black hat USA 2021
August 4-5, 2021
ARSENAL

Automated Attack Path Planning and Validation

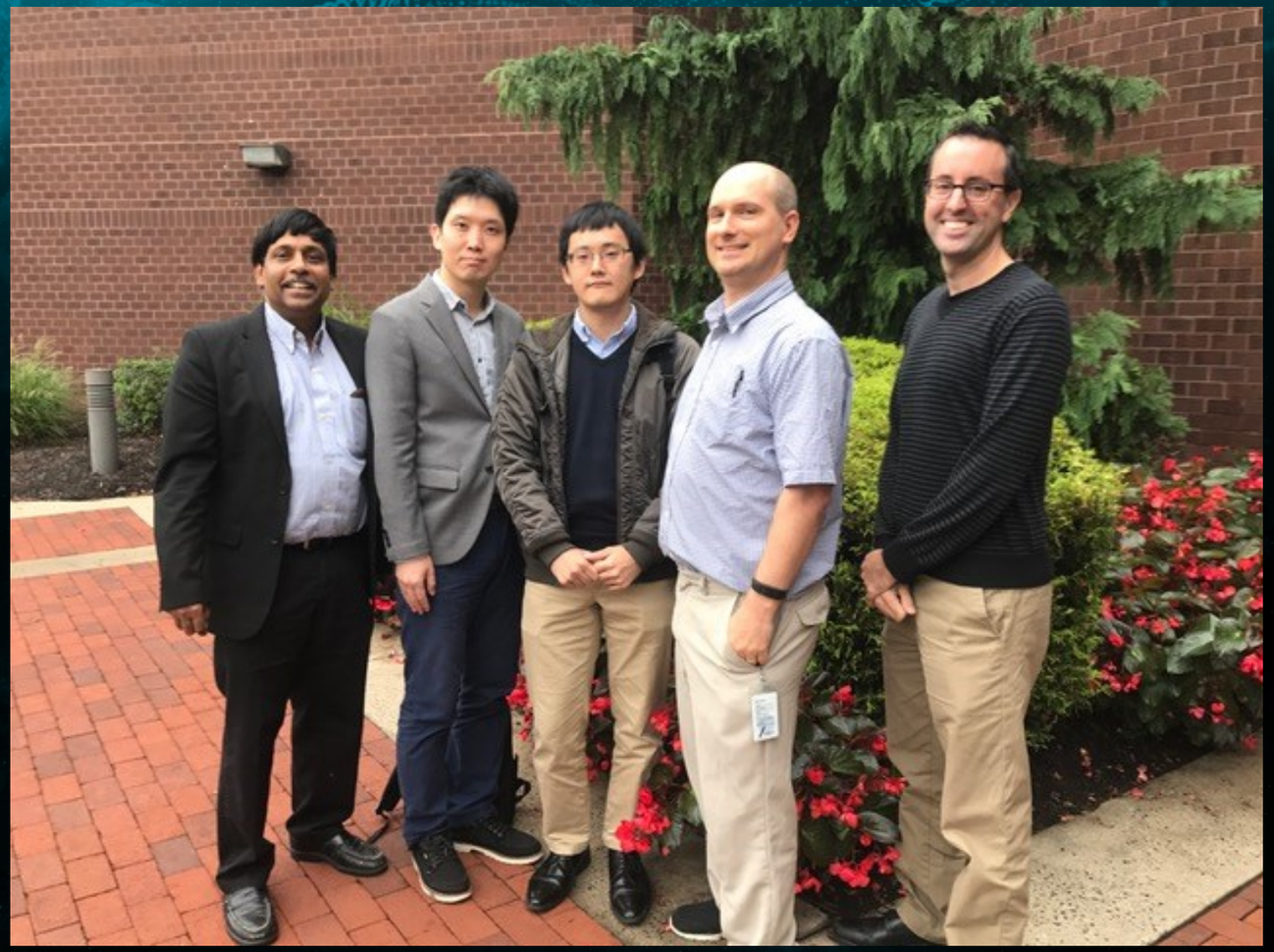
Michael Hylkema
Jason Youzwak
Peraton Labs

Fukutomo Nakanishi
Satoshi Aoki
Toshiba

https://github.com/pentest-a2p2v

Who we are:

Joint Research Team:
Toshiba (Japan) and
Peraton Labs (USA)



Background

The number of cyber attacks targeting IT and industrial systems is increasing, for example:

- Colonial pipeline system disruption
- SolarWinds supply chain attack
- JBS Foods shuts down production
- And many others...

How can we best understand the vulnerabilities of a system and the risks they present?

- Assess **attack surface**
- Prioritize identified issues by looking at **intrusion steps**

Penetration Testing is key to finding attack surface and intrusion steps.

Source: U.S. Department of Defense (Sep 2020) https://www.defense.gov/observe/photo-gallery/igphoto/2002499155/
The appearance of U.S. Department of Defense (DoD) visual information does not imply or constitute DoD endorsement.

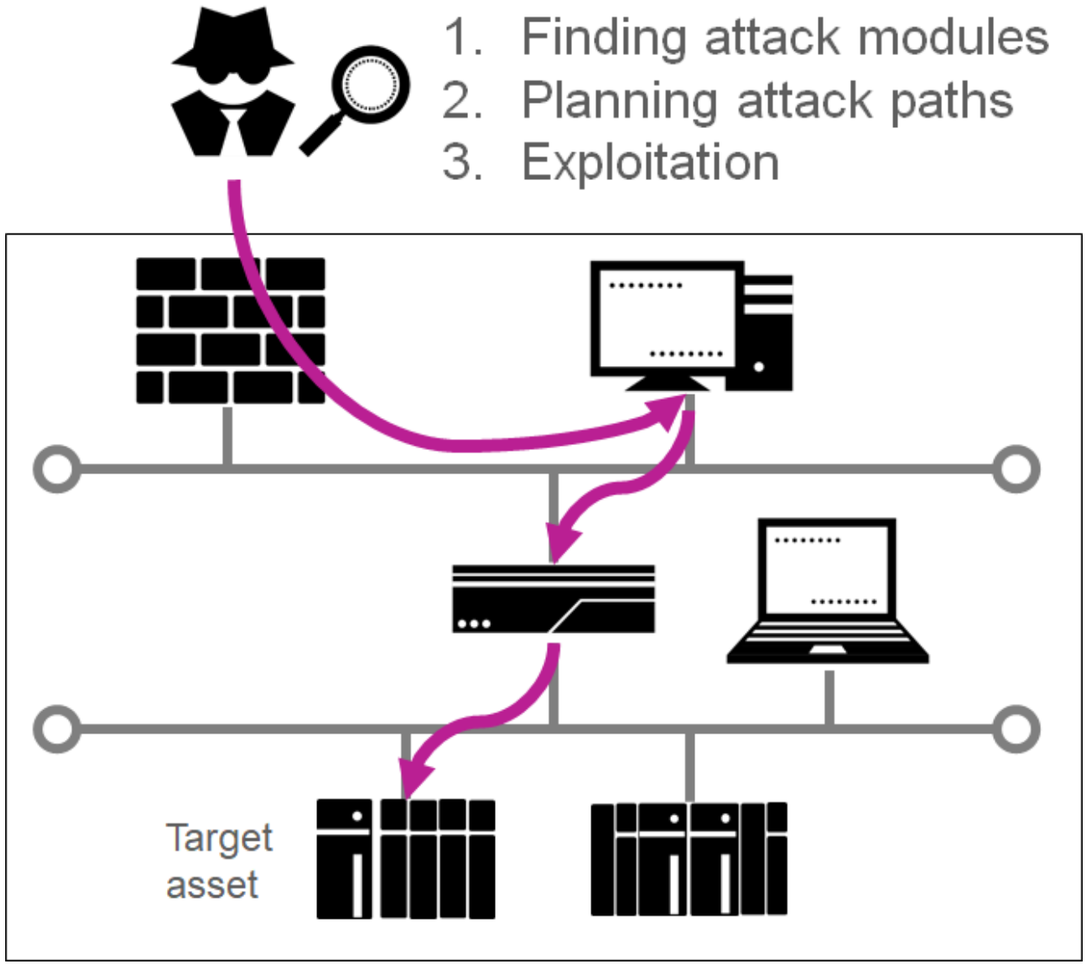
Motivation

Challenges in Penetration Testing

1. Proficiency in attack techniques
   - Evolving techniques
   - Finding appropriate attack modules
2. Combining attacks to generate attack paths
   - Lateral movement between network segments
   - Finding paths reaching a target (goal) asset
3. Experience with Exploitation Framework (Metasploit*)

Given these challenges

Traditional Penetration Testing
requires skilled Penetration Testers



* Metasploit is a product and a trademark of Rapid7 LLC.

Our solution

Automated Attack Path Planning and Validation

A tool to help non-security experts perform penetration testing

Challenge 1: Proficiency in attack techniques

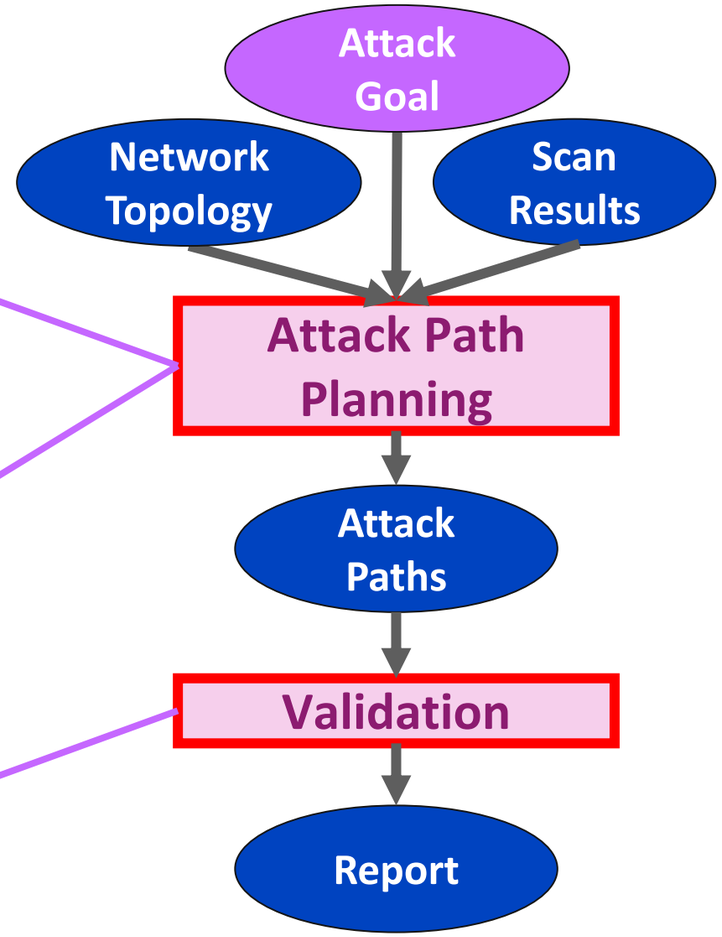
Vulnerabilities are automatically mapped to corresponding attacks

Challenge 2: Combining attacks to generate attack paths

Explore attack paths which achieve a user-supplied goal

Challenge 3: Experience with Exploitation Framework

Metasploit interactions are fully automated and executed as though by a skilled penetration tester



Key Features

Our tool was designed to be simple to use yet provide a large degree of extensibility. We designed the tool with these principals in mind:

- ✓ Automate as much as possible
- ✓ Exploit only the most likely attack vectors
- ✓ Provide flexible methods to define goal conditions
- ✓ Make it easy to add new services and support for new exploits
- ✓ Simplify input files and formats
- ✓ Keep reporting concise, but with sufficient information to validate

Attack Path Planning

Attack Path Planning: Overview

The Planning module is the brains behind the building of attack paths

It pulls together numerous pieces of information analyzed from inputs:

- the initial conditions and goals provided in a configuration file
- the network topology
- the services and vulnerabilities on each host

To build attack paths the Planning module algorithm uses modified versions of:

- A formal action modeling methodology
- A depth-first tree traversal algorithm

Attack Path Planning: Input/Output

Inputs that users need to configure

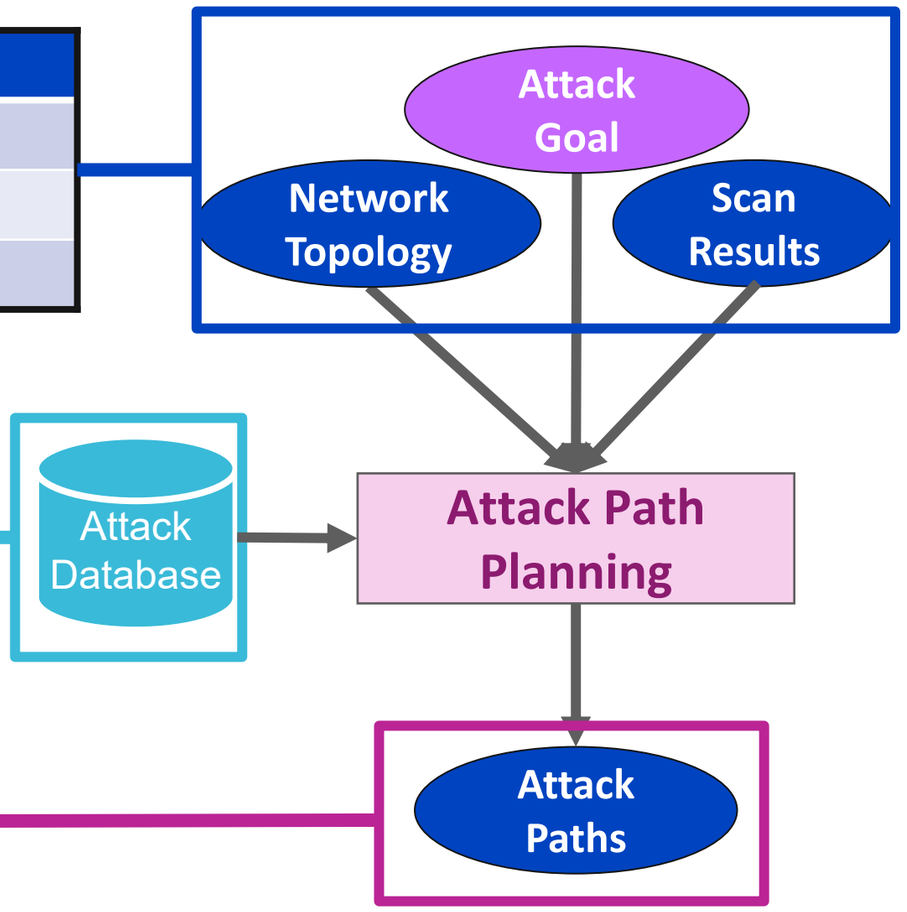
Input	Format
Initial Conditions & Attack Goals	Python ConfigParser
Network Topology	XML
Vulnerability Scan Results	XML, CSV

Inputs that users can modify to extend the tool's capability

Input	Format	Description
Attack database	XML	Definitions for attack modules

Output from the attack path planning module

Output	Format
Attack Paths	JSON



Attack Path Planning: Action Modeling

The Planning Module uses a formal action modeling methodology, with components defined as:

- ***Pre-conditions:*** conditions which must be met before a certain action can be taken on a target host
- ***Actions:*** Metasploit module and all appropriate parameters for executing the module
- ***Post-conditions:*** updates to the system state as an effect of executing the actions

Each attack module (referred to as capabilities) contains a set of these components

Example PAP Definition

Precondition

Attacker can access *$Host* : *$Port*
$Host has BlueKeep Vulnerability on port *$Port*

Action

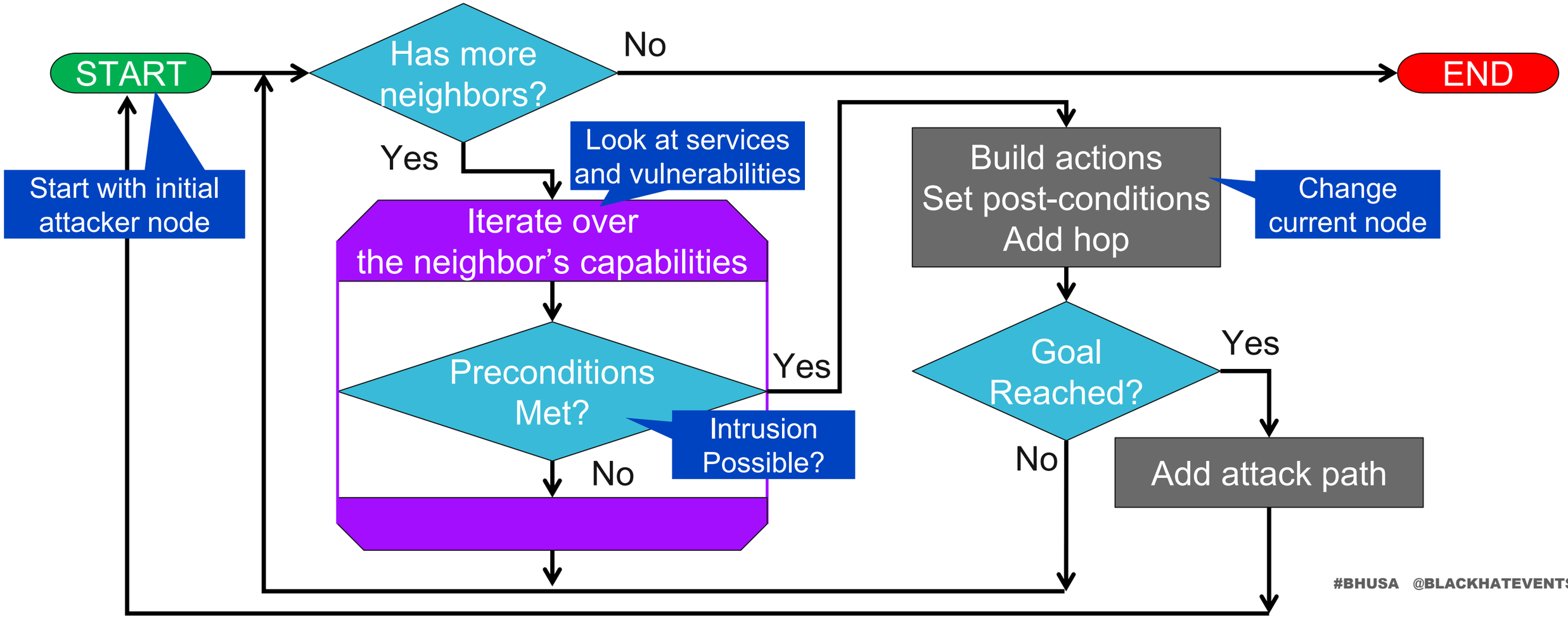
Run Metasploit/BlueKeep module
with RHOST=*$Host* PORT=*$Port*

Postcondition

Attacker has access
to *$Host*

Attack Path Planning: Graph Traversal

The Planning module uses a modified depth-first tree traversal algorithm to build attack paths



Attack Path Planning: Scoring

Once an attack path has been generated a score is computed

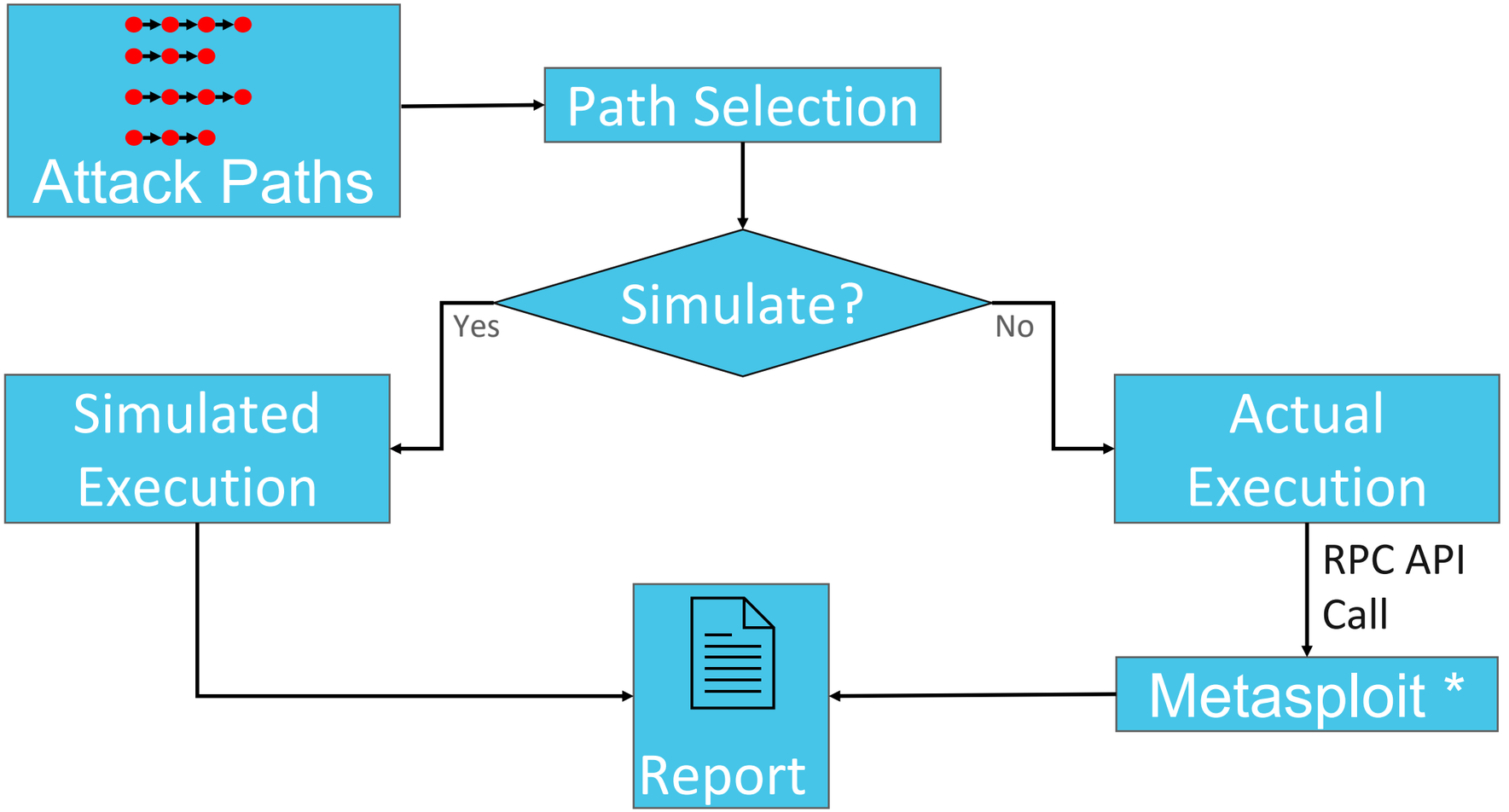
- Scoring is components based and takes into account various aspects of the attack path
- A configurable weighting is applied to each component value
- Higher scoring paths are considered more desirable
- A minimum score can be set to filter out less likely attack paths
- Attack paths are ordered by score during the selection phase of validation

Current Scoring Algorithm

```
round( 10 + ( NUM_EXPLOITS * EXPLOITS_WEIGHT ) +
           ( NUM_SERVICES * SERVICES_WEIGHT ) +
           ( PATH_LENGTH * LENGTH_WEIGHT ) +
           ( TOTAL_SEVERITY * SEVERITY_WEIGHT ), 2)
```

Validation (Execution)

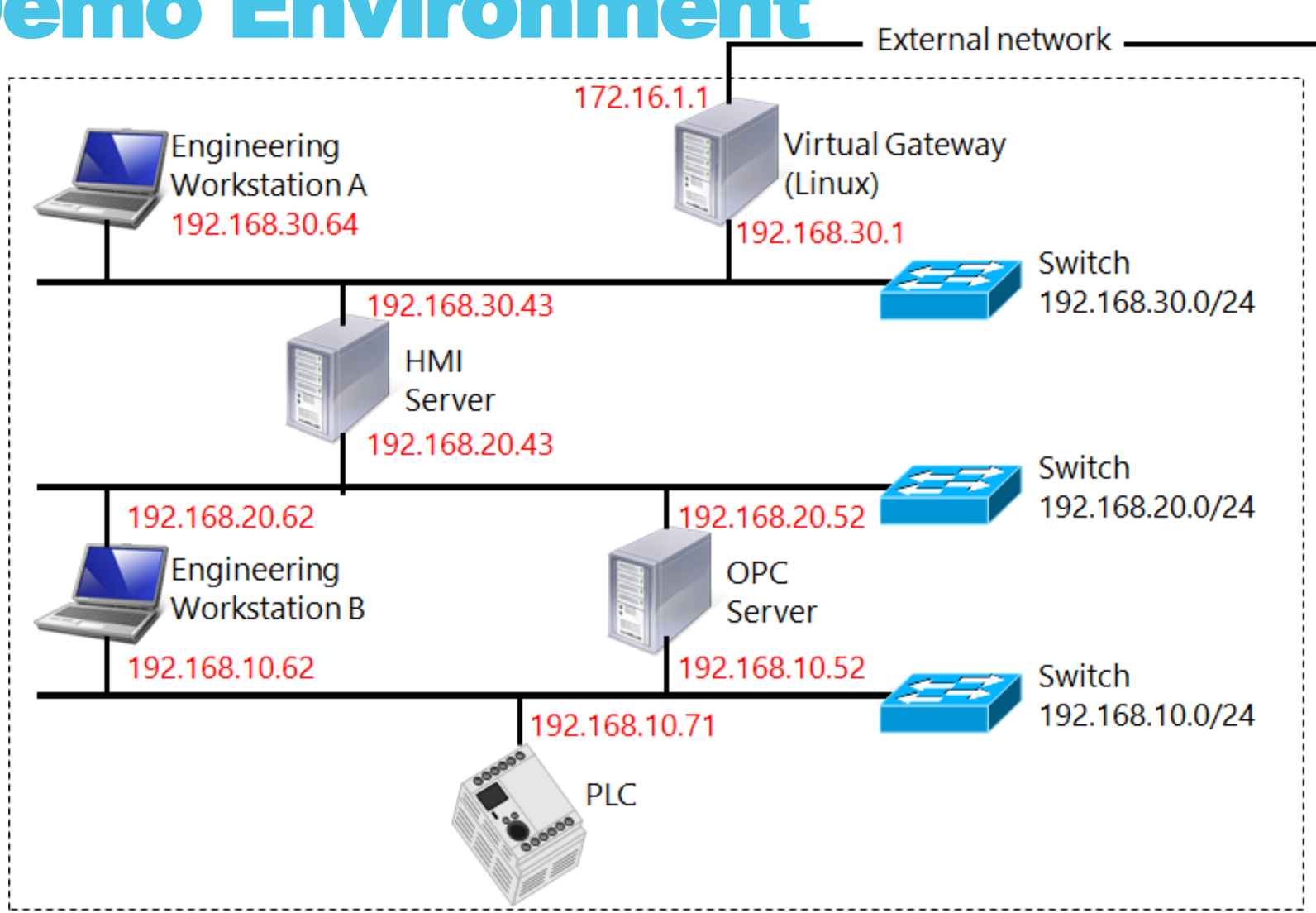
Validation: Overview



* Metasploit and its logo are trademarks of Rapid7 LLC.

Tool Demonstration

Demo Environment



Goal: Change Temperature

Host	Attack Module(s)
Gateway	SSH
Eng A, HMI, Eng B, OPC	Bluekeep, MS Spool Vuln, EternalBlue, EternalRomance
PLC	Modbus

Future Plans

- Improve and streamline user experience
- Perform exploit actions outside of Metasploit:
  - Command line tools
  - User interactive mode
  - Other exploit frameworks
- Enhance attack path planning
  - Support more exploit types and actions
  - Improve algorithm efficiency
  - Improve Scoring

Thank You!

- Please give the tool a try and provide feedback on how you're using the tool.
  - Github repository: https://github.com/pentest-a2p2v
  - Contact our team: a2p2v@peratonlabs.com