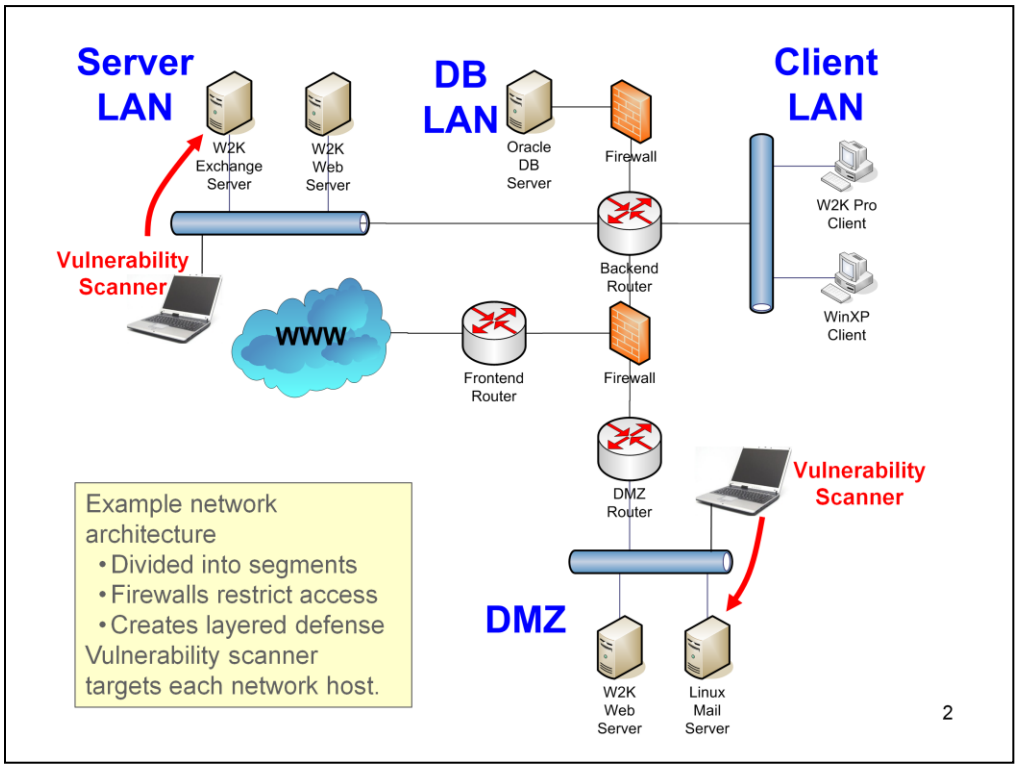
Combinatorial Analysis Utilizing Logical Dependencies Residing On Networks (CAULDRON)

Dr. Steven Noel

Center for Secure Information Systems
George Mason University

csis.gmu.edu

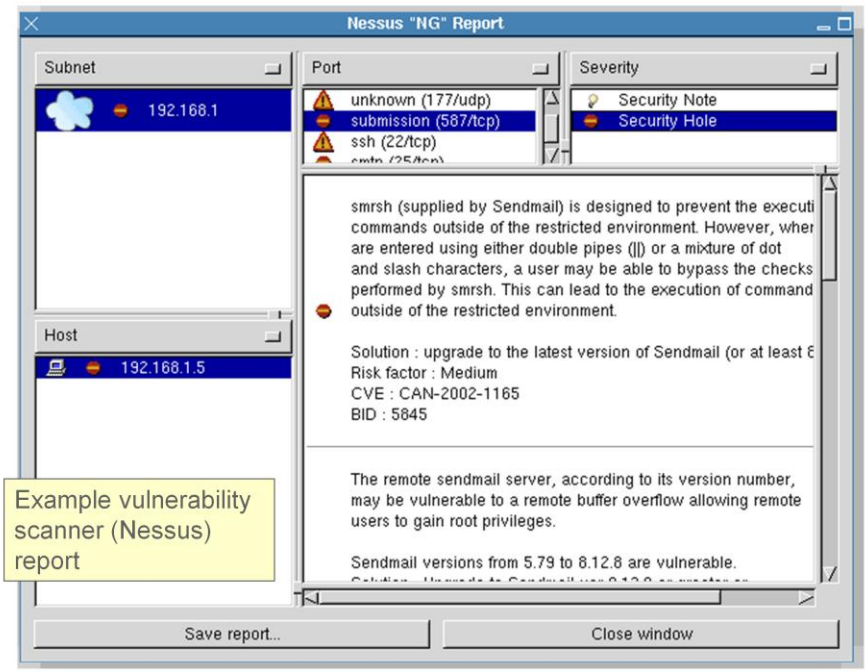


This shows a typical architecture that enterprises use to secure their networks:

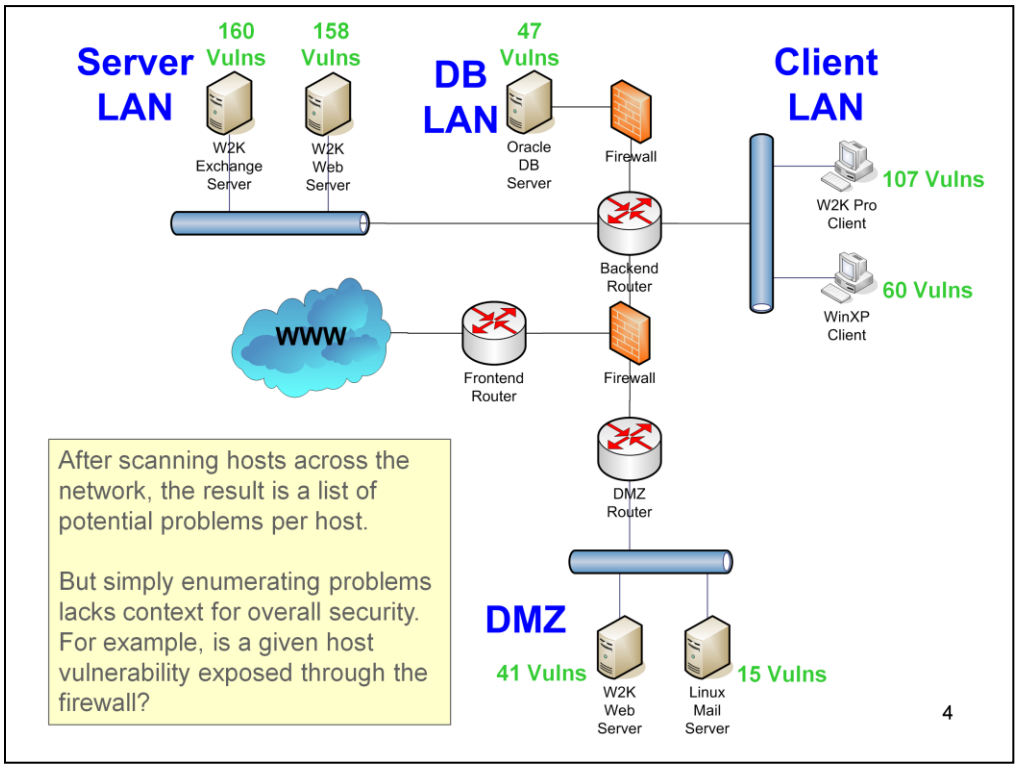
- The network is divided into a number of segments
- Firewalls restrict access between segments
- This creates a layered defense

A vulnerability scanner tool detects known software vulnerabilities. A scanner is put on the network, a host is targeted for a scan.

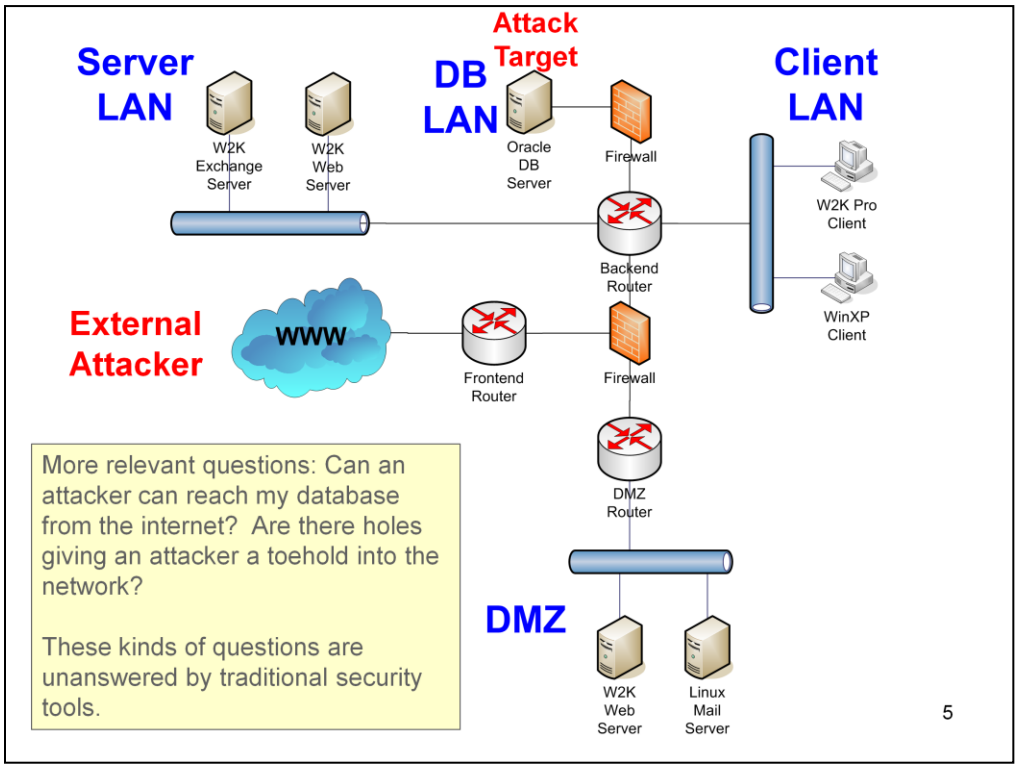
Then, to find problems somewhere else, target another host, run another scan, and generate another report.



This is an example vulnerability scanner (Nessus) report. It is a textual listing of potential security problems for a target host.



The result is a list of problems for each machine.

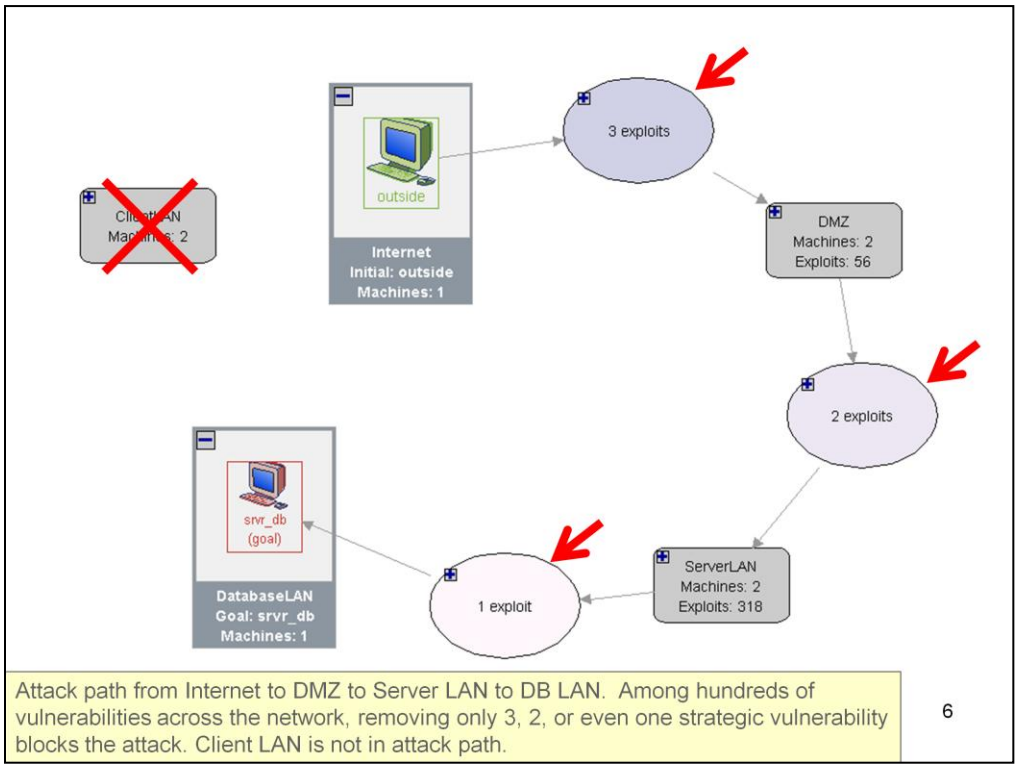


But what you really want to know is things like whether an attacker can reach my database server from the internet.

For example, does the firewall block these vulnerabilities?

Or are there real holes that an attacker can use to gain an initial toehold into the network (say, from the DMZ), further penetrate the network, and lead to devastating consequences?

These kinds of questions are completely unanswered by traditional security tools, which simply enumerate potential problems.

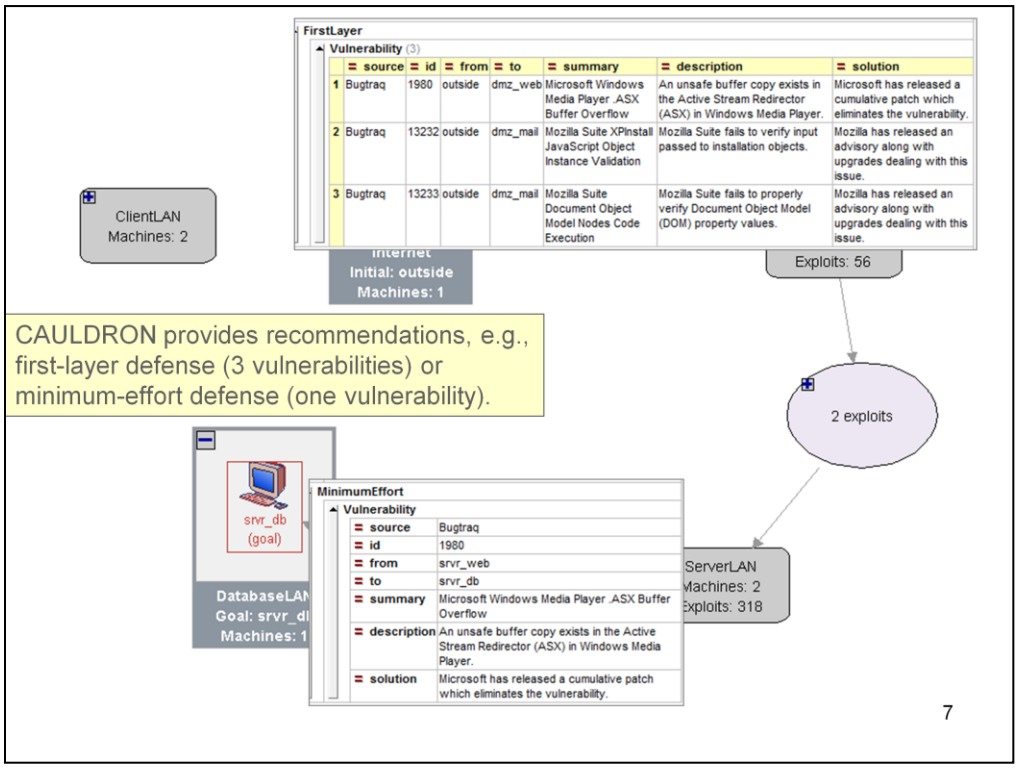


This is what CAULDRON shows.

It shows that it is indeed possible to reach the DB server from the internet (from internet to DMZ to Server LAN to DB LAN).

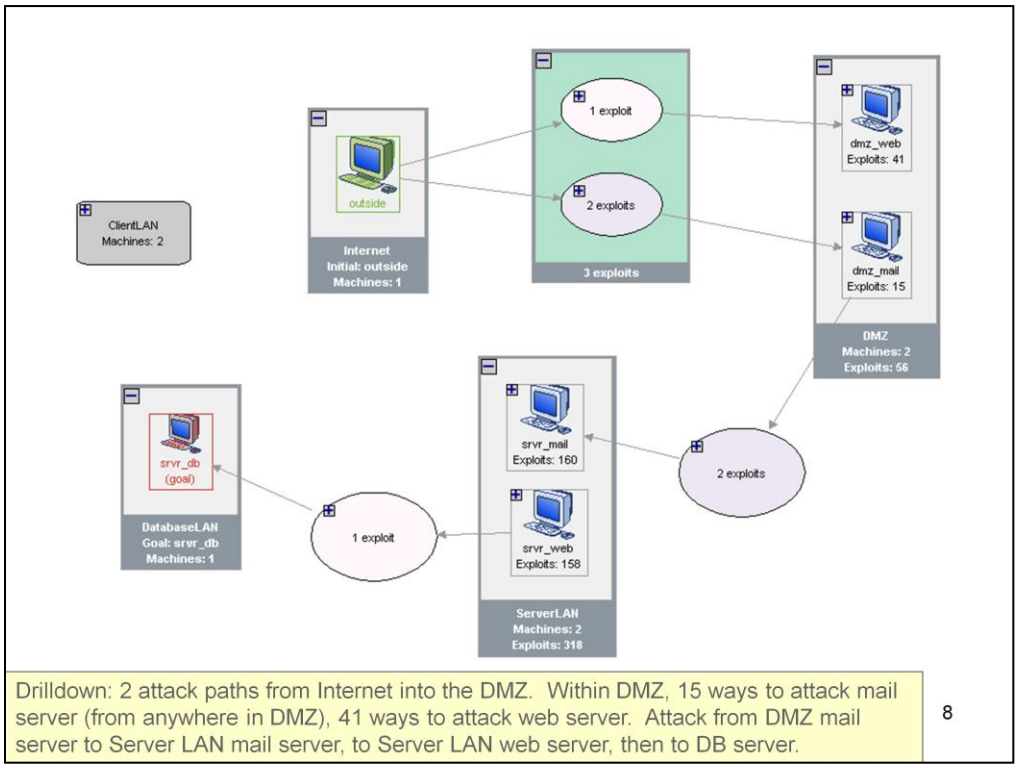
It shows that among the hundreds of vulnerabilities across the network, removing only 3, 2, or even one strategic vulnerability will block the attack.

In fact, the Client LAN lies outside the attack paths, so we can safely disregard that part of the network.



CAULDRON provides recommendations, e.g., first-layer defense (3 vulnerabilities) or minimum-effort defense (one vulnerability).

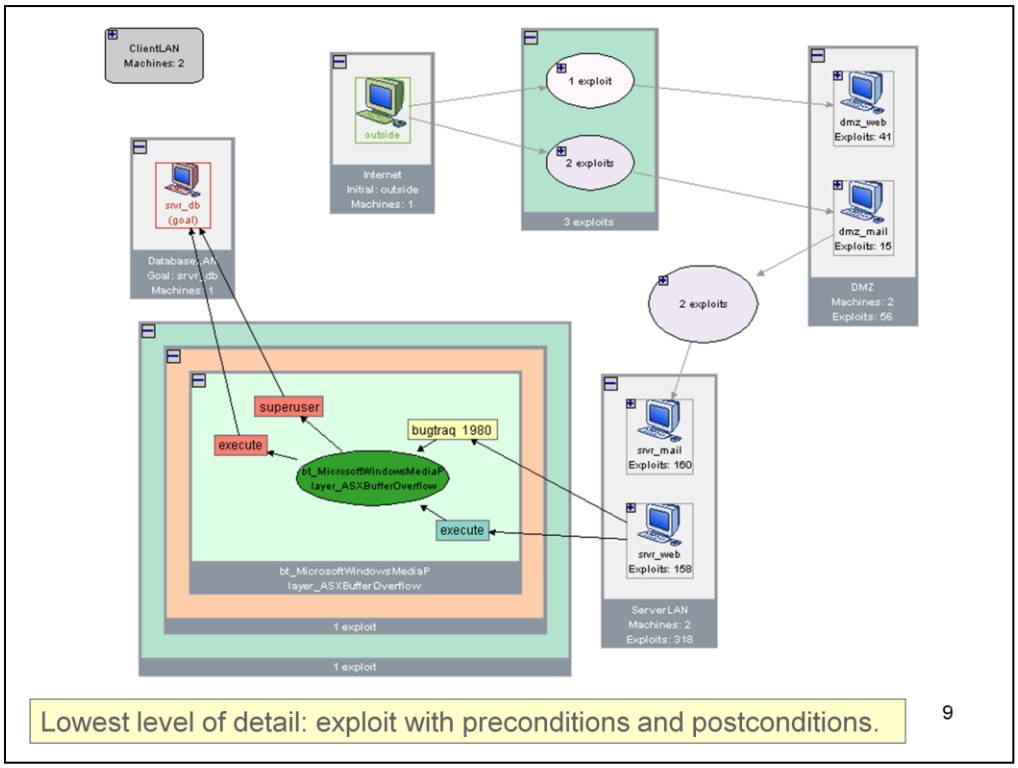
Or, since CAULDRON retains the full attack details, you can perform more detailed analysis by drilling down as desired.



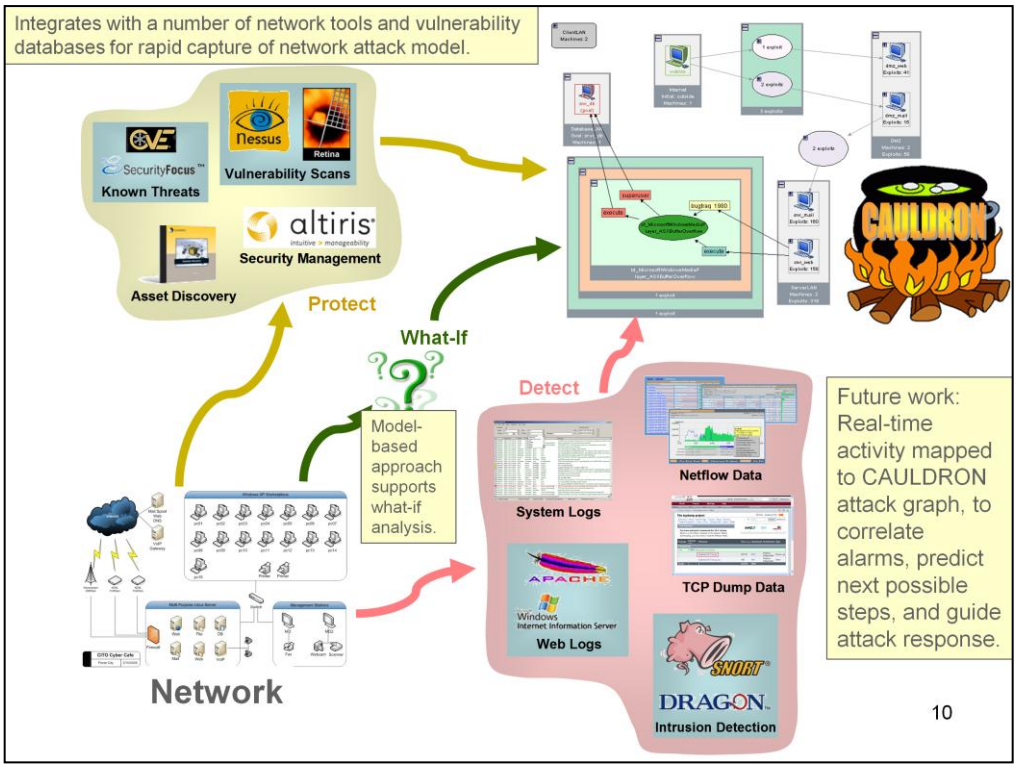
For example, this shows 2 independent attack paths from the internet into the DMZ.

Within the DMZ, there are 15 different ways to attack the mail server (from anywhere in the DMZ), and 41 different ways to attack the web server.

The attack proceeds from the DMZ mail server to the Server LAN mail server, from there to the Server LAN web server, to the DB server.



You can drilldown to the lowest level of detail, i.e., a single exploit in terms of its input and output dependencies.

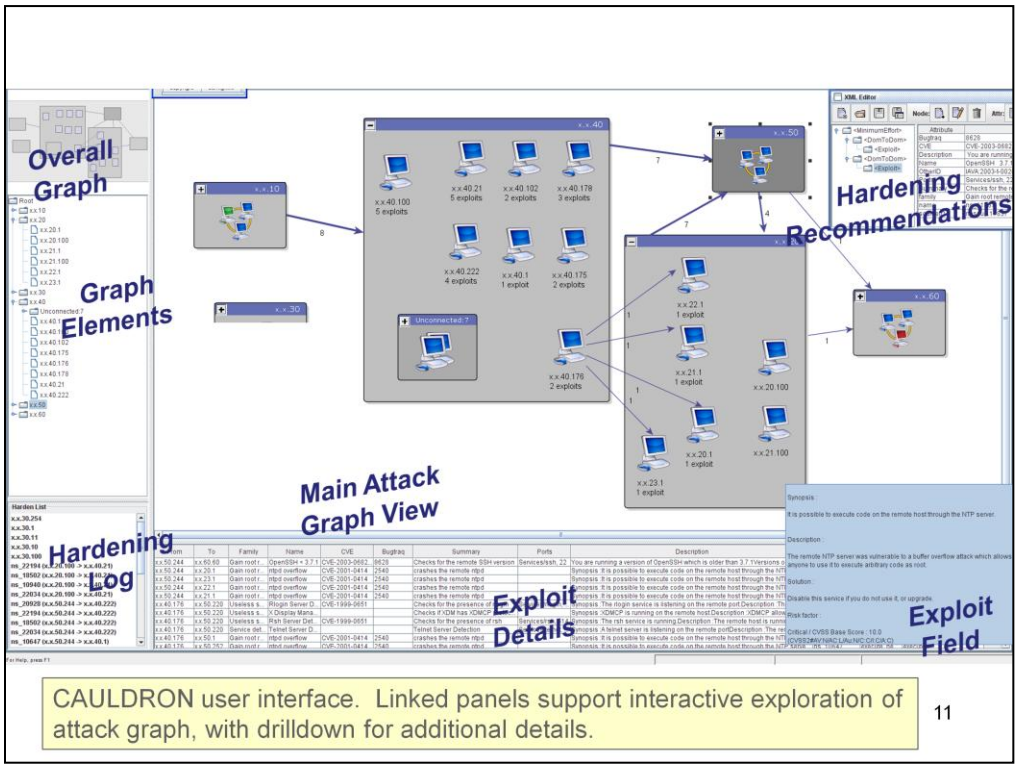


CAULDRON integrates with a number of popular tools (vulnerability scanners, security management tools, host asset discovery tools, and databases of known threats and vulnerabilities) for building its predictive model of all possible attack paths.

This allows you to proactively protect the network.

Then, real-time network activity can be mapped to CAULDRON's predictive model (future work). With this context, CAULDRON could reduce false alarms, correlate alarms, and give recommendations for attack response and forensics.

CAULDRON's model-based approach supports inexpensive what-if analysis of proposed network changes.

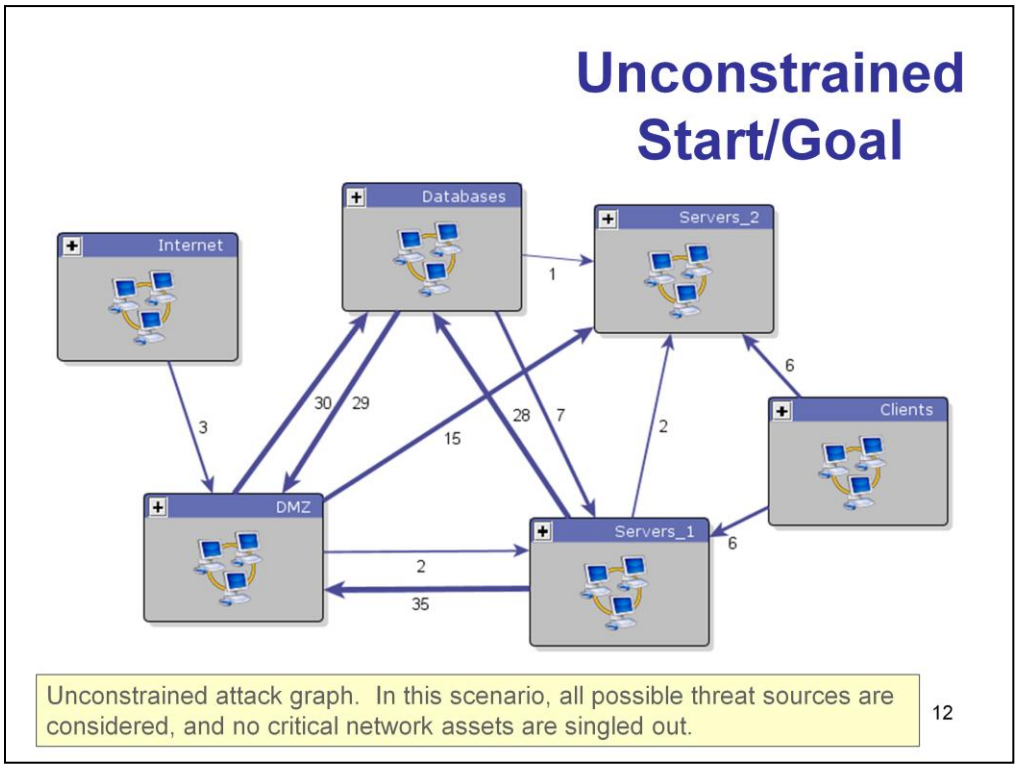


CAULDRON user interface. Linked panels support interactive exploration of attack graph, with drilldown for additional details.

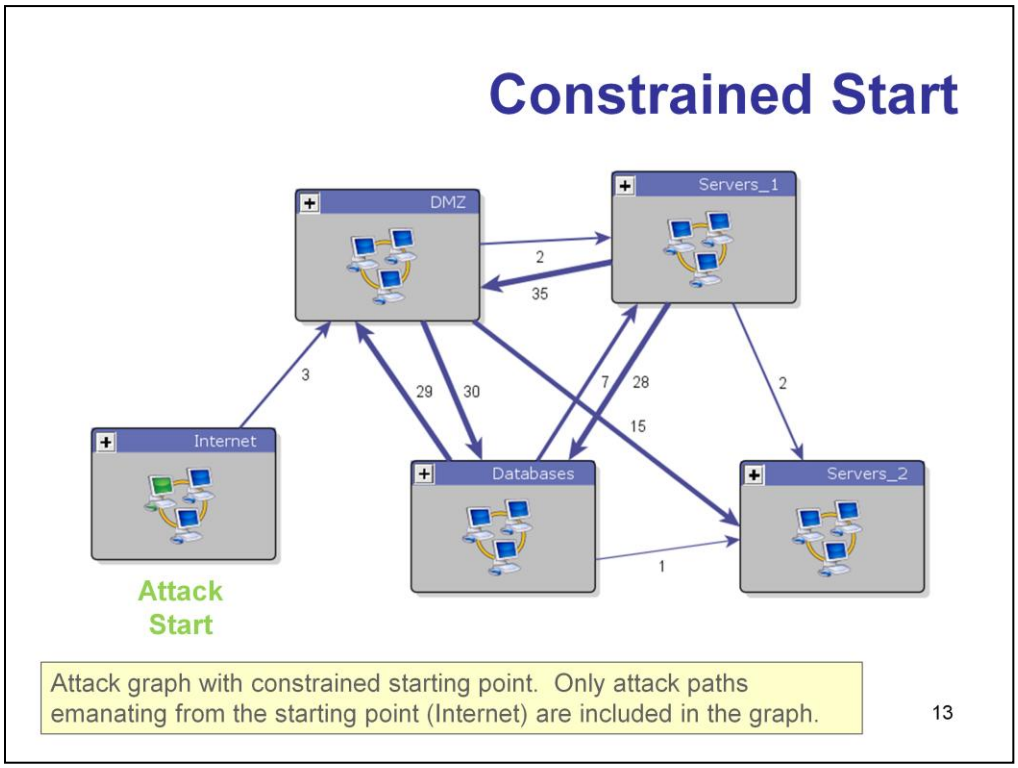
11

Through sophisticated visualization, graphs can be rolled up (aggregated) or drilled down (expanded) as the graph is explored. This shows the CAULDRON visualization interface for attack graph exploration and analysis. The main view of the graph shows all possible paths through the network, based on the user-defined attack scenario. In this view, the analyst can expand or collapse graph clusters (protection domains) as desired, rearrange graph elements, and select elements for further details. Here, two domains are expanded to show their specific hosts and exploits between them.

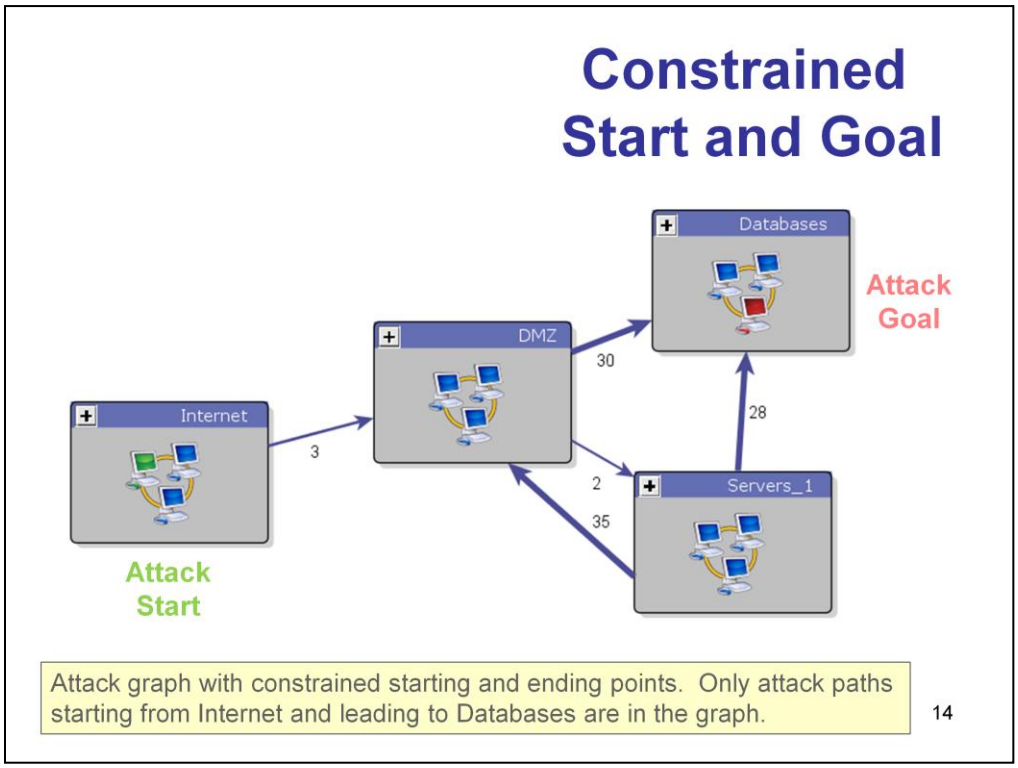
When an edge (set of exploits) is selected in the main view, details for the corresponding exploits are provided. Each exploit record contains a number of relevant fields describing the underlying vulnerability. A hierarchical (tree) directory of all attack graph elements is provided, linked to other views. A view of the entire graph is constantly maintained, providing overall context as the main view is rescaled or panned. Automated recommendations for network hardening are provided, and specific hardening actions taken are logged.



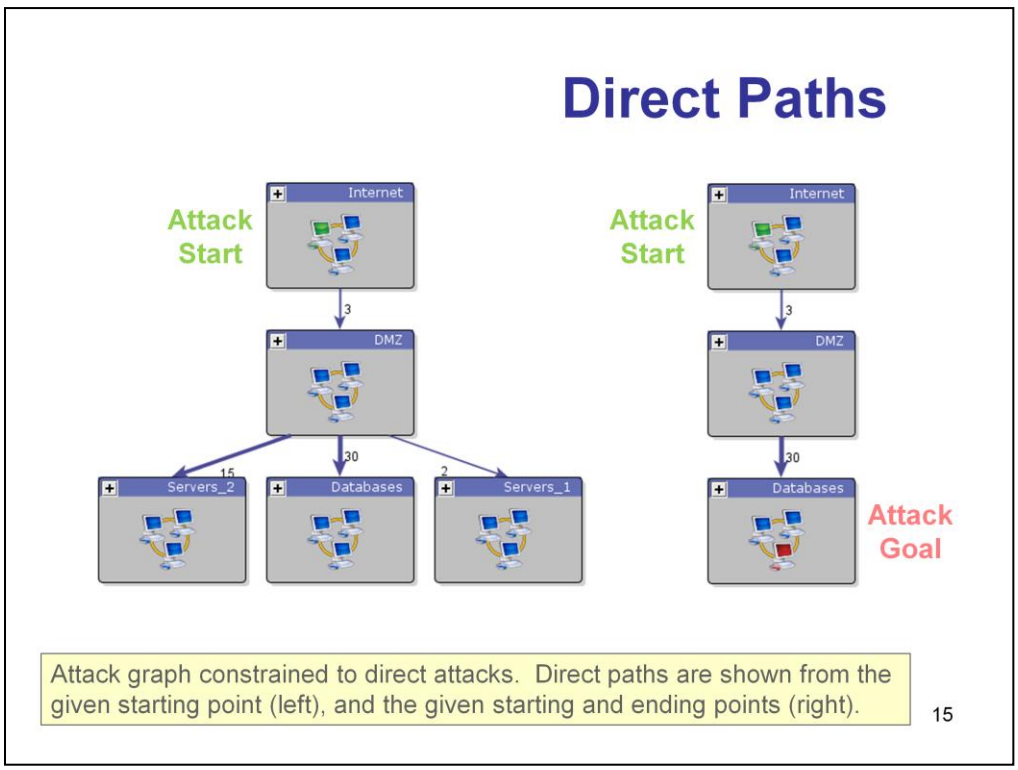
In CAULDRON, an attack graph can be completely unconstrained, i.e., all possible attack paths regardless of assumed starting and ending points in the network. In such a scenario, the source of the threat is assumed unknown and no particular critical network assets are identified as specific attack goals. This is an example of such an unconstrained attack graph.



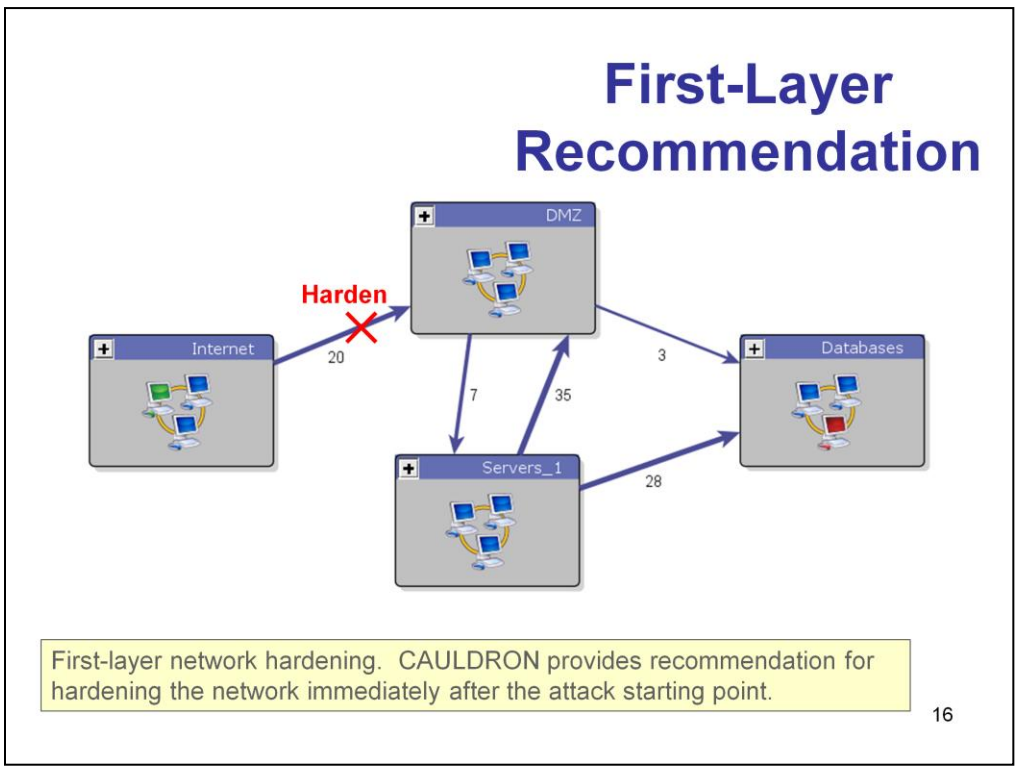
Another option is to constrain the attack graph to a given starting point (or points) for the attack. The idea is that the origin of the attack is assumed, and that only paths that can be reached from the origin are to be included. Here is an example attack graph in which the attack starting point (Internet) is specified.



Another option is to constrain the attack graph so that it ends at a given ending point (or points) serving as the attack goal. The idea is that certain critical network assets are to be protected, and only attack paths that reach the critical assets are to be included. This option could be exercised alone (with unconstrained starting point), or combined with a constrained starting point. This is an example of the latter, in which the both the attack starting point (Internet) and attack ending point (Databases) are specified.

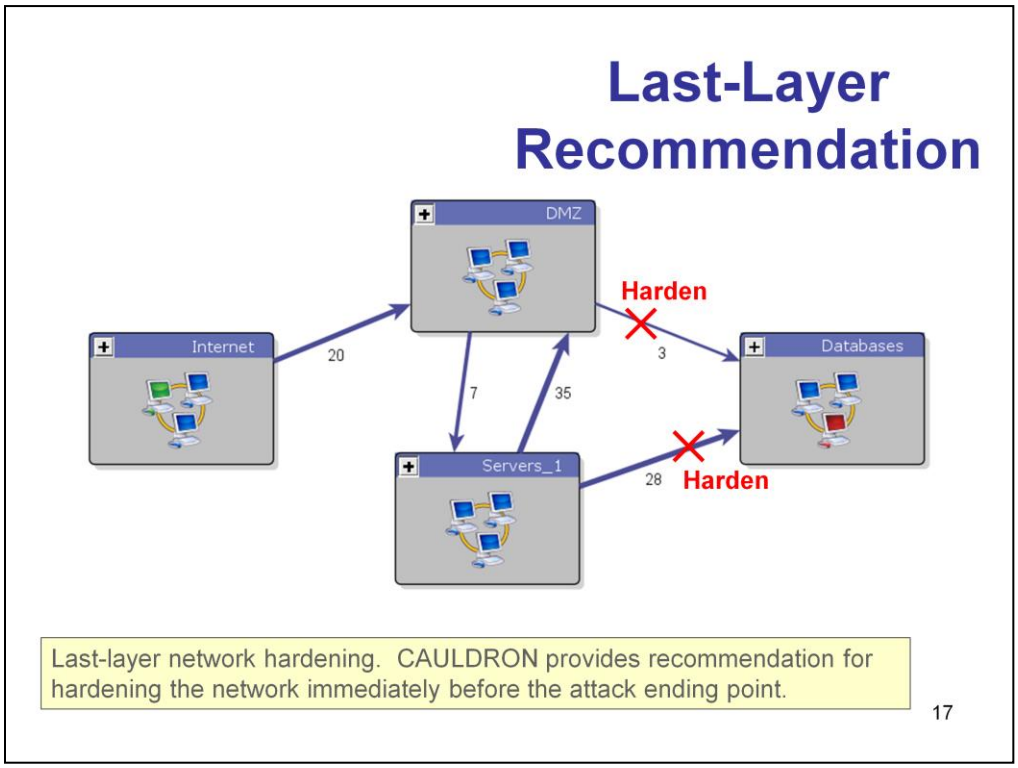


Important attack paths to consider are the most direct ones, i.e., leading from the attack start or leading to the attack goal. This is shown here. There are two scenarios considered. In the scenario on the left, the attack starting point is assumed, and the graph consists of all direct paths from the starting point. In the right, both the attack starting point and goal points are assumed, and the graph consists of all direct paths from the starting point to the goal point.



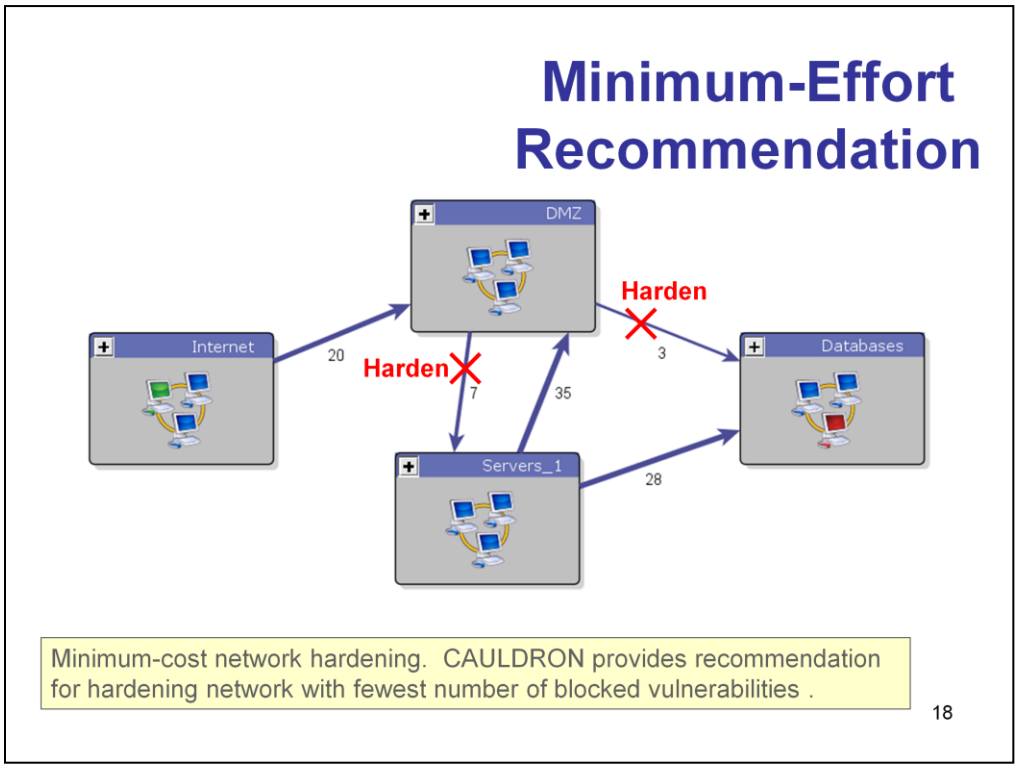
One kind of recommendation is to harden the network at the attack source, i.e., the first layer of defense. This option prevents all further attack penetration beyond the source.

We use the same attack scenario (starting and ending points) as before. However, the network configuration model is changed slightly, with a resulting change in the attack graph (numbers of exploits between protection domains). For first-layer defense for this network configuration, the recommendation is to block the 20 exploits from Internet to DMZ.



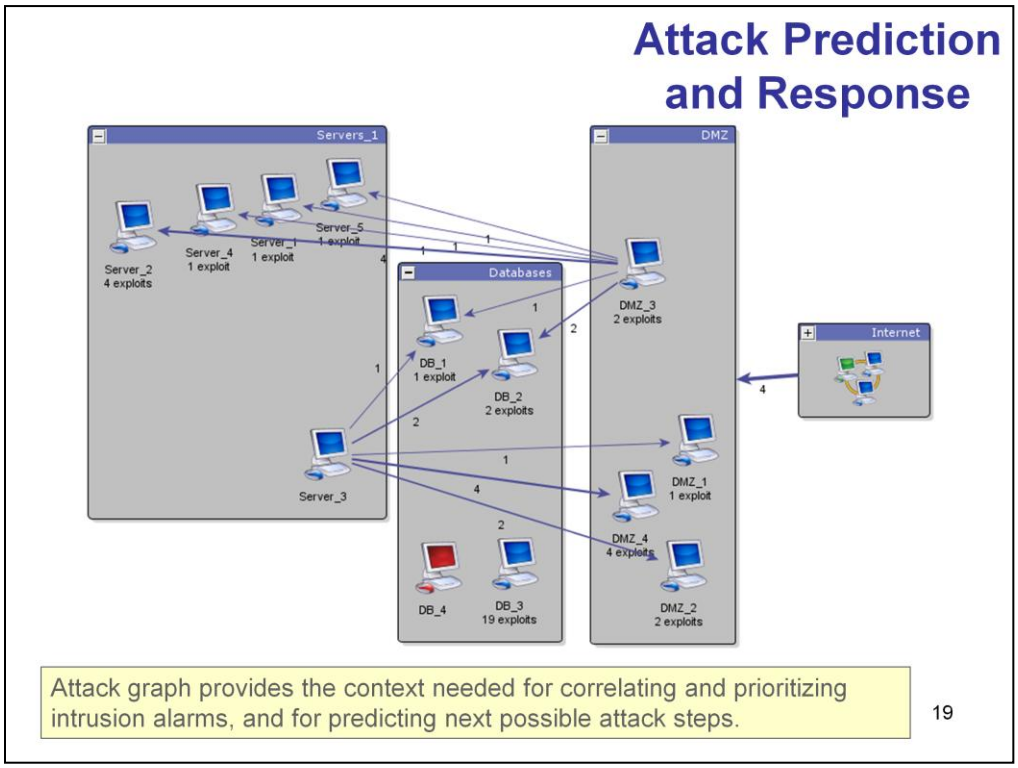
This shows a different kind of recommendation for network hardening, i.e., hardening the network at the attack goal at the last layer of defense. This option protects the attack goal (critical network resource) from all sources of attack, regardless of their origin.

For last-layer defense, the recommendation is to block the 3 exploits from DMZ to Databases plus the 28 exploits from Servers_1 to Databases, for a total of 31 exploits.



Another kind of recommendation is to find the minimum number of blocked exploits that breaks the paths from attack start to attack goal. In other words, we seek to break the graph into two components that separate start from goal, minimizing the total number of blocked exploits.

For the minimum-cost defense, the recommendation is to block the 3 exploits from DMZ to Databases plus the 7 exploits from DMZ to Servers_1, for a total of 10 exploits. This is a savings of 10 blocked exploits in comparison to first-layer hardening, and a savings of 21 blocked exploits in comparison to last-layer hardening.



Predictive capabilities of CAULDRON support alarm correlation and attack prediction. Seemingly isolated events may in fact be incremental network penetration. This also helps predict missed events.

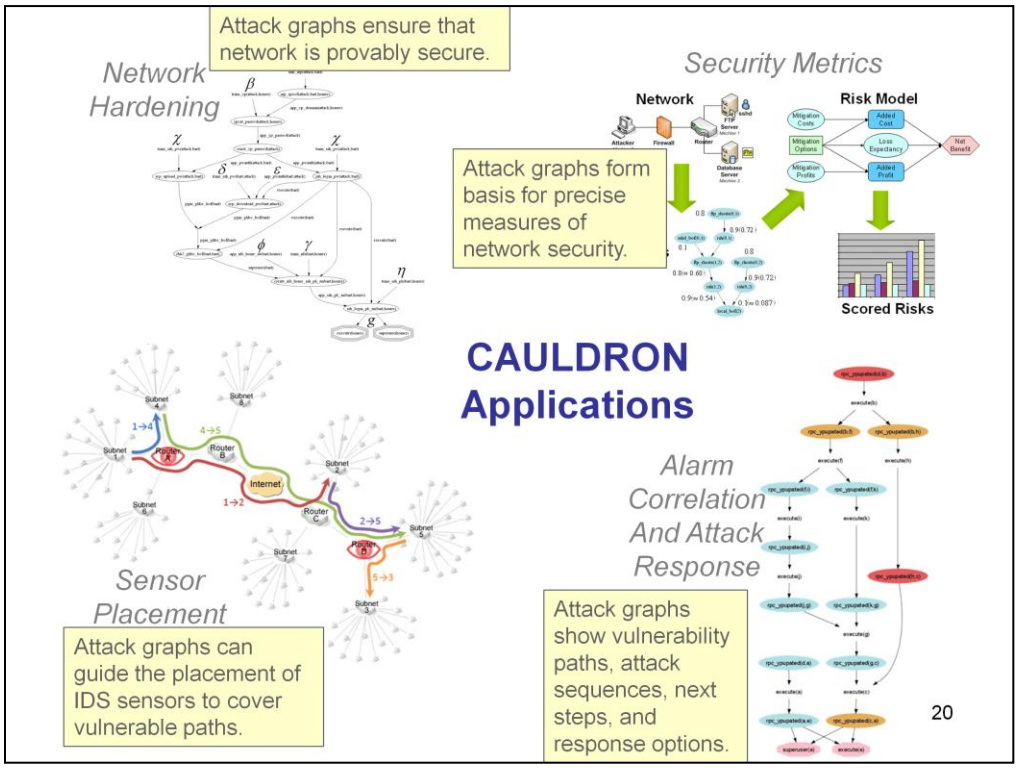
Here, attack graph is expanded to show details. Suppose alarm is raised within the DMZ (e.g., DMZ_1 to DMZ_2). But an alarm is raised from Internet into DMZ, followed by an alarm within the DMZ, it is a much stronger indicator that the attack may be a real security breach, and is given higher priority.

As a response, we block traffic from DMZ_3 to DB_1 and DB_2. This is limited to their vulnerable ports only, so other services remain unblocked. We could keep traffic from DMZ_3 into Servers_1 machines unblocked, since those machines are one less attack step from critical machine DB_4. We could wait to see if an alarm is raised from the DMZ into the Servers_1, at which point we block the vulnerable baths from Servers_1 to Databases. More aggressive response: block all outgoing traffic from DMZ to vulnerable services in Servers_1 and Databases.

The alarm in the DMZ could be follow-on from a missed intrusion from the Internet into the DMZ. This could guide further investigation into traffic logs into the DMZ, looking for missed attacks, especially against the four vulnerable paths into the DMZ.

If an attack was detected within Servers_1 (e.g., from Server_1 to Server_2), we could block traffic from Server_3 to vulnerable ports on DB_1 and DB_2. But blocking traffic from Server_3 into the DMZ is less indicated, because it is leading away from the critical Databases domain. Similarly, any alerts from Server_3 into the DMZ are lower priority, especially if they are not against vulnerable DMZ services.

CAULDRON gives a range of reasonable responses, ranked by severity or actual likelihood of attack. Severity is in terms of critical vulnerability paths, close to critical assets, and likelihood is increased by causal correlation of alerts. Multiple options are available that allow us to fine tune responses as potential attacks unfold.



For network hardening, CAULDRON provides a proof of the security of your network. This gives your customers great confidence in your actions.

Attack graphs can provide measures of overall network security, which you can monitor over time.

Attack graphs can guide the placement of sensors to monitor all vulnerable paths through your network.

Attack graphs can place alarms within the context of known vulnerability paths, so you can see attack sequences, predict next steps, and take appropriate and precise response.

Summary of CAULDRON

- Automated analysis of all possible attack paths through a network
  - Resulting attack "roadmap" provides context for optimal defenses
  - Transforms volumes of isolated facts into manageable, actionable results
- Integrates with existing tools for capturing network configuration
- Your network is provably secure, with minimum effort
- **Best tool for making informed decisions about network security**

Further Information:

Sushil Jajodia
jajodia@gmu.edu
(703) 993-1653
http://csis.gmu.edu/

Steven Noel
snoel@gmu.edu
(703) 993-2029
http://csis.gmu.edu/noel

22