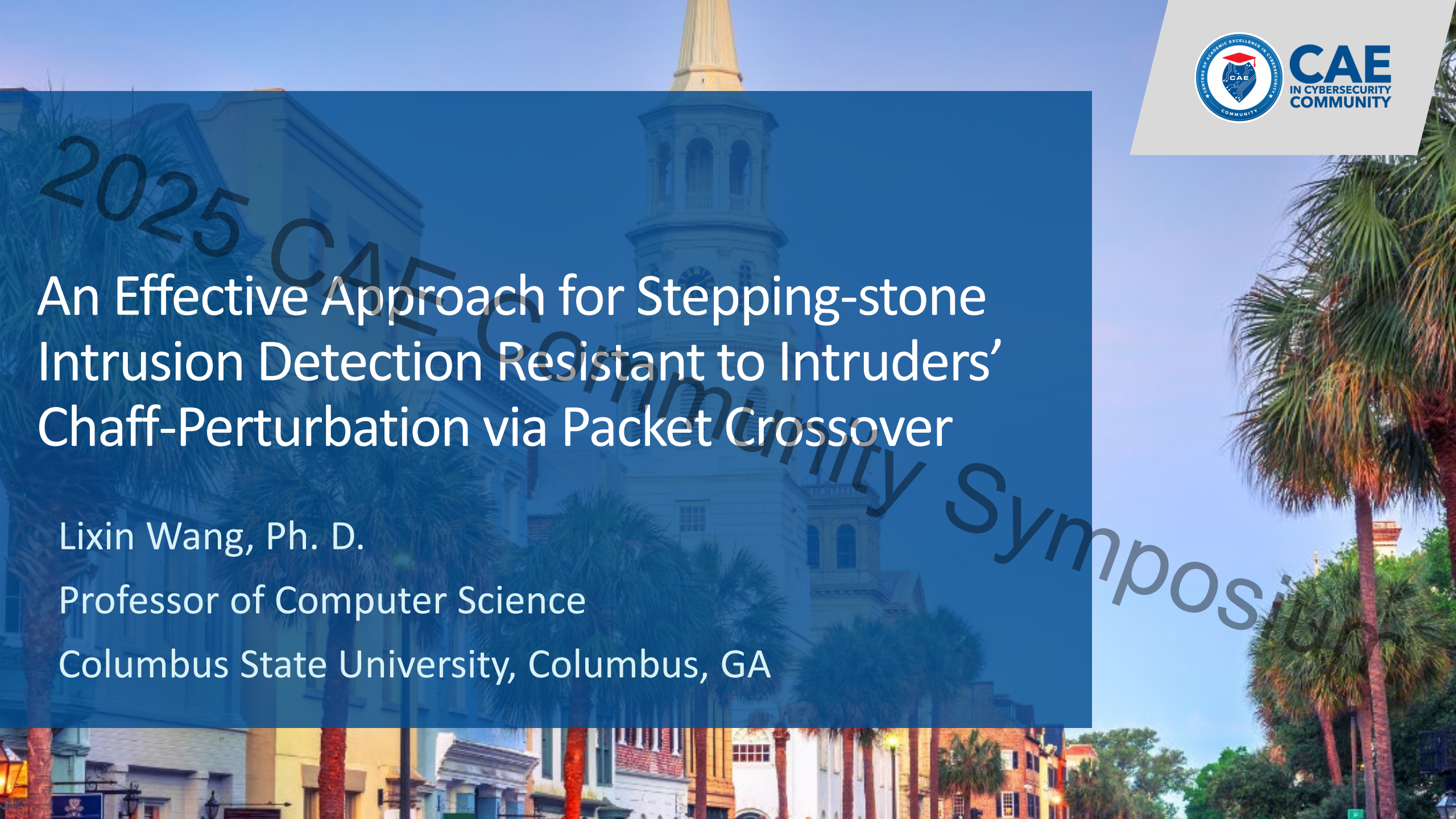


2025 CAE Community Symposium

An Effective Approach for Stepping-stone Intrusion Detection Resistant to Intruders' Chaff-Perturbation via Packet Crossover

Lixin Wang, Ph. D.

Professor of Computer Science

Columbus State University, Columbus, GA

Outline

- What is Stepping-stone Intrusion (SSI)?
- Literature review: Known methods for SSI detection (SSID)
- Our proposed method for SSID and contributions
- Conclusion

Intrusion through sending attack commands directly

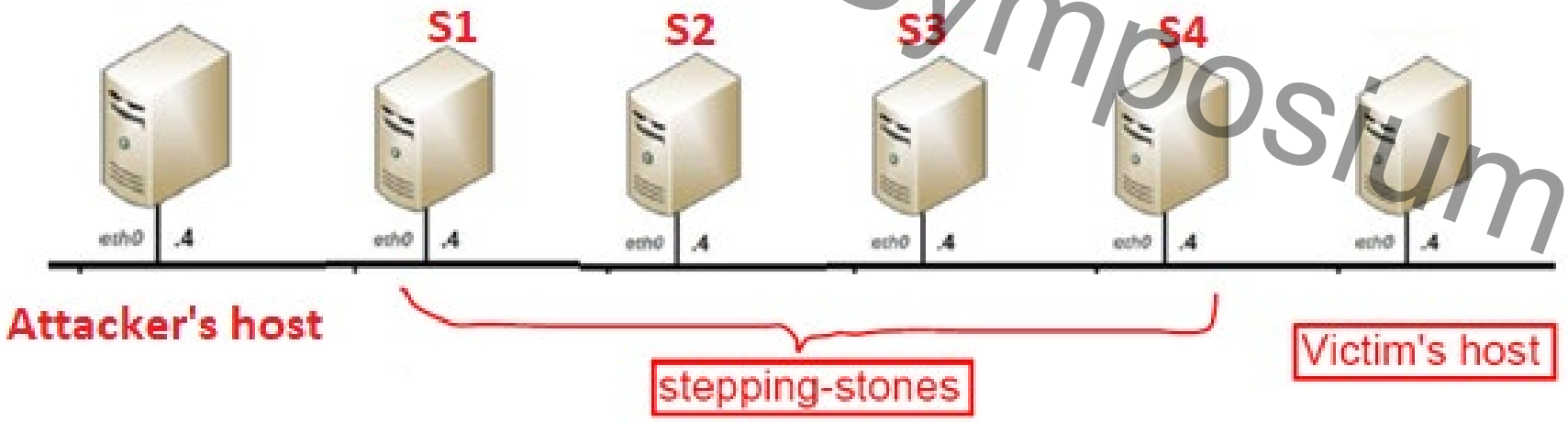
Attacking commands

Attacker

Victim

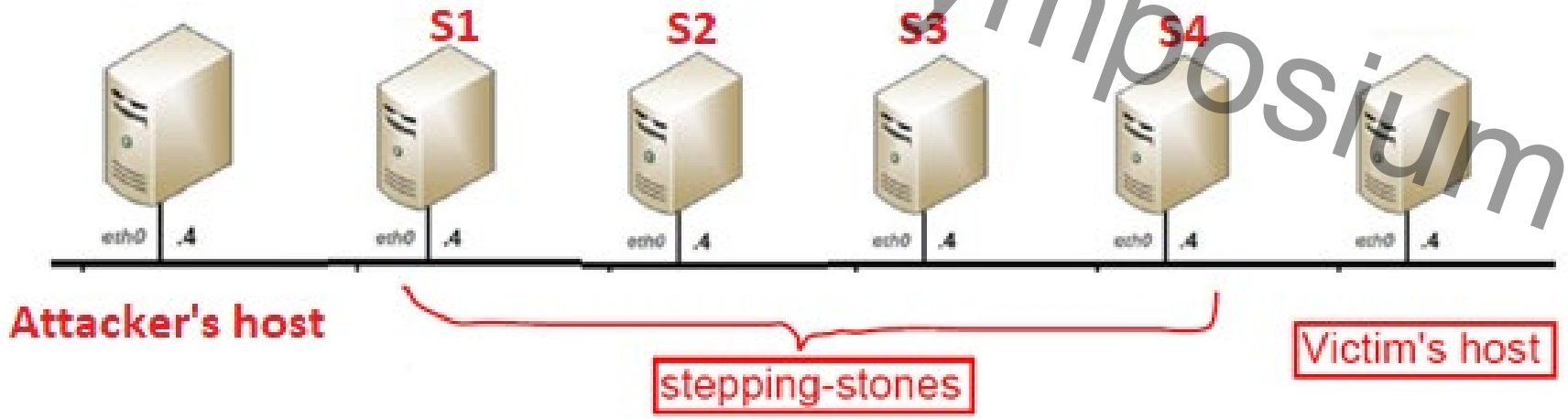
Stepping-Stone Intrusion (SSI)

- In order to reduce the chance of being detected, intruders often launch attacks through compromised hosts (called ***stepping-stones***)
- With SSI, an intruder remotely login these stepping-stones, and uses a chain of hosts as relay machines to launch the attack



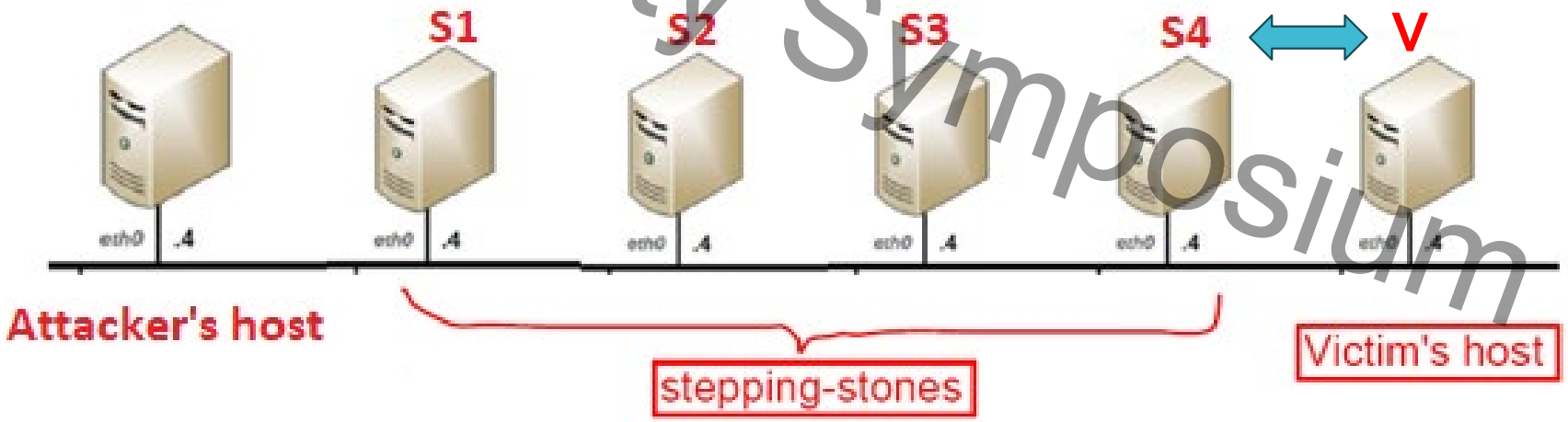
Intruders' Benefits of Using SSI

- The intruder is hidden behind a long interactive session
- **A nature of TCP**: Each TCP session between a client and a server is independent of other sessions even though the sessions may be relayed



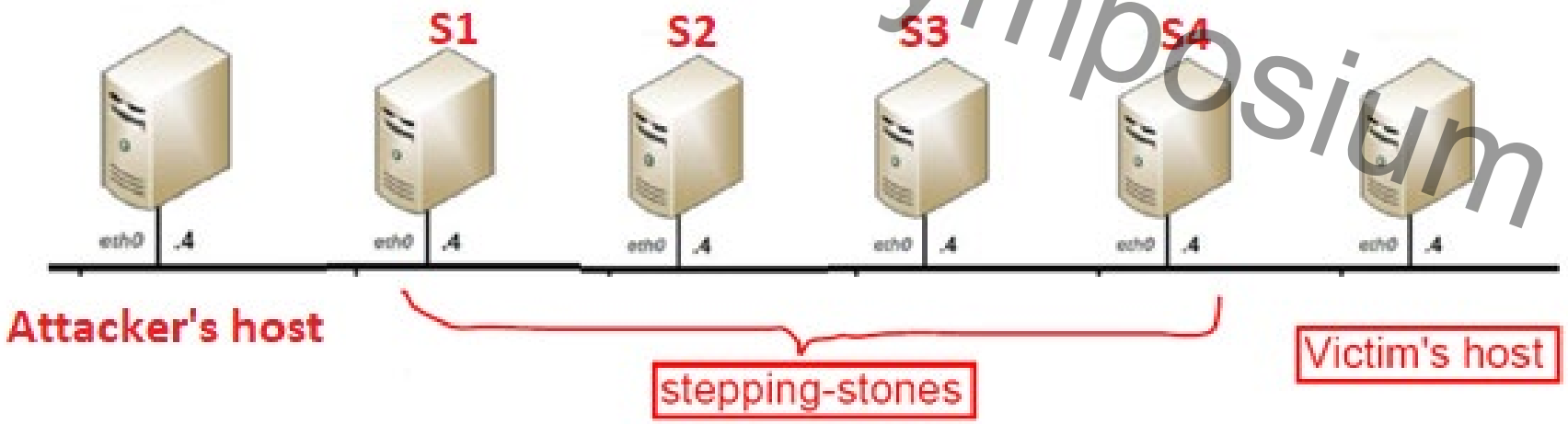
Intruders' Benefits of Using SSI (cont.)

- The victim host can only see packets from the last connection of the chain (**S4** ⟺ **V**)
- With SSI, it is extremely difficult for the victim to learn any information about the origin of the attack



Stepping-Stone Intrusion Detection (SSID)

- SSID can be performed at any of the stepping-stone hosts
- A detection **sensor** – the one has a detection program installed such as Wireshark, tcpdump, etc.
- **SSID is to determine whether the sensor host is used for malicious intrusion**

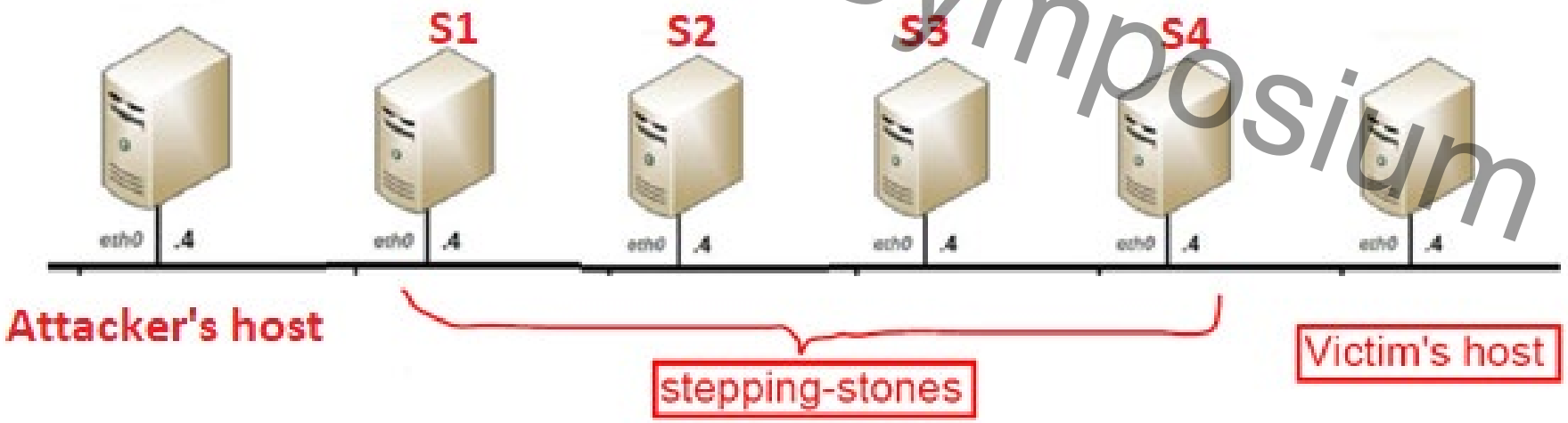


Stepping-Stone Intrusion Detection

- A great number of detection methods have been proposed for Stepping-stone Intrusion since 1995
  1. **Host-based approaches**
  2. **Network-based approaches**

Host-based SSID

- One type of approach for SSID is to compare all the incoming connections with all the outgoing connections of the sensor **to see if there exists a relayed pair**
- Called the ***host-based*** **methods** as only a single host (sensor) is used for the detection



Many Host-based SSID Methods Proposed

1. Content-thumbprint (1995)
2. Time-thumbprint (2000)
3. Watermarking detection (2001, 2011)
4. Packet counting (2006)
5. Random-walk detection (2007)

Issues with Host-based SSID Methods

- Stepping-stone hosts can be ***legally used*** by some apps to access a remote server. E.g.,

a widely used architecture in today's IT industry

- May produce high **false-positive errors**
  - A stepping-stone is used legally but SSID methods output SSI present

Network-based SSID

- ***Network-based* SSID -** estimate the number of connections from the attacker host to the victim
  - length of the connection chain
- The more hosts involved in a session to access a remote server, the slower the data communication
  - Accessing a server indirectly via **3 or more hosts** generates lots of unnecessary network traffic, and makes it very ineffective

Network-based SSID (cont.)

- Unless there are something hidden, otherwise, it doesn't make sense to access a remote sever via **3 or most stepping-stones**
- The number "**3**" is used because most legal apps barely used 3 or more stepping-stones to access a remote server
  - **If a remote server is accessed thru 3 or more stepping-stones, it is highly suspicious that it is an SSI**

Several Known Network-based SSID Methods

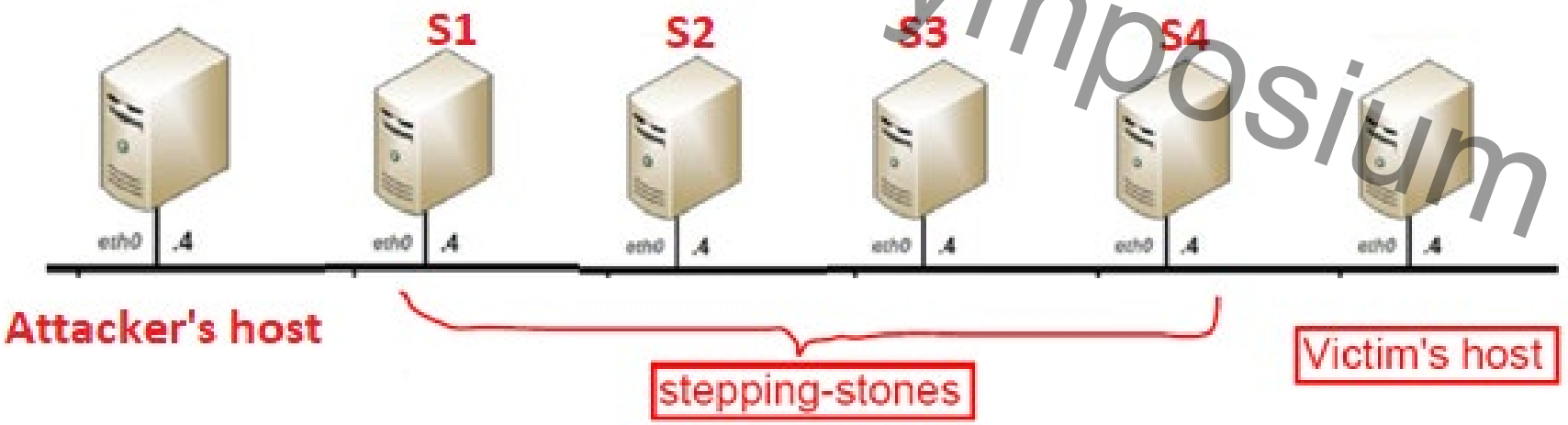
1. Yung's approach (Yung 2002)
2. Step-function approach (Yang et al. 2004)
3. Clustering-partitioning data mining approach (Yang et al. 2007)
4. K-means Clustering (Wang et al. 2021)

Issues with Existing Network-based SSID Methods

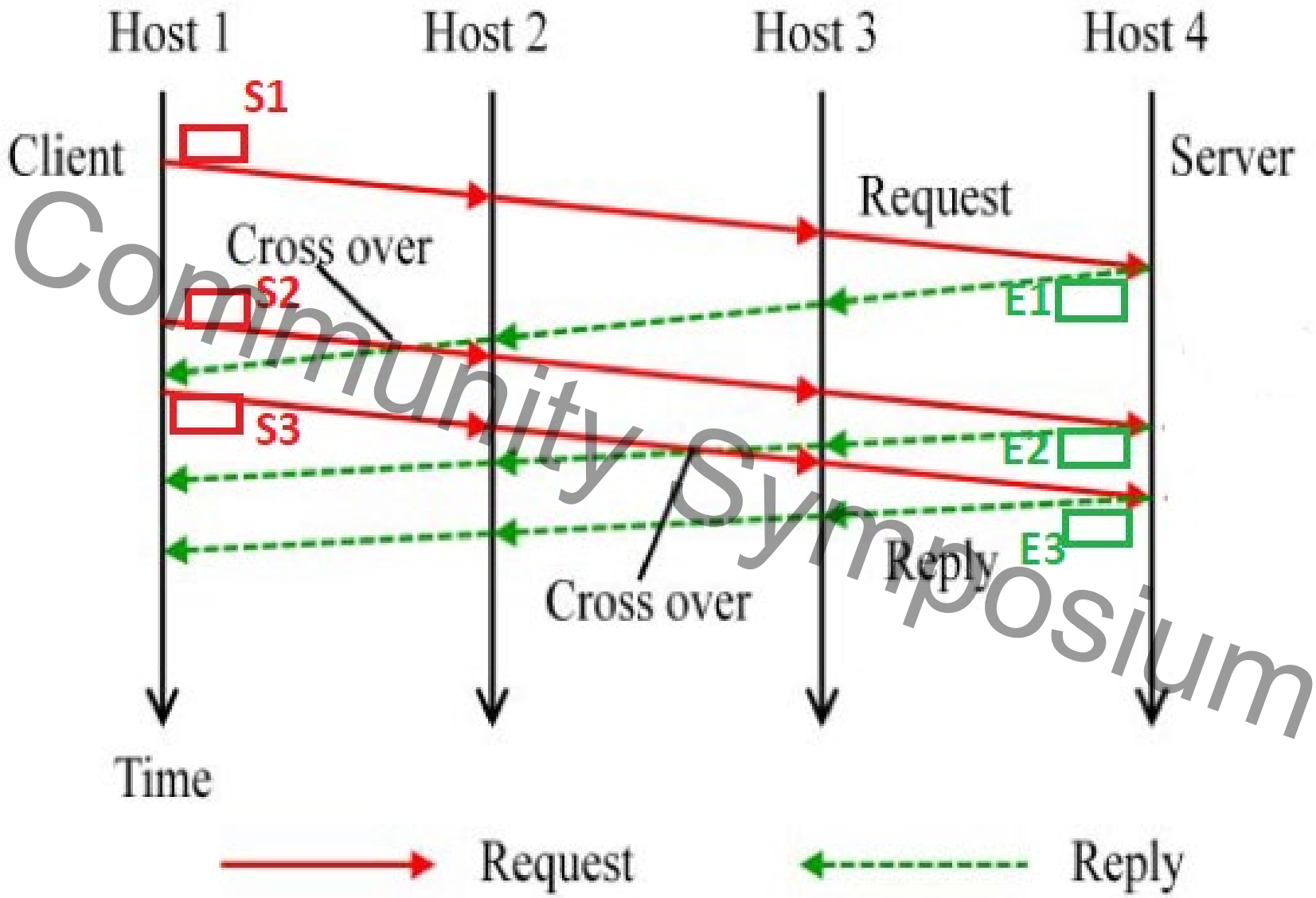
- Most existing network-based SSID only worked for network traffic without intruders' session manipulation
- These known SSID algorithms are
  - **either** weak to resist intruders' chaff attacks,
  - **or** having very limited capability in resisting attacker's session manipulation
  - **or** requiring a large number of TCP packets to be captured and analyzed, and thus they are not efficient for detection

Definitions of Send packets and Echo packets

- A ***Send packet*** is defined as a TCP **data packet** sent from the attacker host to victim host, with the **flag bit TCP.Flag.PSH** set
- An ***Echo packet*** is defined as a TCP **data packet** sent from the victim to attacker host, with the **flag bit TCP.Flag.PSH** set



Packet crossover occurs in which a new Send packet meets an Echo packet of a previous Send along the chain between a client and a server



Our SSID method design is based on an Observation

1. If the packet crossover ratio of an incoming connection of a sensor and one of its outgoing connections are almost equal, then it is highly likely that these two connections are a relayed pair
2. Otherwise, they are **not** a relayed pair

Algorithm to Compute Packet Crossover Ratio

Algorithm 1 (Compute Packet Crossover Ratio)

Input: a TXT file containing packet timestamp, packet type (Send or Echo), and index of Send or Echo
Output: Packet Crossover Ratio

```
sendIndex, echoIndex, crossoverCount = 0
while more packets in data capture file:
        if currentPacket is Acknowledgement:
                discard packet
                break

        else if currentPacket is Echo:
                if echoIndex less than sendIndex:
                        crossoverCount+=(sendIndex–echoIndex)
                echoIndex += 1

        else if (currentPacket is Send):
                sendIndex += 1

PacketCrossoverRatio = crossoverCount / (2 * echoIndex)
Print PacketCrossoverRatio
```

Proposed SSID Algorithm

Algorithm 2: SSID Algorithm using Packet Crossover

Input: None

Output: SSI detected or not

Begin:

1. Set up a connection chain A→S1→S2→S3→V of length four, where the hosts S1, S2 and S3 are the stepping-stones (S1 serves as the sensor), host A the attacker, and host V the victim. The length of the downstream sub-chain from S1 to V is three.
2. Some standard Linux commands (such as ls, dir, mkdir, etc.) are entered into a terminal in the attacker host A for a couple of minutes, and at the same time all the packets are captured at the sensor S1 from the connection S1→S2 in the chain. Totally, 10 datasets will be captured. Then we use the Packet Crossover Ratio algorithm (Algorithm 1 of [17]) to calculate the packet crossover ratio for each dataset of the above captured packets.
3. Calculate the intrusion threshold crossover ratio which is the average packet crossover ratio among the 10 captured datasets at Step 2.
4. To perform SSID, at the same time, we also use host S1 as the sensor and observe one of its outgoing links. We then determine whether this outgoing link from the sensor S1 is used by an intruder for a malicious SSI. We capture 10 datasets at the sensor S1 from this outgoing connection and calculate the average packet crossover ratio over all the 10 captured datasets using the Packet Crossover Ratio algorithm (Algorithm 1 of [17]).
5. If the average packet crossover ratio obtained at Step 4 is greater than or equal to the intrusion threshold crossover ratio obtained at Step 3, it is most likely that this out-going link is used by a hacker for malicious SSI.
6. Repeat Step 4 for every outgoing link from the sensor host S1 (except for the connection S1→S2 in the chain created in Step 2) to see whether it is used by a hacker for malicious SSI.

End

Summary

- Our proposed network-based SSID approach using packet crossover is resistant to intruders' chaff manipulation
- Since packet crossover can be easily calculated, our proposed SSID method is easy to implement as well as efficient
- According to our experiment results, the proposed SSID algorithm performs perfectly in resisting intruders' chaff-perturbation up to 50% chaff rate

Thank you!

2025 CAE Community Symposium

Questions?