Don’t Yank My Chain
Auditable NF Service Chaining

Guyue Liu, Hugo Sadok, Anne Kohlbrenner, Bryan Parno, Vyas Sekar, Justine Sherry

Carnegie Mellon University
Network Function Virtualization (NFV)
Network Function Virtualization (NFV)
Network Function Virtualization (NFV)

Benefits: ① Cost ② Elasticity ③ Richer Policies
Policies in NFV

Policy Graph

in → FW → IDS → Transc. → out

dst port 80 or 443
!(dst port 80 or 443)
NFs are mandated by legal and policy requirements

HIPPA

PCI

FERPA
Regulatory Compliance (NIST 800-53)*
Regulatory Compliance (NIST 800-53)*

*There is also ISO 270001, which is an equivalent international standard
Regulatory Compliance (NIST 800-53)*

Periodically test that the infrastructure is running properly

*There is also ISO 270001, which is an equivalent international standard
Regulatory Compliance (NIST 800-53)*

Periodically test that the infrastructure is running properly

System must provide logs of anomalies and past behavior

*There is also ISO 270001, which is an equivalent international standard
Regulatory Compliance (NIST 800-53)*

- **Periodically test** that the infrastructure is running properly

- System must provide **logs** of anomalies and past behavior

- **Independent auditors** must verify that security mechanisms are in place and running correctly

*There is also ISO 270001, which is an equivalent international standard*
Auditing Network Function Chains

Traditional Network

NF 1

NF 2

NF 3
Auditing Network Function Chains

Traditional Network

NF 1  NF 2  NF 3
Auditing Network Function Chains

Traditional Network

NF 1 ➔ NF 2 ➔ NF 3

NFV

VM ➔ VM ➔ VM

Host 1 ➔ Host 2
Auditing Network Function Chains

Traditional Network

NF 1
NF 2
NF 3

NFV

VM
Host 1

VM
Host 2
Auditing Network Function Chains

Traditional Network

NF 1 → NF 2 → NF 3

NFV

VM → VM → VM → VM

Host 1

Host 2
Auditing Network Function Chains

Traditional Network

NF 1

NF 2

NF 3

NFV

VM

Host 1

Host 2
Auditing Network Function Chains

Traditional Network:
- NF 1
- NF 2
- NF 3

NFV:
- VM
- Host 1
- Host 2
Auditing Network Function Chains

Traditional Network

NF 1 -> NF 2 -> NF 3

NFV

VM

Host 1

VM

Host 2
Traditional networks

1. Auditors can rely on the network topology to ensure that the correct NF chain is being used.

2. Auditors can inspect and approve HW boxes.

3. Auditors can trust logs captured by the HW boxes.
Traditional networks

1. Auditors can rely on the network topology to ensure that the correct NF chain is being used.

2. Auditors can inspect and approve HW boxes.

3. Auditors can trust logs captured by the HW boxes.

NFV
Traditional networks

1. Auditors can rely on the network topology to ensure that the correct NF chain is being used.
2. Auditors can inspect and approve HW boxes.
3. Auditors can trust logs captured by the HW boxes.

NFV

1. NF chains are dynamic.
Traditional networks

1. Auditors can rely on the network topology to ensure that the correct NF chain is being used
2. Auditors can inspect and approve HW boxes
3. Auditors can trust logs captured by the HW boxes

NFV

1. NF chains are dynamic
2. Software NFs can be modified by an attacker
Traditional networks

1. Auditors can rely on the network topology to ensure that the correct NF chain is being used.
2. Auditors can inspect and approve HW boxes.
3. Auditors can trust logs captured by the HW boxes.

NFV

1. NF chains are dynamic.
2. Software NFs can be modified by an attacker.
3. Logs stored in the server can be tampered with.
These limitations are not fundamental to NFV
These limitations are not fundamental to NFV

With AuditBox for NFV, auditors have even stronger auditing guarantees than traditional NF deployments
AuditBox
AuditBox

Provides auditing capabilities to NFV deployments
AuditBox

Provides auditing capabilities to NFV deployments

① NF functionality cannot be modified/manipulated

② Traffic is steered between NFs according to the administrator’s policy

③ Provide logs that attest that the correct policy is being followed

④ Good performance
AuditBox

Key Techniques:

① Secure Enclaves

② NF Hop-by-Hop Updated Attestation

③ Secret Logging

④ Efficient Crypto Mechanisms
Logical view of an NF chain

NF 1 —— NF 2 —— NF 3

1. Secure Enclaves

Example: switch, OS
Secure Enclaves

Logical view of an NF chain

NF 1 → NF 2 → NF 3

Untrusted

E.g., switch, OS
Logical view of an NF chain

Need to attest that all components in the path are correct
Run NFs inside secure enclaves (e.g., Intel SGX)
Run NFs inside secure enclaves (e.g., Intel SGX)

- Auditors have guarantees that the audited NF software is running
Run NFs inside secure enclaves (e.g., Intel SGX)

- Auditors have guarantees that the audited NF software is running
- Remaining untrusted functionality is responsible for packet forwarding
Verified Routing Protocols (VRPs)

Examples: EPIC [USENIX '20], OPT [SIGCOMM '14], ICING [CoNEXT '11]
Verified Routing Protocols (VRPs)

Examples: EPIC [USENIX '20], OPT [SIGCOMM '14], ICING [CoNEXT '11]

VRP Assumptions:

NFV Needs:
Verified Routing Protocols (VRPs)

Examples: EPIC [USENIX '20], OPT [SIGCOMM '14], ICING [CoNEXT '11]

VRP Assumptions:
- Immutable Packets

NFV Needs:
- Mutable Packets
Verified Routing Protocols (VRPs)

Examples: EPIC [USENIX '20], OPT [SIGCOMM '14], ICING [CoNEXT '11]

VRP Assumptions:
- Immutable Packets
- Pre-known Paths

NFV Needs:
- Mutable Packets
- Dynamic Paths
Verified Routing Protocols (VRPs)

Examples: EPIC [USENIX '20], OPT [SIGCOMM '14], ICING [CoNEXT '11]

VRP Assumptions:

- Immutable Packets
- Pre-known Paths
- Stateless Processing Nodes

NFV Needs:

- Mutable Packets
- Dynamic Paths
- Stateful Processing Nodes
2 NF Hop-by-Hop Updated Attestation
NF Hop-by-Hop Updated Attestation

- A **shim** in every enclave mediates all incoming and outgoing packets
- A **shim** in every enclave mediates all incoming and outgoing packets
- Attestation happens between **pairs of shims**
NF Hop-by-Hop Updated Attestation

Enclave

NF 1
Shim

Enclave

NF 2
Shim

Audit Trailer
Packet ID | SRC NF | DST NF | Tag
NF Hop-by-Hop Updated Attestation

\[ \text{Tag} = \text{GMAC(key, Packet | Packet ID | SRC NF | DST NF)} \]
NF Hop-by-Hop Updated Attestation

Audit Trailer

\[ \text{Tag} = \text{GMAC(key, Packet | Packet ID | SRC NF | DST NF)} \]

*AuditBox also supports flow-level correctness which detects packet duplication, reordering and drops (refer to paper for details)
AuditBox Design Overview

Control Plane

Data Plane
AuditBox Design Overview

Control Plane

Data Plane

NF

Shim

Host 1

Host N

Untrusted Network (e.g., switches, routers)

Intel SGX Enclave
AuditBox Design Overview

Control Plane

Data Plane

Runtime Correctness

① Secure Enclaves
② Hop-by-hop Verification Protocol
AuditBox Design Overview

Runtime Correctness
1. Secure Enclaves
2. Hop-by-hop Verification Protocol
AuditBox Design Overview

- **Offline Auditability**
  - ③ Secret Logging

- **Runtime Correctness**
  - ① Secure Enclaves
  - ② Hop-by-hop Verification Protocol

**Control Plane**
- Controller
- Secure Channels
- Audit Trails
- Administrator/Auditor

**Data Plane**
- NF
- Shim
- Host 1
- Host N
- Untrusted Network (e.g., switches, routers)
- Intel SGX Enclave
Potentially Cryptographically Expensive
Potentially Cryptographically Expensive

Verify incoming packet by computing GMAC
Potentially Cryptographically Expensive

NF modifies packet

Verify incoming packet by computing GMAC

Enclave

Shim

NF 2

Shim
Potentially Cryptographically Expensive

- NF modifies packet
- NF 2
- Shim
- Enclave

Verify incoming packet by computing GMAC
Potentially Cryptographically Expensive

NF modifies packet

Verify incoming packet by computing GMAC
Potentially Cryptographically Expensive

NF modifies packet

Enclave

NF 1

Shim

P'

Verify incoming packet by computing GMAC

Enclave

NF 2

Shim

Update AuditTrailer by computing GMAC over modified packet
Potentially Cryptographically Expensive

NF modifies packet

Verify incoming packet by computing GMAC

NF 1

Shim

P'

Update AuditTrailer by computing GMAC over modified packet

NF 2

Shim

Enclave
Potentially Cryptographically Expensive

- Asymmetric key at every hop
- 2 GMACs at every hop
### Efficient Crypto Mechanisms

**Audit Trailer**

<table>
<thead>
<tr>
<th>Packet ID</th>
<th>SRC NF</th>
<th>DST NF</th>
<th>Tag</th>
</tr>
</thead>
</table>

Tag = GMAC(key, Packet | Packet ID | SRC NF | DST NF)

One symmetric key for all NFs in the same policy pipelet
Efficient Crypto Mechanisms
Verify incoming packet by computing GMAC.
Verify incoming packet by computing GMAC.

GMAC

NF modifies packet

Enclave

Shim

Headers | Payload | Trailer
4 Efficient Crypto Mechanisms

GMAC

Verify incoming packet by computing GMAC
Verify incoming packet by computing GMAC

NF modifies packet
Verify incoming packet by computing GMAC

Update AuditTrailer by computing GMAC over modified packet
Efficient Crypto Mechanisms

Updatable GMAC [1]: Reuse first GMAC when computing the second GMAC

GMAC for incoming packet

<table>
<thead>
<tr>
<th>Headers</th>
<th>Payload</th>
<th>Trailer</th>
</tr>
</thead>
</table>

Blocks that changed

GMAC for outgoing packet

Evaluation

1. **Proofs:** We provide security proofs that AuditBox can achieve both runtime correctness and offline auditability.

2. **Functionality Evaluation:** AuditBox correctly detects a broad class of policy violations.

3. **Performance Evaluation:** AuditBox enables auditing for unmodified NFs with low overhead.
Achieves 18 Gbps for a simple NF chain
Summary

AuditBox is the first NFV auditing system. It leverages trusted execution environments to provide:

• Runtime correctness
• Offline auditability
• While still achieving good performance

Contact: sadok@cmu.edu