Remote Attestation using AMD-Xilinx U280 on FABRIC

Hyunsuk Bang, Illinois Tech
Nishanth Shyamkumar, Illinois Tech
Christopher E. Neely, AMD-Xilinx
Nik Sultana Illinois Tech

ILLINOIS TECH
College of Computing

Introduction

- The behavior of network equipment has been subverted in attacks that surreptitiously reconfigured the equipment to enable subsequent attacks.
- We need a systematic way to recognize that the configuration of network devices is not aligned with our instructions.
- Using **Remote Attestation** we can enable dynamic assessment of network security and configuration characteristics.

Motivation

1) Adapt third-party, “black box” equipment to provide verifiable evidence about its configuration and behavior.

2) Construct a stateful, secure network function that provides this functionality at line rate.

3) Design distributed systems that can validate evidence integrity of multiple target switches to detect tampering.

References

http://transparnet.cs.iit.edu/

Acknowledgement

We thank the KNIT8 organizers for a travel stipend that enabled the first author to present this work in person.

```
topology = '2,I - 2,S - 2,E - 2,I - 2,S - 2,E'
ra.submit_config(topology)
```

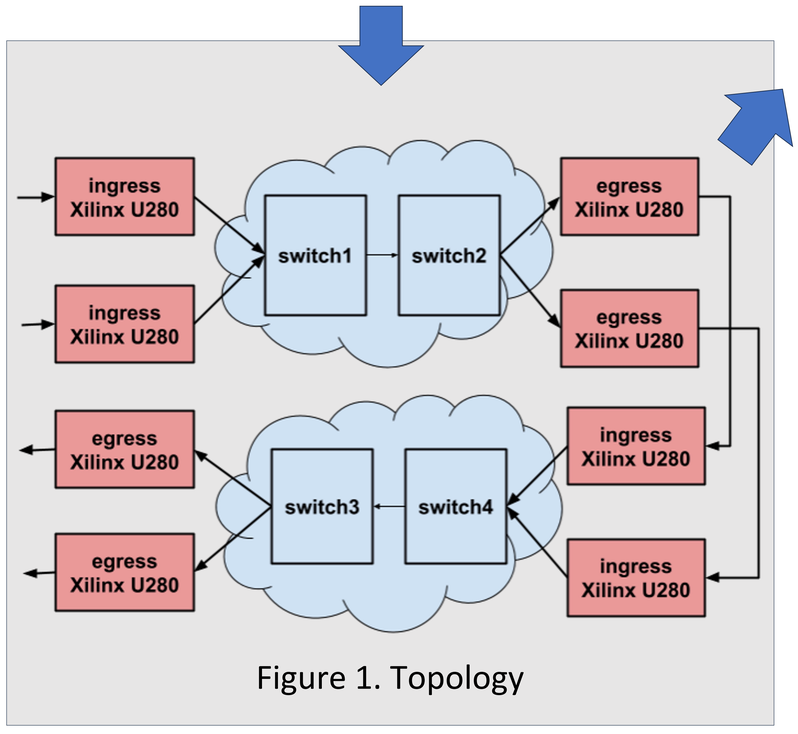

Figure 1. Topology

```
Internet Protocol Version 6, Src: fe80::b6:acff:fe20:c541, Dst: fe80::c83:feff:fe10:e9cd
    0110 .... = Version: 6
  > .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... 1000 1010 0110 1111 1010 = Flow Label: 0x8a6fa
    Payload Length: 1128
    Next Header: IPv6 Hop-by-Hop Option (0)
    Hop Limit: 64
    Source Address: fe80::b6:acff:fe20:c541
    Destination Address: fe80::c83:feff:fe10:e9cd
  v IPv6 Hop-by-Hop Option
      Next Header: UDP (17)
      Length: 14
      [Length: 120 bytes]
```

```
0030  fe ff fe 10 e9 cd 11 0e  37 74 00 00 ef 98 e6 c2   Epoch
0040  08 02 b2 4a 13 e4 34 37  fe c4 0b 00 0a cb 98 37   Program
0050  fc 54 f1 8c 87 cd 21 6d  7d 42 62 c1 24 30 c2 3e   Register
0060  7e 74 87 89 ab 1c 35 d9  f1 e3 6f 2c dc 51 af c1   Table
0070  5a e8 3b 94 fa 83 d0 fe  6a 0c a8 af e7 01 98 37   Program Digest
0080  fc 54 f1 8c 87 cd 21 6d  7d 42 62 c1 24 30 c2 3e   Register Digest
0090  7e 74 87 89 ab 1c 35 d9  f1 e3 6f 2c dc 51 af c1   Table Digest
00a0  5a e8 3b 94 fa 83 d0 fe  6a 0c a8 af e7 01 cf 6a
```

Figure 2. Captured evidence

```
[  5]   0.00-1.00   sec  11.9 MBytes   100 Mbits/sec  9616
[  5]   1.00-2.00   sec  11.9 MBytes   100 Mbits/sec  9614
[  5]   2.00-3.00   sec  11.9 MBytes   100 Mbits/sec  9617
[  5]   3.00-4.00   sec  11.9 MBytes   100 Mbits/sec  9614
[  5]   4.00-5.00   sec  11.9 MBytes   100 Mbits/sec  9616
[  5]   5.00-6.00   sec  11.9 MBytes   100 Mbits/sec  9615
[  5]   6.00-7.00   sec  11.9 MBytes   100 Mbits/sec  9615
[  5]   7.00-8.00   sec  11.9 MBytes   100 Mbits/sec  9617
[  5]   8.00-9.00   sec  11.9 MBytes   100 Mbits/sec  9614
^[[B[  5]   9.00-10.00  sec  11.9 MBytes   100 Mbits/sec  9616
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.000 ms  0/96154 (0%)  sender
[  5]   0.00-10.32  sec   123 MBytes   100 Mbits/sec  0.063 ms  2748/96154 (2.9%)  receiver
```

Figure 3. iperf3 session

Approach

- Two AMD-Xilinx U280 Smart NICs are used to “sandwich” a switch that is treated as a “black box”.
- Control Planes of Xilinx U280 (consumers) synchronously retrieves configuration from the host of black-box switches (producer) and updates lookup tables on Xilinx U280s.
- Ingress Xilinx U280 embeds the switch evidence into the packet.
- Egress Xilinx U280 verifies the integrity of evidence to detect alterations through checksum calculatåion and P4 lookup table.
- Python scripts automate site and network configurations for ease of implementation and result reproducibility.

Results

- We used 12 Xilinx U280s, configured as shown in Fig 1.
- We compare the HBH header (Fig 2) as seen at the receiver and verify that the state values have been transmitted successfully and correctly.
- We monitor the output from the egress Xilinx U280 to see whether a predicted/unpredicted changes has been made
- We use iperf (Fig 3) across the first ingress sites and the last egress sites. The software switch is the performance bottleneck.