APPLIED COMMUNICATION SCIENCES

Increasing the Reliability and Productivity of Cyber Testing via Experiment Speedup






Alex Poylisher, Constantin Serban

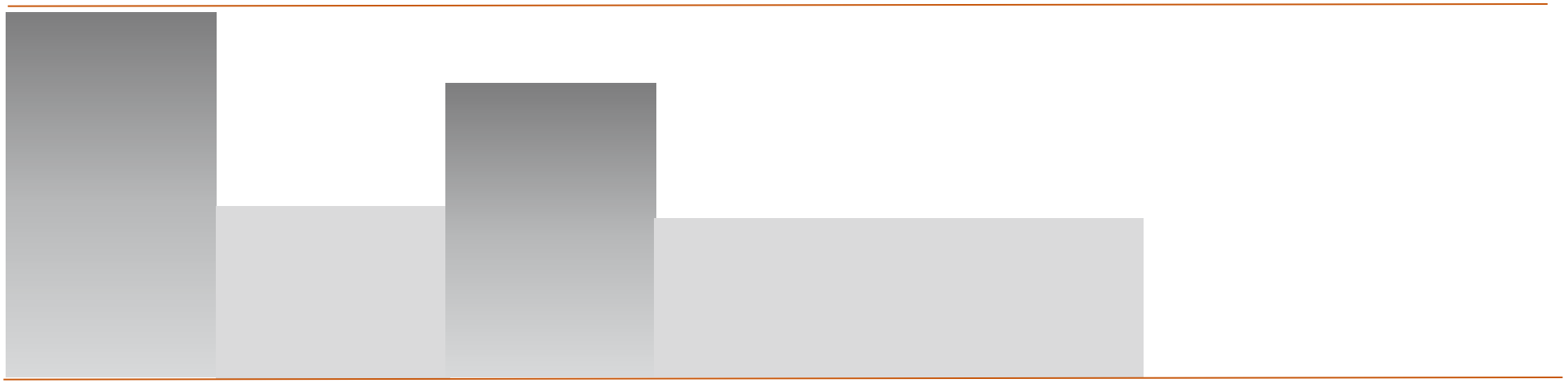
{apoylisher,cserban}@appcomsci.com

November 14, 2013

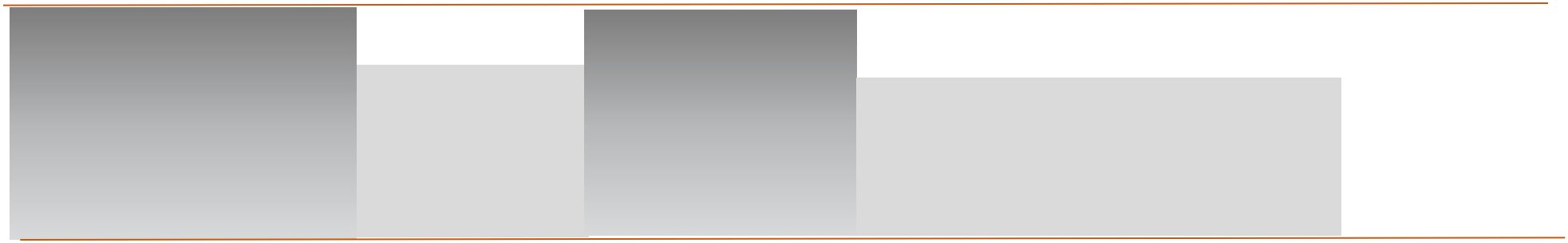
Motivation

- Cybersecurity testing requires a high fidelity representation of both host and network resources
  - Current cyber-security testbeds run full OS environments in physical or virtual hosts
  - Employ a large and diverse network, emulated, or simulated
  - DETER, NCR, CyberExata, etc…
- Both hardware resources and evaluators' time are limited
- The usual tradeoffs:
  - Limit the size of the testing activity(i.e. fewer nodes, hosts, etc.) and keep high fidelity
  - Reduce fidelity by abstracting the representation of the least important aspects of the cyber-system (if known in advance)
  - Reduce the amount of testing activity (fewer runs, attacks, scenarios)
- Can we do better than that?

A New dimension: Stretched Time

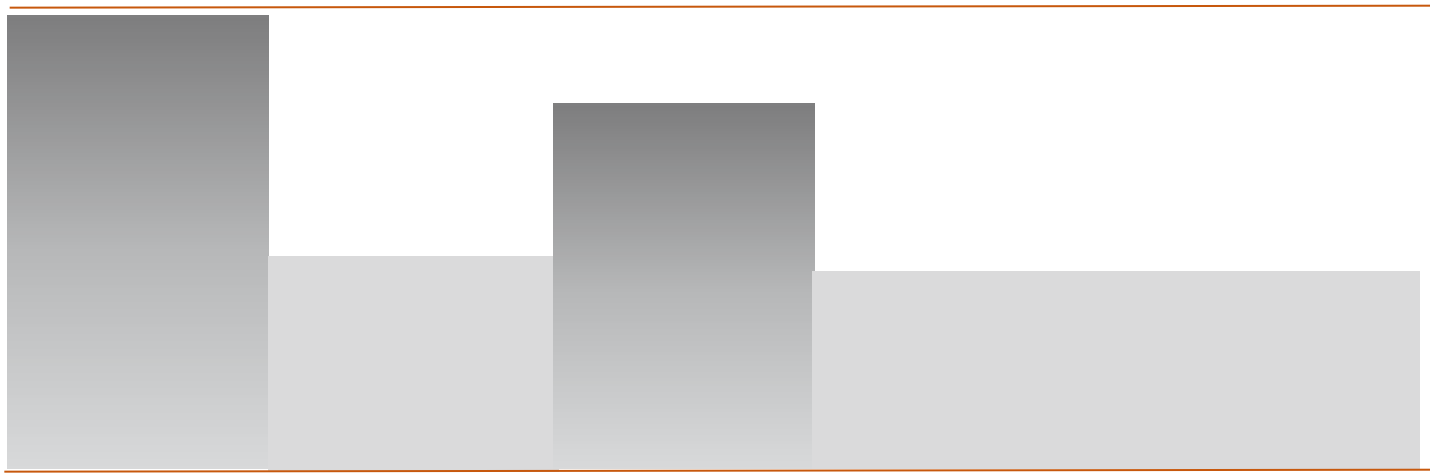


- Provision resources for peak load, unutilized most of the time

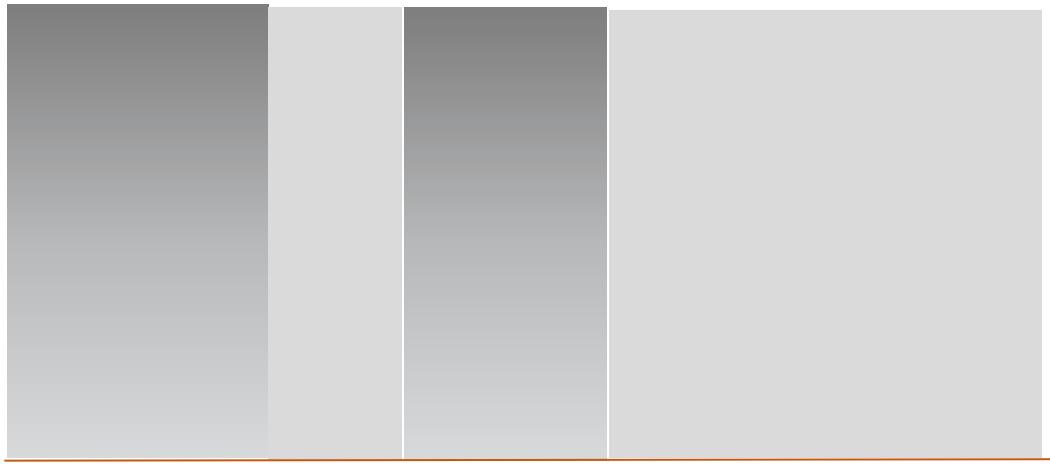


- Provision fewer resources, stretch execution time of peak loads

A New dimension: Compressed Time



- Provision resources for peak load, unutilized most of the time



- Use resources fully, compress execution time for non peak loads

TimeSync

- Stretched (slower than real time) execution of testing scenarios to increase scalability with fewer resources previously implemented in the VAN Testbed
  - See [MsWIM12], [MILCOM12]
- This work introduces the compressed time testing in cybersecurity

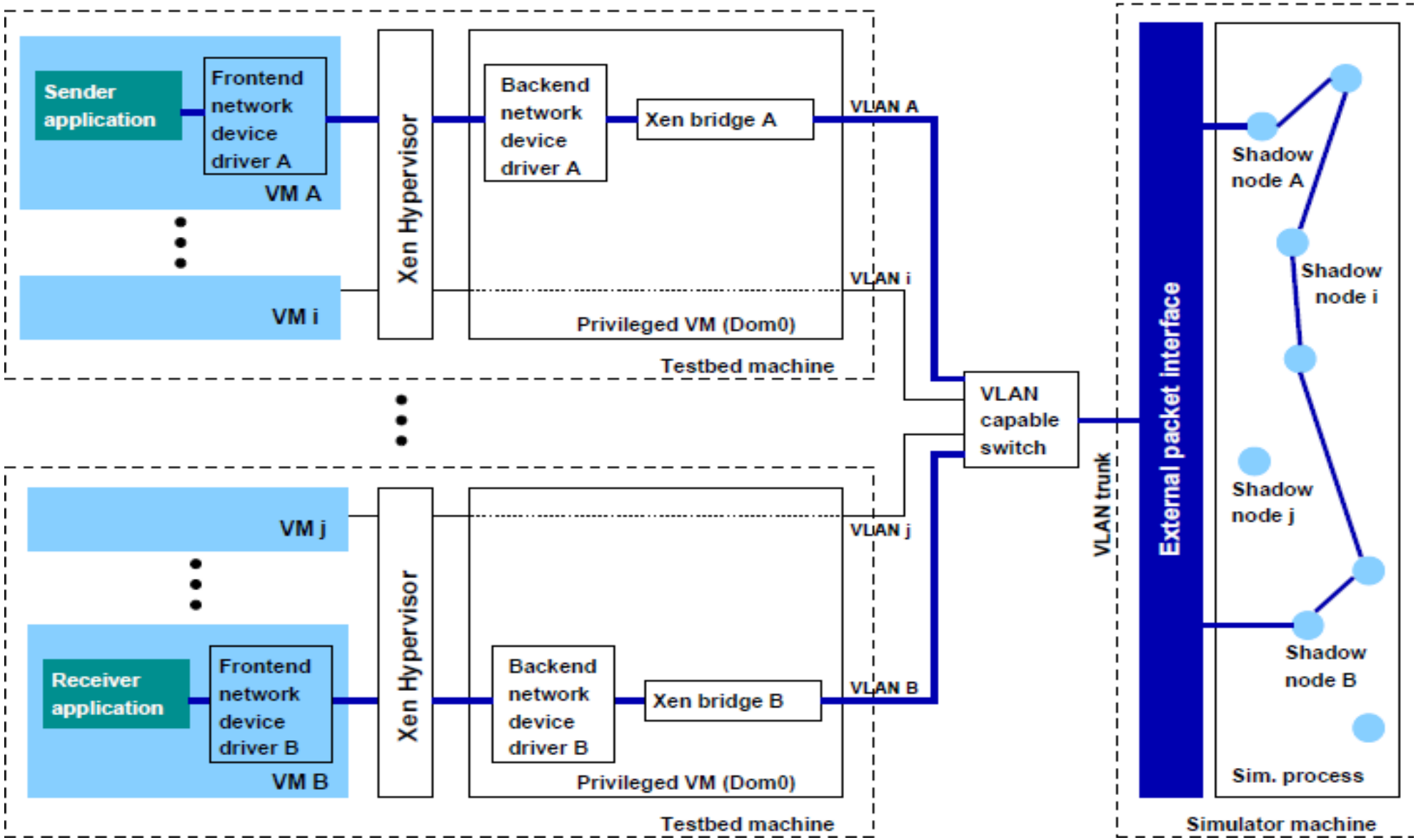
Compressed Execution in Cybersecurity

- Typical cyber activities:
  - Reconnaissance (active/passive scanning, sniffing, etc)
  - Penetration attacks
  - Botnet setup and maintenance
  - Information exfiltration
  - Denial of Service/Disruption
- Most of the above activity are either:
  - Short and resource intensive
  - Long and low-key (i.e. low resource usage)
  - Long periods of time where no activity occurs
- Testbed resources are significantly underutilized
- Can benefit greatly from compressed execution
  - Can get more work done
  - More evaluations, more runs, diverse scenarios, etc.
  - Better confidence in results

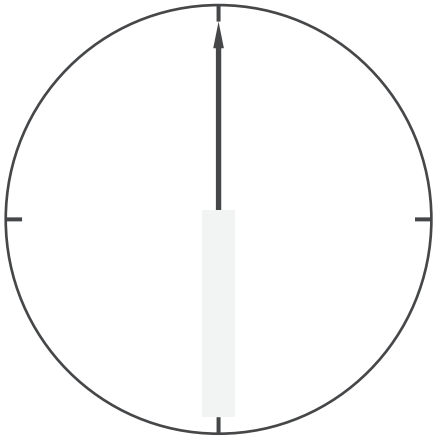
Compressed Time Execution in Practice

- Preserving high fidelity requires no change to applications and their ecosystem (i.e. libraries, OS, network, etc.)
- Change the source of time at the lowest level practically available
- Virtual hardware amenable to such manipulation
- Hence alter the time perception of entire virtual machines from the hypervisor.
- Implemented in Xen hypervisor without any VM modifications

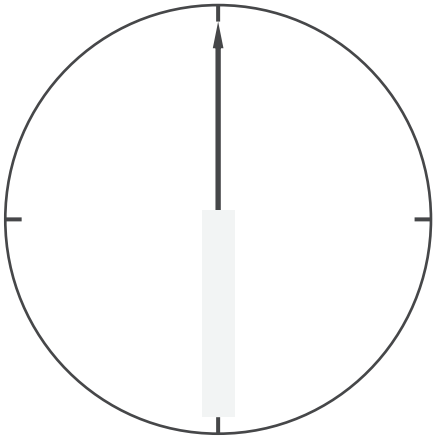
VAN Testbed: Our Cyber Testbed



The Concept of Testing Speedup



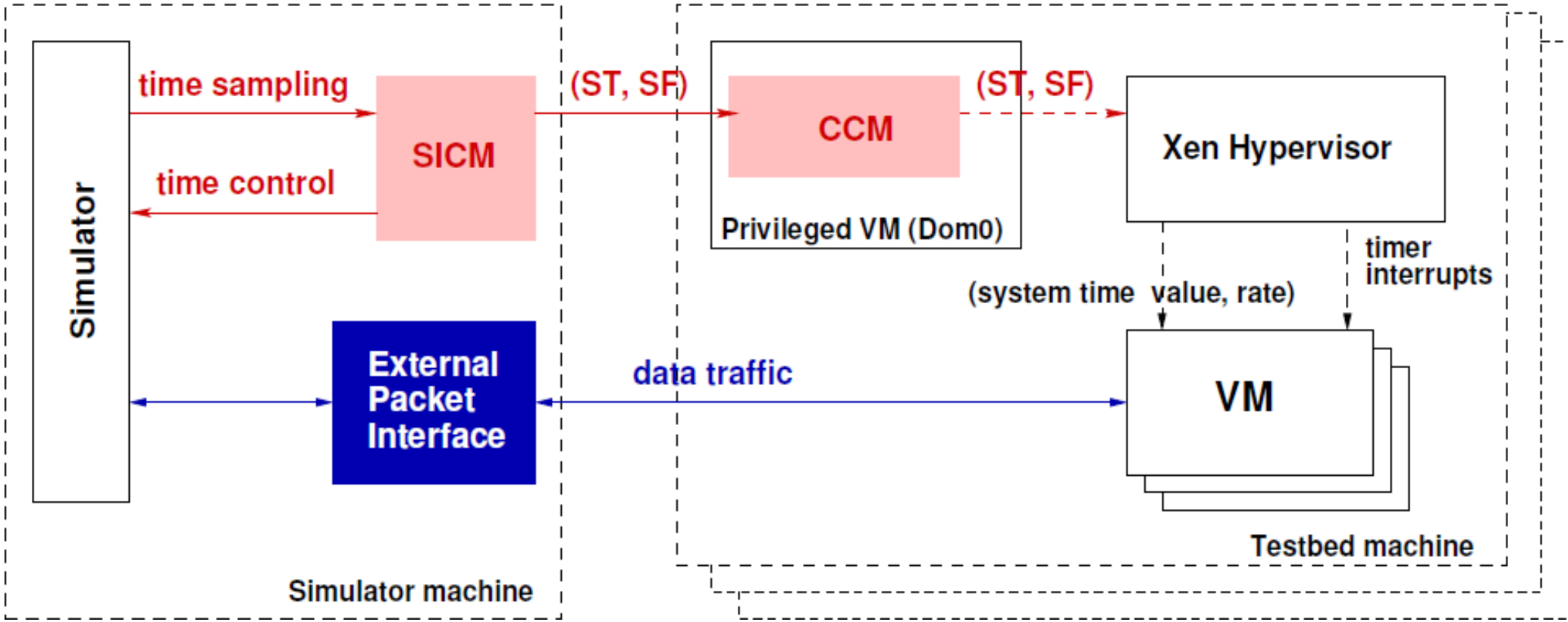
Real Time



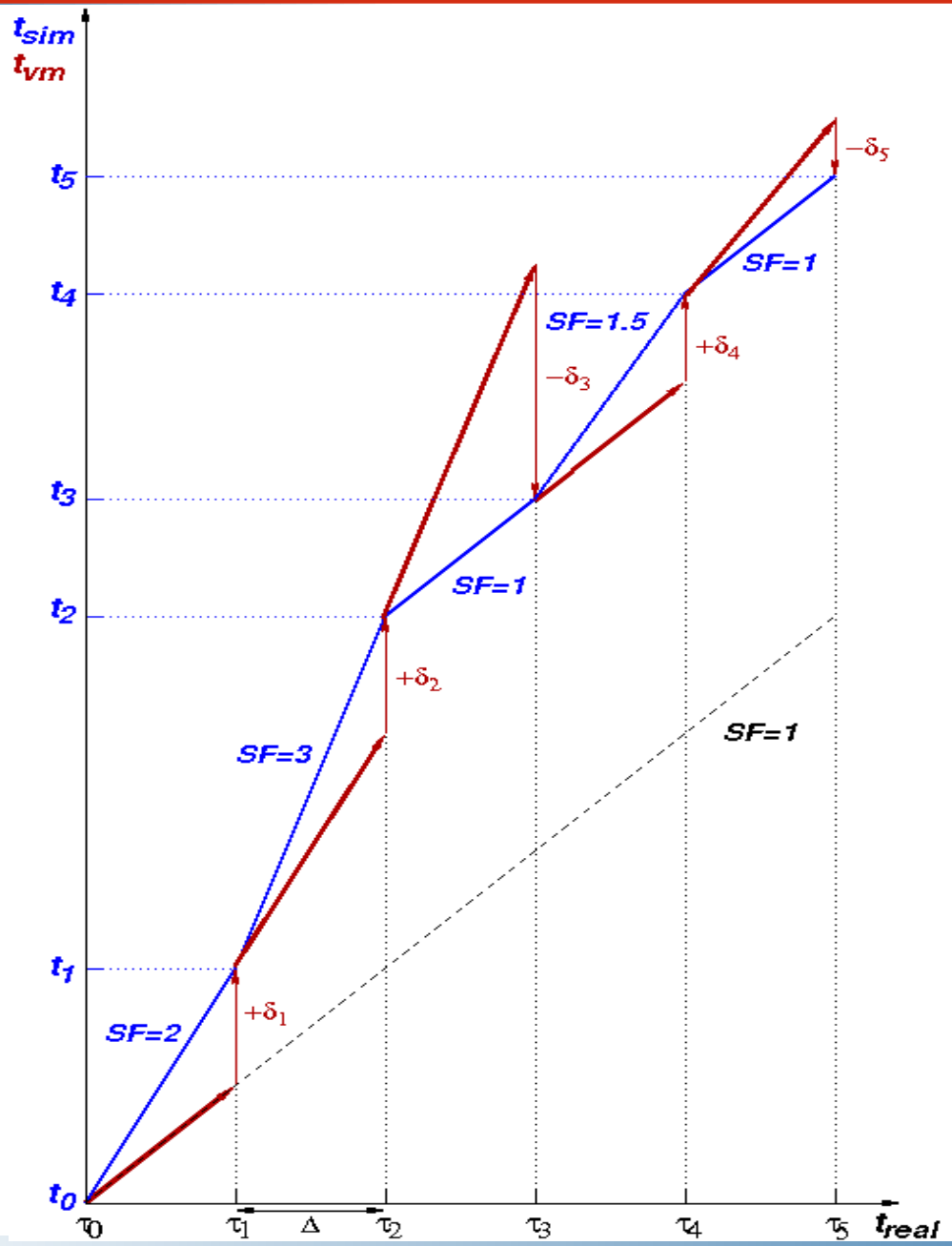
Compressed Time

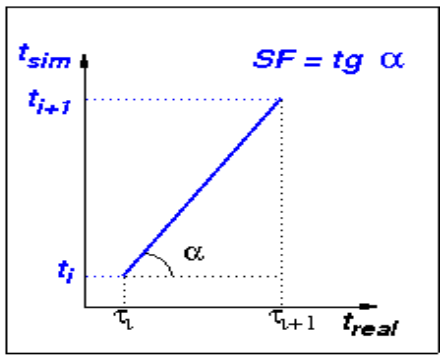
- Software under test will perceive the time passing faster than it actually does in real time.
- The rate of compressed time advancement (i.e. the speedup factor) can be constant or variable

TimeSync Architecture



VM time tracking external time source





Experiments

- Purpose:
  - Assess the fidelity of cybertesting under speedup
- Metrics:
  - Functional correctness
  - Total execution time of typical cyber activity
- Representative activities:
  - nmap reconnaissance
  - metasploit attack

NMAP Experiments

Why NMAP :

- Widely used tool in penetration testing and vulnerability discovery
- It employs a variety of methods to profile a network, or set of hosts, and discover information an attacker may later exploit
- Complex tool that performs a variety of tasks both computational and network related
- Can be viewed as a macro benchmark for our cyber-security testing

Nmap Experiment Details

Executed the folowing nmap commands over a set of FC18_64 hosts in the VAN Testbed:

nmap -sS -PN -A -T2 -vv 10.2.34.2

- TCP SYN Stealth scanning disabling PING probing, with OS and version detection, with polite timing template mode (somewhat lower traffic rate), verbose output, with a single host target located across a WAN ~100ms away.

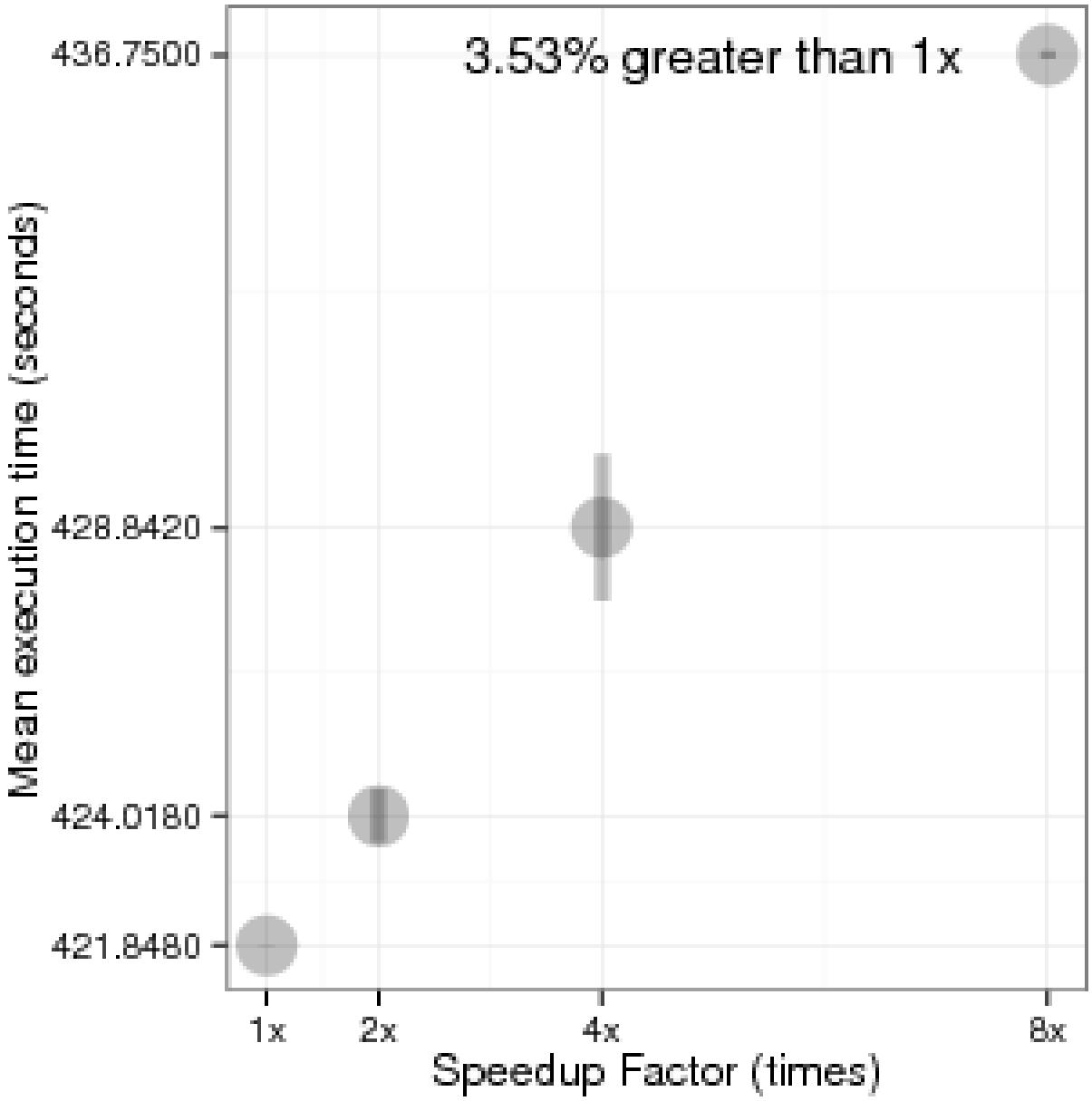
nmap -sS -PN -A -T3 -vv 10.2.34.2

- Same as above, except the timing of the request is normal (i.e higher frequency than above)

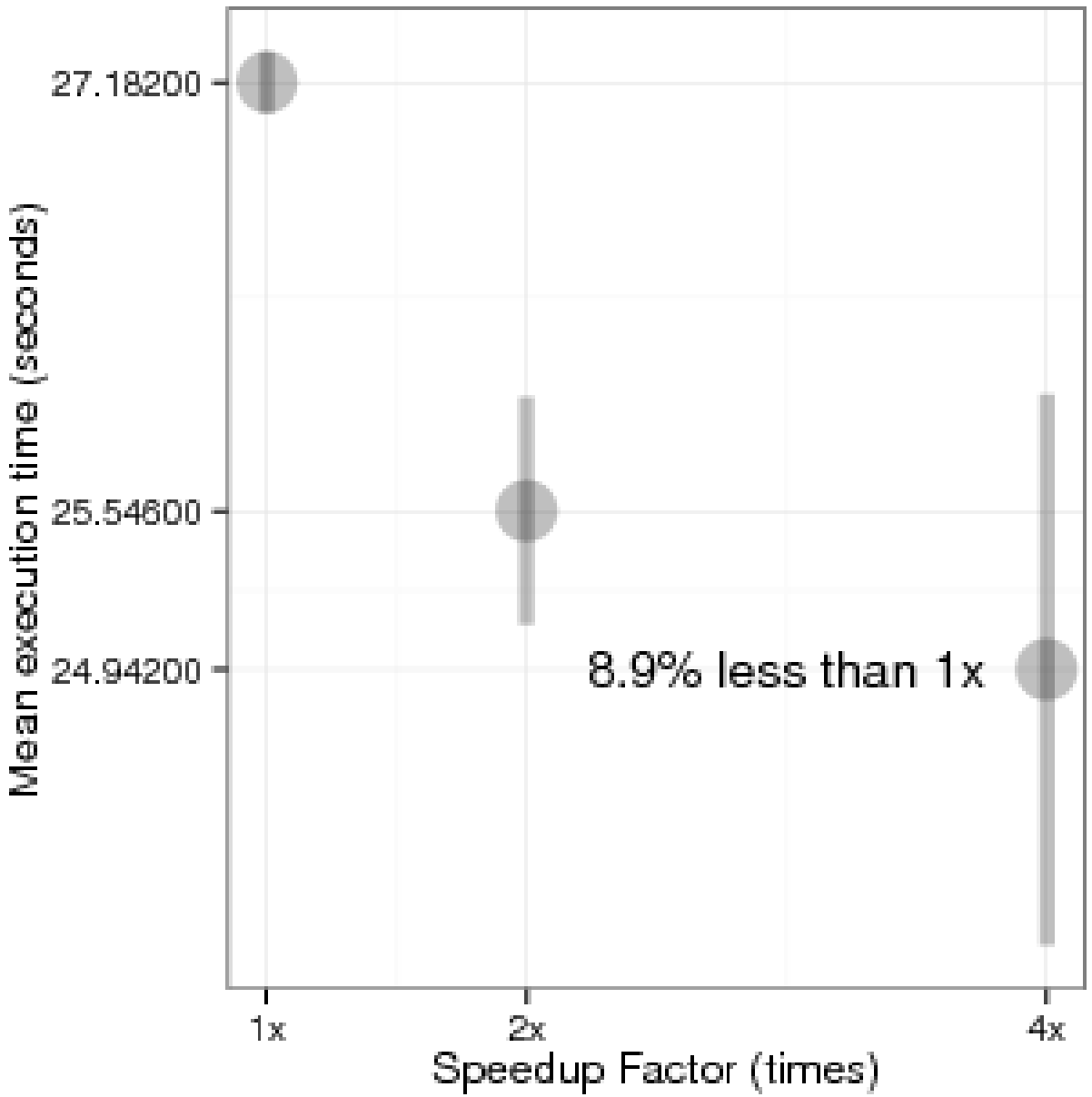
nmap -O --osscan-limit 10.2.32-48.2

- Perform an OS detection scan, but abandon host if not at least one TCP open port and one TCP closed port are found. This operation is performed on a subnet of 17 hosts.

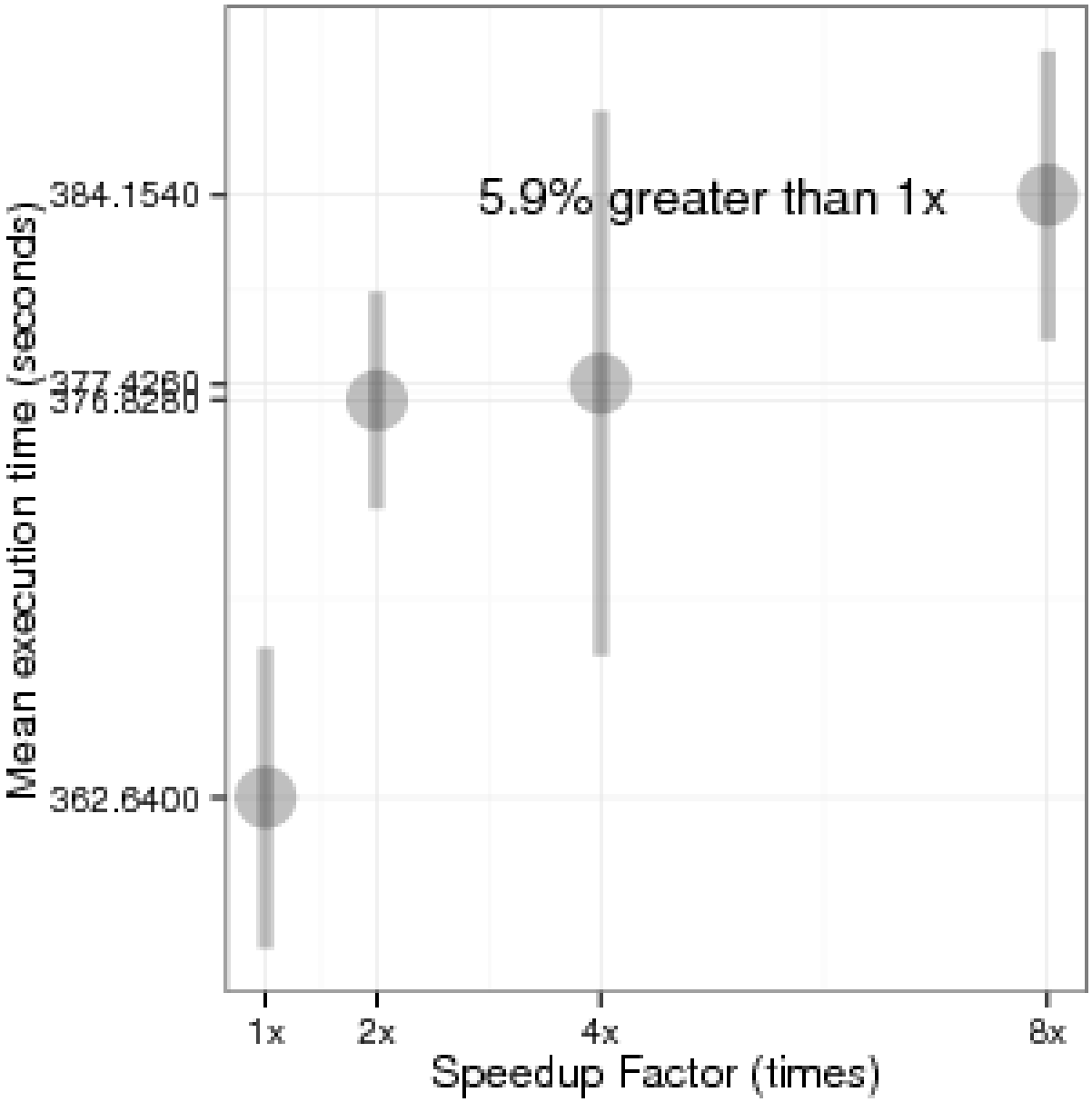
NMAP 1 Fidelity Results



NMAP 2 Fidelity Results



NMAP 3 Fidelity Results



Metasploit Experiments

Why Metasploit:

- Widely used framework with a large database of vulnerabilities and exploits for the most common operating systems and userland services.
- Used for prototyping and developing attacks / testing defenses
- Exploits are complex pieces of software consisting of sequences of execution involving bot computation and network activity
- Appropriate as a macro-benchmark for cyber-security testbeds.

Metasploit Experiment Details

```
tcpscan:
    use auxiliary/scanner/portscan/tcp
    set PORTS 1-200
    set RHOSTS 10.2.34.2
    set THREADS 1
    set INTERFACE eth0
    set TIMEOUT 4000
    set CONCURRENCY 1
    run
```

- Performs a discovery of services bound to TCP ports between 1 and 200 on a fc8_64 target host

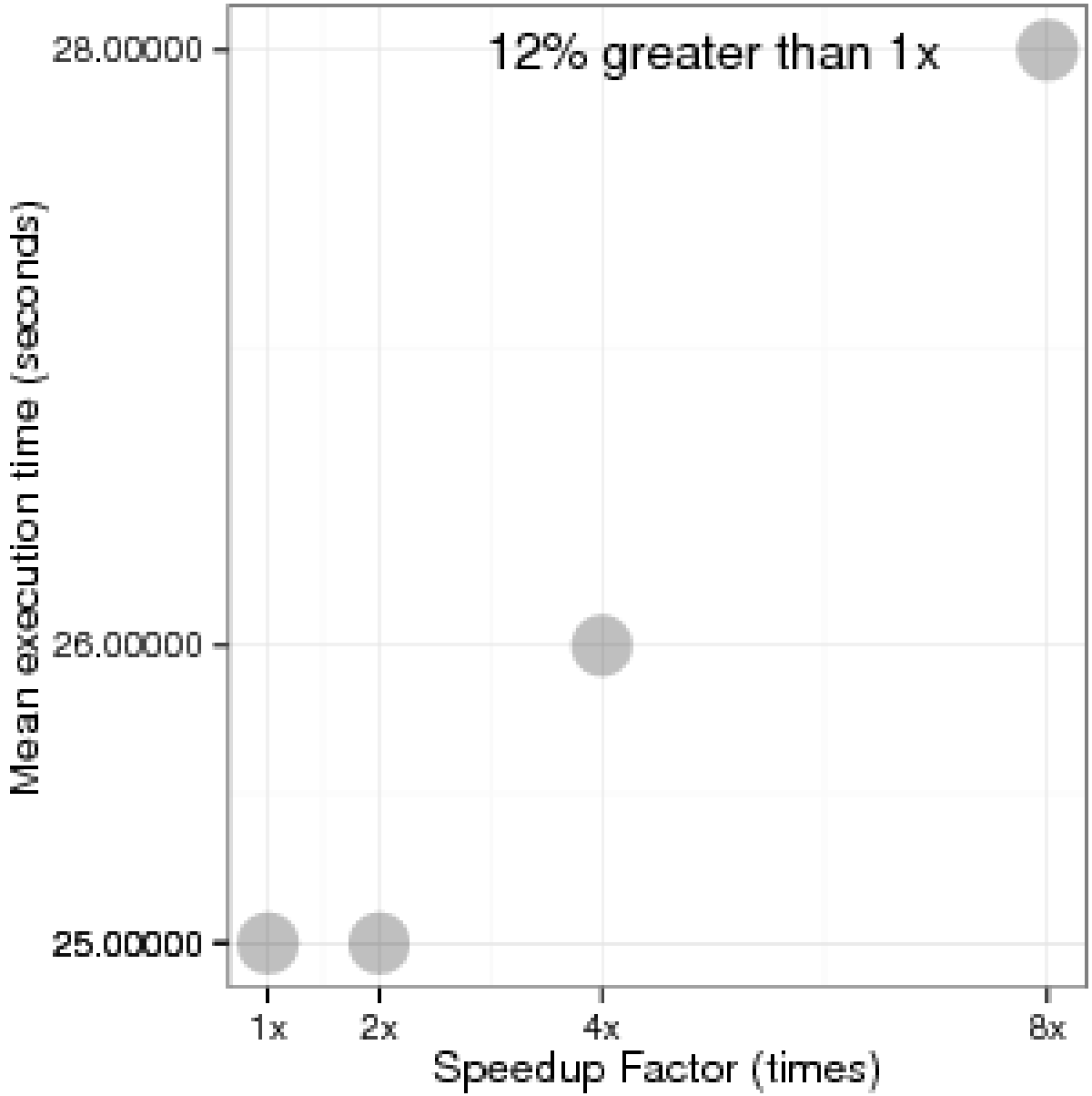
```
smbversion:

    use auxiliary/scanner/smb/
                        smb_version
    set THREADS 1
    set RHOSTS 10.2.34.2
    run
```

- Once a samba service had been discovered on the given target host, determine its precise version (3.0.21)

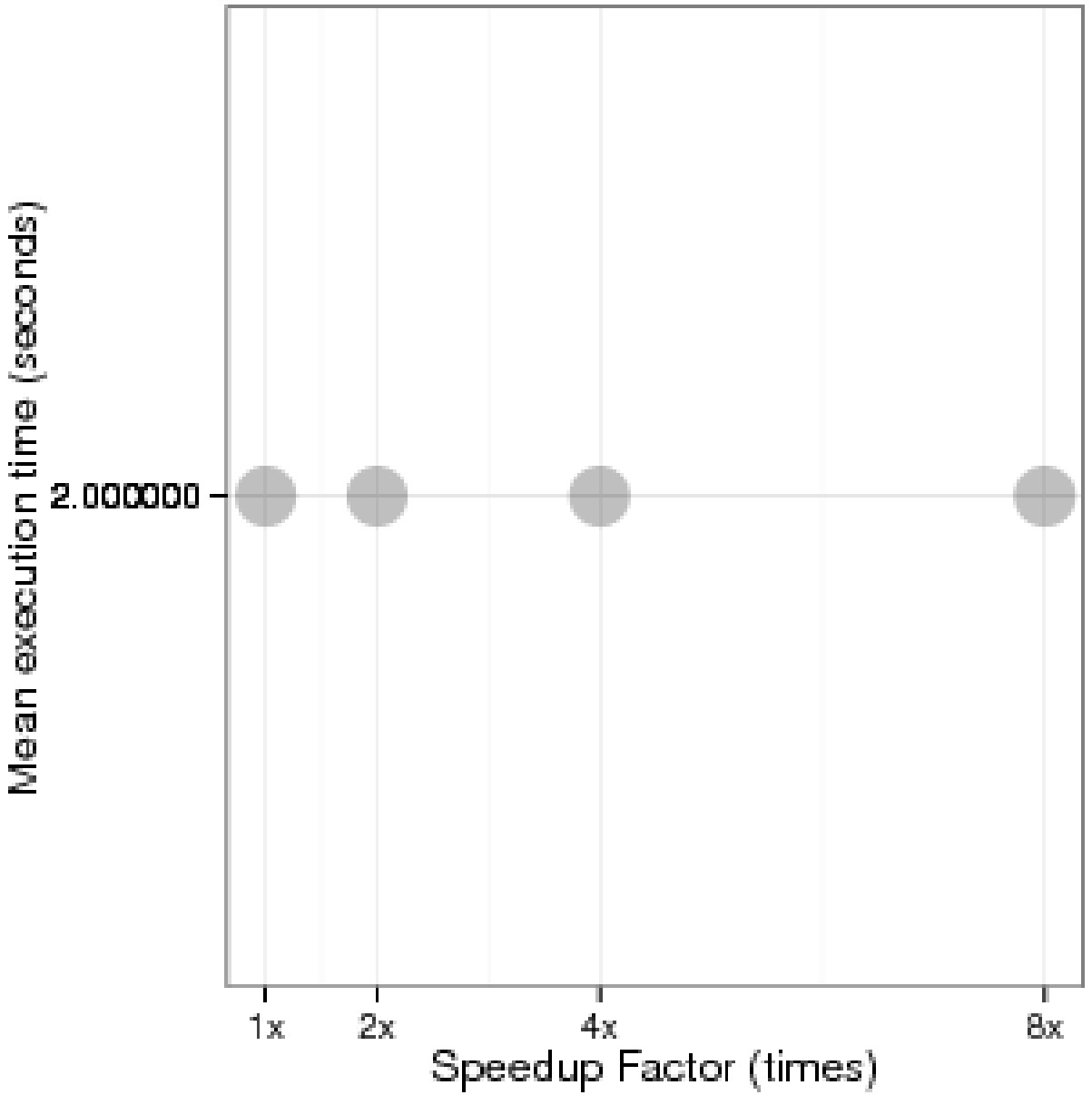
```
smblink:

    use auxiliary/admin/smb/
            samba_symlink_traversal
    set RHOST 10.2.34.2
    set SMBSHARE tmp
    set SMBTARGET rootfs
    run
```

- If the samba service had been discovered to be vulnerable, deploy a samba symlink traversal exploit which allows escalation of privileges and takeover of the root file system on the target host

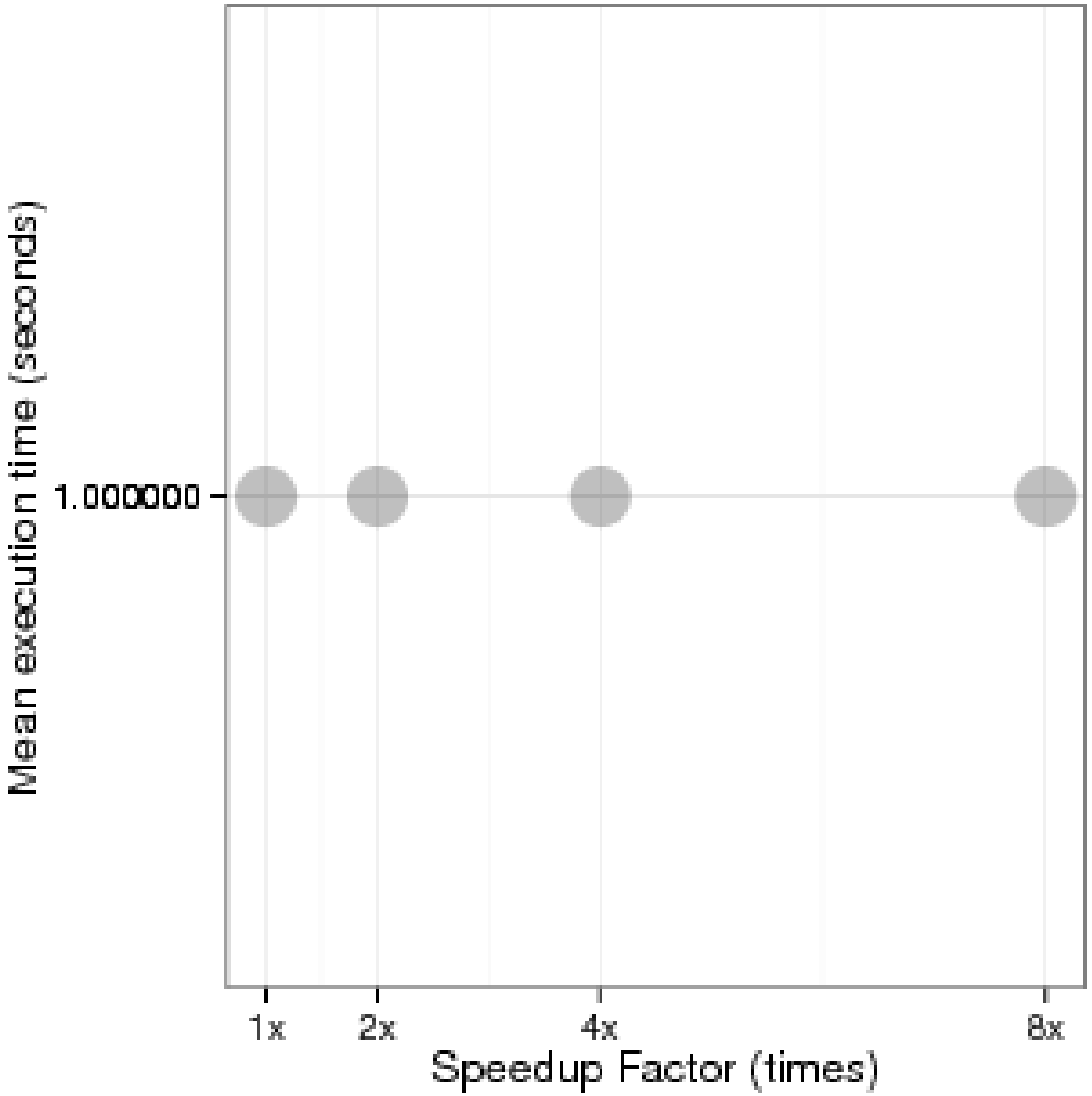
Metasploit 1 Fidelity Results



Metasploit 2 Fidelity Results



Metasploit 3 Fidelity Results



Result Interpretation

- Using diverse cyber-security tools, functional results are identical
- Timing fidelity acceptable under many circumstances
- Distortion is small for speedup factors up to 4x
- Even for 8x speedup factor, distortion under 13%
- Several sources of distortion:
  - Sampling interval
  - Speedup factor itself
  - Incorrect prediction of speedup for scenarios using variable speedup
  - Latency of speedup signal propagation in distributed testbed (i.e. non local source of virtual time)

Conclusions

- Cyber security testing under speedup is practical
- It can be a great benefit to testing by significantly reducing the duration of testing in real time
- Contact us at apoylisher@appcomsci.com, cserban@appcomsci.com

Thank You !