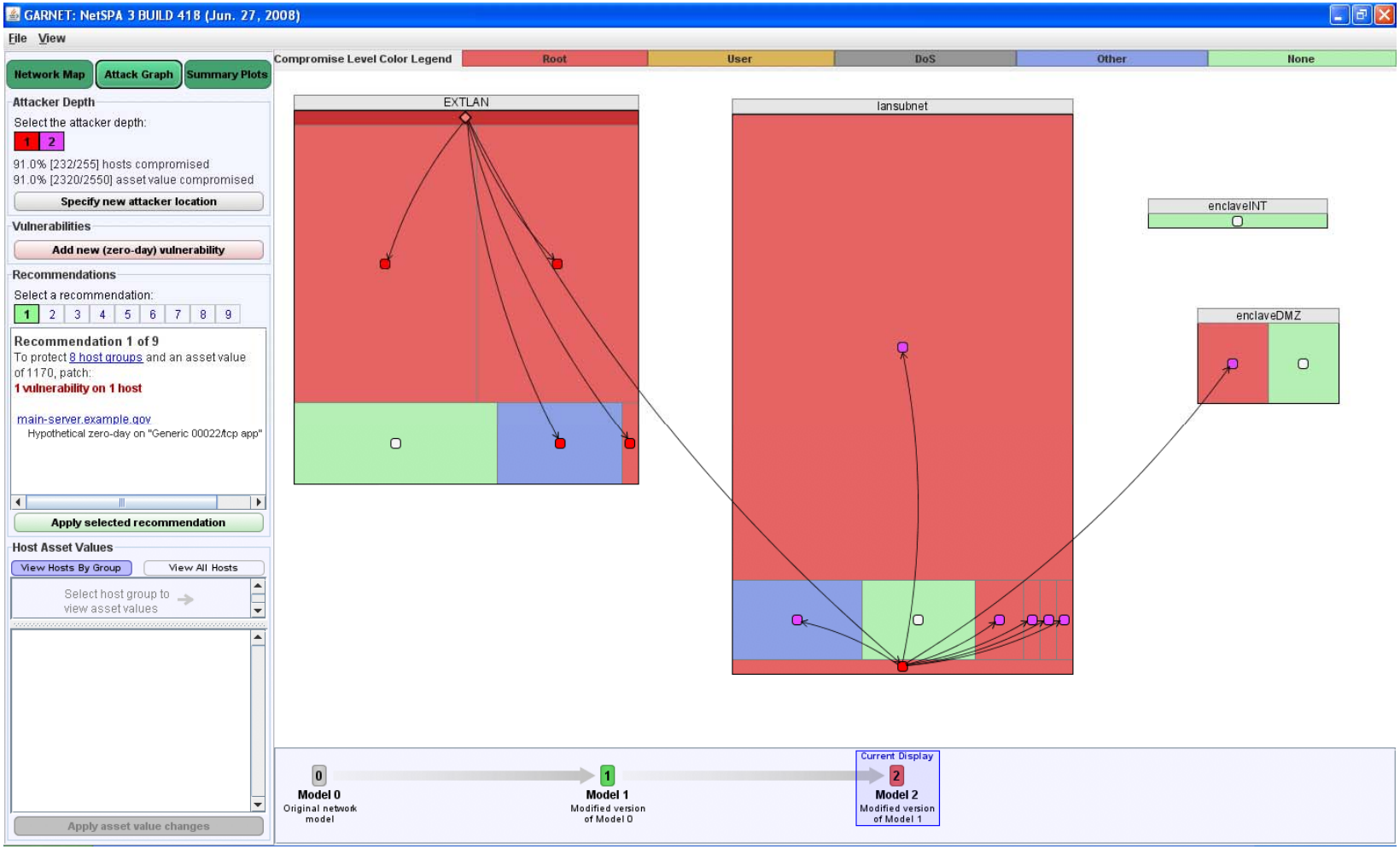
Demonstration: Zero-Day Adversary



Demonstration: Zero-Day Adversary
Select Worst-Case Zero-Day Application

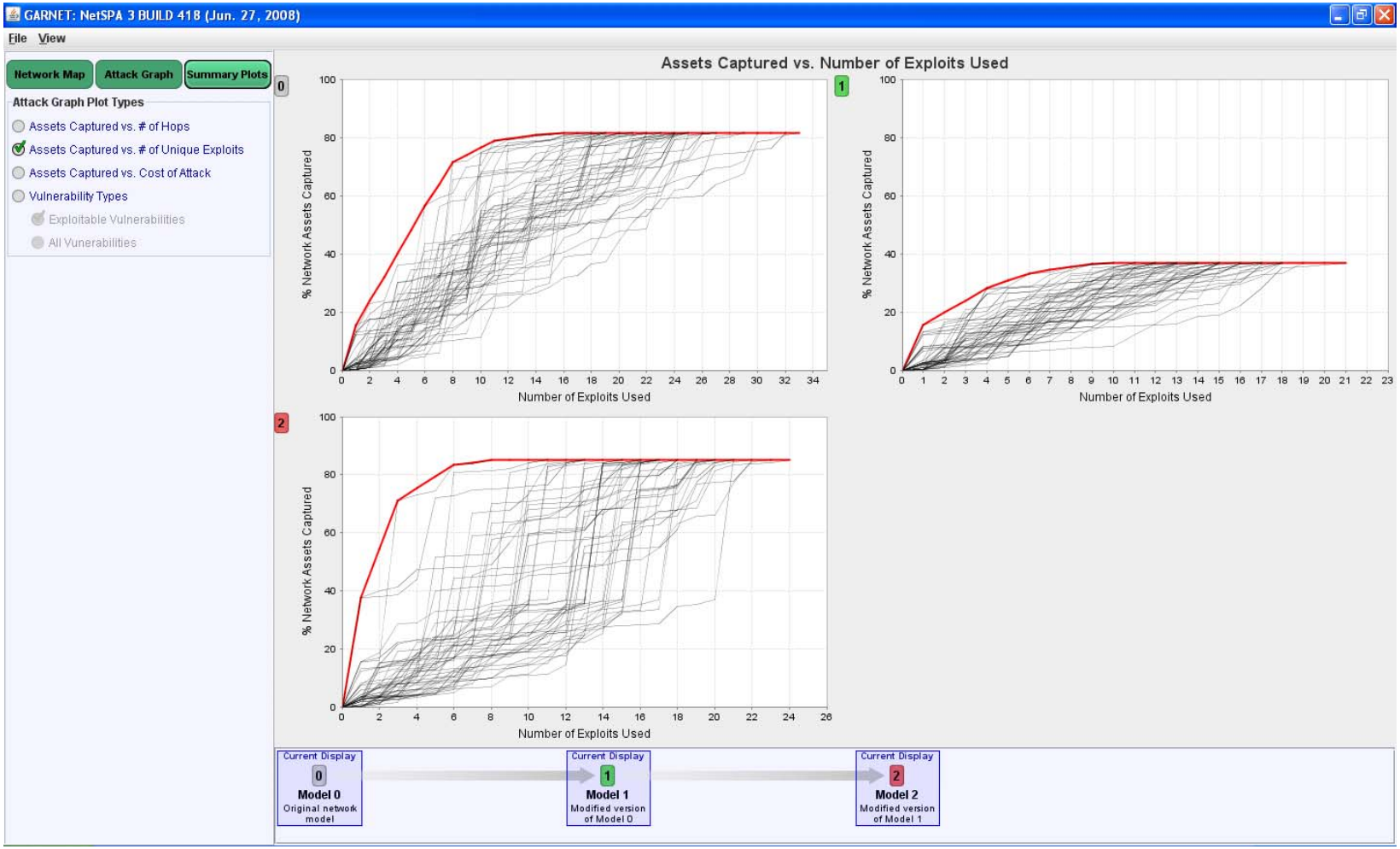


Demonstration: Zero-Day Adversary Using Port 22/tcp



Demonstration: Assets Capture Versus Hops

Demonstration: Assets Captures Versus Unique Exploits Required



GARNET Summary

- Rapid Interactive response
- Easy to use and intuitive GUI
- Supports “What-If” experiments
- Computes and displays …
  - Recommendations
  - Security metrics (attacker effort)
  - Host-to-host reachability
  - Attack graphs

Future Work

- Adversary model
  - Visualize client-side attacks
  - Import and use data on trust relationships
- Extend Countermeasure Models
  - Intrusion prevention systems
  - Proxy firewalls
- Efficiently model endpoint or host-based firewalls
- Display physical/logical network topology including firewalls, routers, and switches