Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks

EndBox: Scalable Middlebox Functions Using Client-Side Trusted Execution

David Goltzsche,[1] Signe Rüsch,[1] Manuel Nieke,[1] Sébastien Vaucher,[2] Nico Weichbrodt,[1] Valerio Schiavoni,[2] Pierre-Louis Aublin,[3] Paolo Costa,[4] Christof Fetzer,[5] Pascal Felber,[2] Peter Pietzuch[3] and Rüdiger Kapitza[1]

[1]TU Braunschweig goltzsche@ibr.cs.tu-bs.de @d_goltzsche

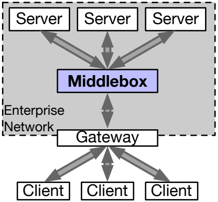
[2]University of Neuchâtel [3]Imperial College London [4]Microsoft Research [5]TU Dresden

What Are Middleboxes?

- **Middleboxes** are essential parts of large networks
  - Example: enterprise networks

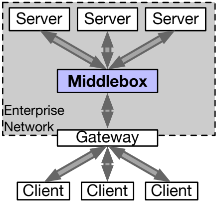
- Functions related to **security** or **performance**

- **Current best practice:**
  central deployment as physical boxes
  - High infrastructure and management costs
    (Sherry et al. SIGCOMM'12)
  - Scalability issues with growing client numbers



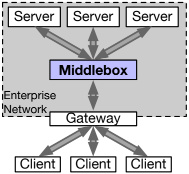
What Are Middleboxes?

- **Middleboxes** are essential parts of large networks
  - Example: enterprise networks
- Functions related to **security** or **performance**
- **Current best practice:**
  central deployment as physical boxes
  - High infrastructure and management costs
    (Sherry et al. SIGCOMM'12)
  - Scalability issues with growing client numbers

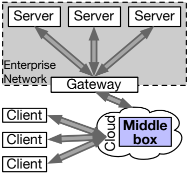


Problem: Middleboxes are necessary for large networks, but come at **high costs** and **do not scale** well with number of clients.

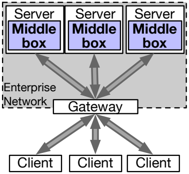
Placement of Middleboxes



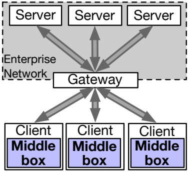
(a) Centralised



(b) Cloud-based



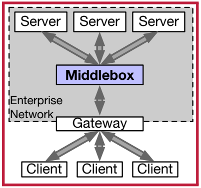
(c) Server-side



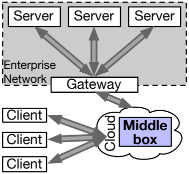
(d) Client-side

	(a)	(b)	(c)	(d)
Low infra. cost				
Low latency				
Good scalability				
Trusted infrastructure				
Easy administration				

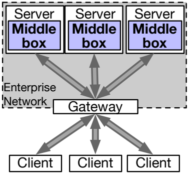
Placement of Middleboxes



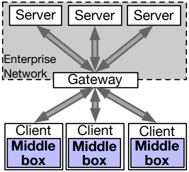
(a) Centralised



(b) Cloud-based



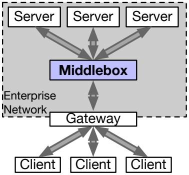
(c) Server-side



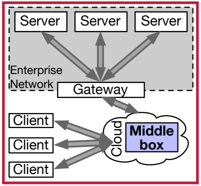
(d) Client-side

	(a)	(b)	(c)	(d)
Low infra. cost	✗			
Low latency	✔			
Good scalability	✗			
Trusted infrastructure	✔			
Easy administration	✔			

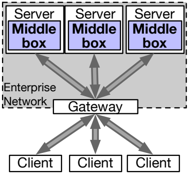
Placement of Middleboxes



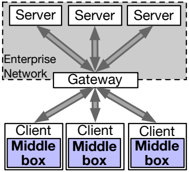
(a) Centralised



(b) Cloud-based



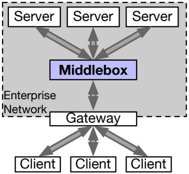
(c) Server-side



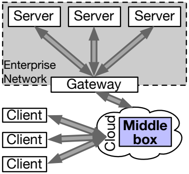
(d) Client-side

	(a)	(b)	(c)	(d)
Low infra. cost	✗	✔		
Low latency	✔	✗		
Good scalability	✗	✔		
Trusted infrastructure	✔	✗		
Easy administration	✔	✔		

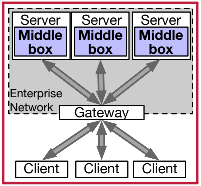
Placement of Middleboxes



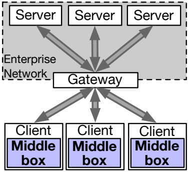
(a) Centralised



(b) Cloud-based



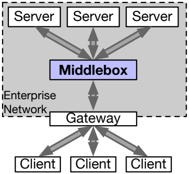
(c) Server-side



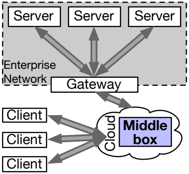
(d) Client-side

	(a)	(b)	(c)	(d)
Low infra. cost	✗	✔	✔	
Low latency	✔	✗	✔	
Good scalability	✗	✔	✗	
Trusted infrastructure	✔	✗	✔	
Easy administration	✔	✔	✗	

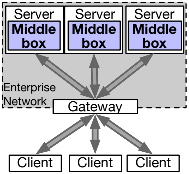
Placement of Middleboxes



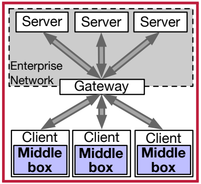
(a) Centralised



(b) Cloud-based



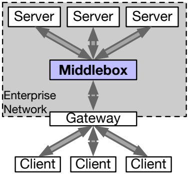
(c) Server-side



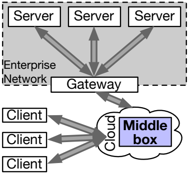
(d) Client-side

	(a)	(b)	(c)	(d)
Low infra. cost	✗	✔	✔	✔
Low latency	✔	✗	✔	✔
Good scalability	✗	✔	✗	✔
Trusted infrastructure	✔	✗	✔	✗
Easy administration	✔	✔	✗	✗

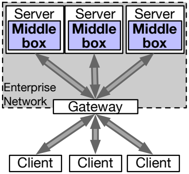
Placement of Middleboxes



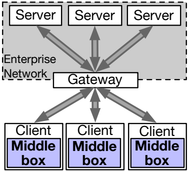
(a) Centralised



(b) Cloud-based



(c) Server-side



(d) Client-side

	(a)	(b)	(c)	(d)
Low infra. cost	✗	✔	✔	✔
Low latency	✔	✗	✔	✔
Good scalability	✗	✔	✗	✔
Trusted infrastructure	✔	✗	✔	✔
Easy administration	✔	✔	✗	✔

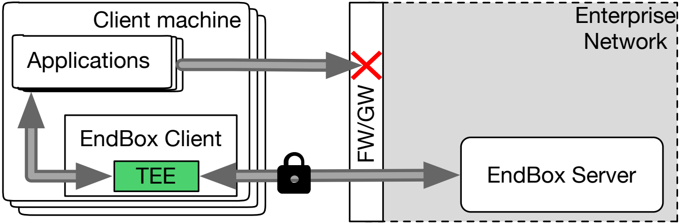
✔ with EndBox

EndBox targets enterprise networks and places middleboxes on **untrusted clients.**

Outline

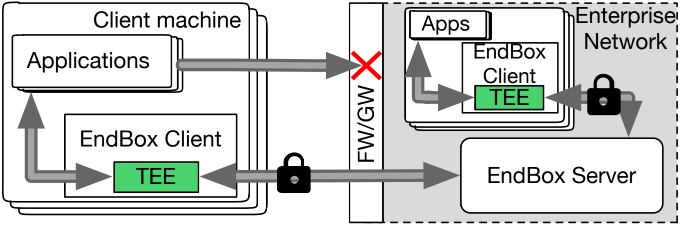
- Introduction to Middleboxes
- Design of EndBox
- Evaluation of EndBox
- Related Work
- Conclusion

Approach of EndBox



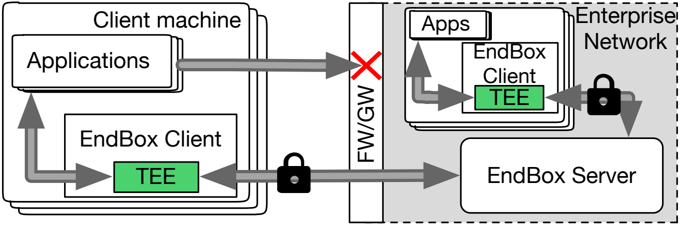
- **Untrusted clients** can manipulate or circumvent traffic analysis
  ➥ Client traffic routed through **trusted execution environments** (TEEs)
- Inside TEE, packets are **processed, signed and encrypted**
- **Unsigned outgoing traffic dropped** by firewall/gateway (FW/GW)
- **Encrypted incoming traffic** cannot be encrypted outside of TEE

Approach of EndBox



- **Untrusted clients** can manipulate or circumvent traffic analysis
  ↳ Client traffic routed through **trusted execution environments** (TEEs)
- Inside TEE, packets are **processed, signed and encrypted**
- **Unsigned outgoing traffic dropped** by firewall/gateway (FW/GW)
- **Encrypted incoming traffic** cannot be encrypted outside of TEE

Approach of EndBox

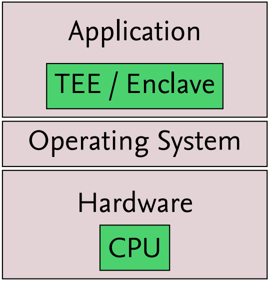


- **Untrusted clients** can manipulate or circumvent traffic analysis
  ↳ Client traffic routed through **trusted execution environments** (TEEs)
- Inside TEE, packets are **processed, signed and encrypted**
- **Unsigned outgoing traffic dropped** by firewall/gateway (FW/GW)
- **Encrypted incoming traffic** cannot be encrypted outside of TEE

EndBox enforces the routing of application traffic through TEEs deployed on untrusted client machines.

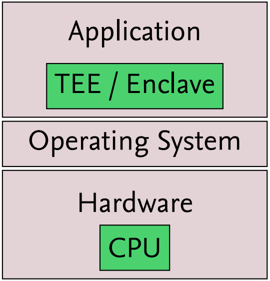
TEE: Intel SGX in a Nutshell

- x86 **instruction set extension** introduced with Skylake architecture
- Creation of **trusted execution environments** (TEEs) $\rightarrow$ **enclaves**
- Execution and data inside enclaves **protected from privileged software**
- Hardware-based **memory integrity protection and encryption**
- **Remote attestation** of enclaves
- Only CPU is **trusted**



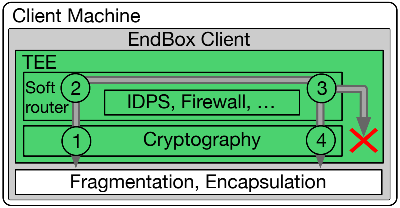
TEE: Intel SGX in a Nutshell

- x86 **instruction set extension** introduced with Skylake architecture
- Creation of **trusted execution environments** (TEEs) $\rightarrow$ **enclaves**
- Execution and data inside enclaves **protected from privileged software**
- Hardware-based **memory integrity protection and encryption**
- **Remote attestation** of enclaves
- Only CPU is **trusted**

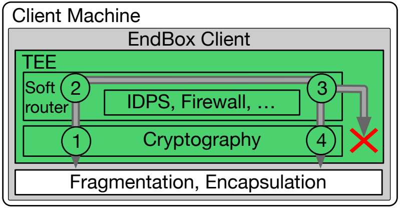


Intel SGX allows the creation of **enclaves**, trusted execution environments (TEEs) protected by hardware.

Implementation of EndBox Prototype

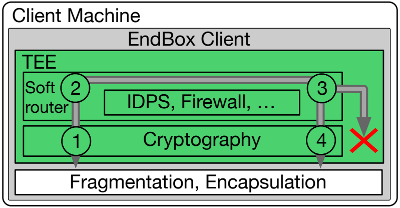


Implementation of EndBox Prototype



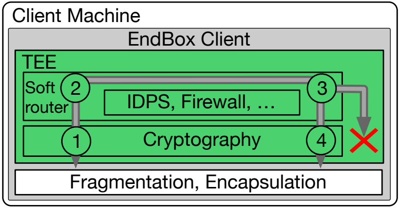
① Packet copied into enclave

Implementation of EndBox Prototype



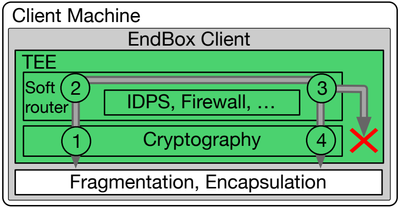
1. Packet copied into enclave
2. Execute middlebox function(s)

Implementation of EndBox Prototype



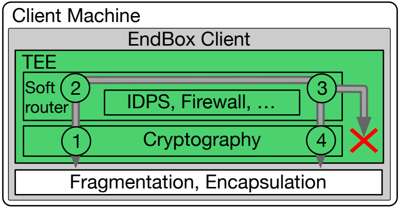
1. Packet copied into enclave
2. Execute middlebox function(s)
3. Packet accepted/discarded

Implementation of EndBox Prototype



1. Packet copied into enclave
2. Execute middlebox function(s)
3. Packet accepted/discarded
4. Packet signed, encrypted and copied out of enclave

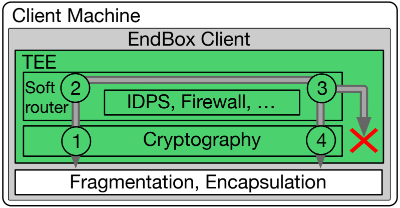
Implementation of EndBox Prototype



1. Packet copied into enclave
2. Execute middlebox function(s)
3. Packet accepted/discarded
4. Packet signed, encrypted and copied out of enclave

- Integration of enclaves into **OpenVPN client**
- Utilise **Click modular router** (Kohler et al. TOCS'00) for **arbitrary** middlebox functions
- TaLoS library (Aublin et al. technical report '17) for in-enclave **TLS termination**

Implementation of EndBox Prototype



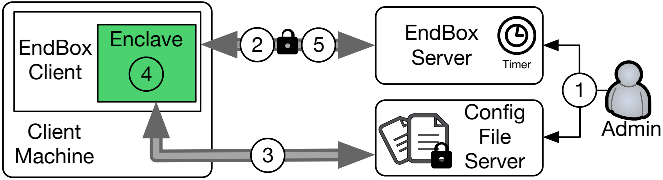
1. Packet copied into enclave
2. Execute middlebox function(s)
3. Packet accepted/discarded
4. Packet signed, encrypted and copied out of enclave

- Integration of enclaves into **OpenVPN client**
- Utilise **Click modular router** (Kohler et al. TOCS'00) for **arbitrary** middlebox functions
- TaLoS library (Aublin et al. technical report '17) for in-enclave **TLS termination**

EndBox executes middlebox functions inside **trusted SGX enclaves** embedded into a **VPN client** and uses the **Click modular router**.

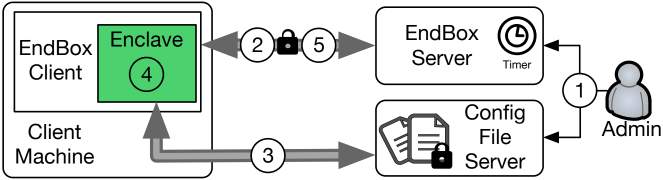
EndBox Configuration Updates

- Configuration updates are **challenging** with distributed middleboxes



EndBox Configuration Updates

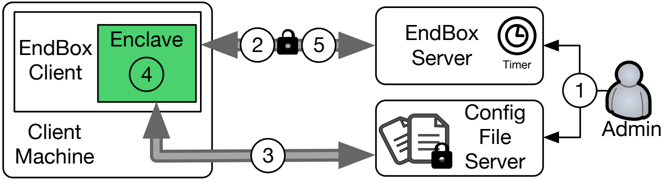
- Configuration updates are **challenging** with distributed middleboxes



① Admin uploads **encrypted configuration** and starts grace period timer

EndBox Configuration Updates

- Configuration updates are **challenging** with distributed middleboxes

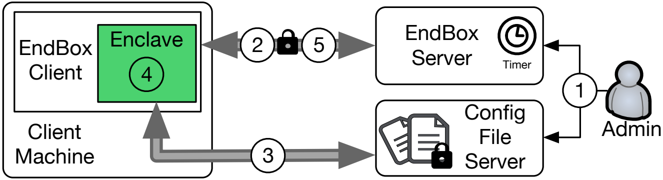


① Admin uploads **encrypted configuration** and starts grace period timer

② New version number piggybacked on **OpenVPN ping messages**

EndBox Configuration Updates

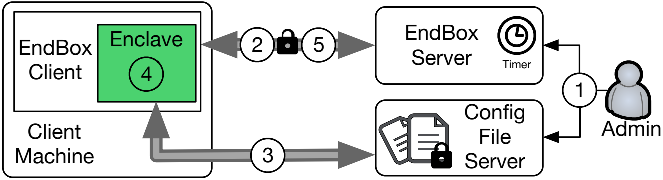
- Configuration updates are **challenging** with distributed middleboxes



① Admin uploads **encrypted configuration** and starts grace period timer
② New version number piggybacked on **OpenVPN ping messages**
③ If necessary, **client obtains new configuration** file

EndBox Configuration Updates

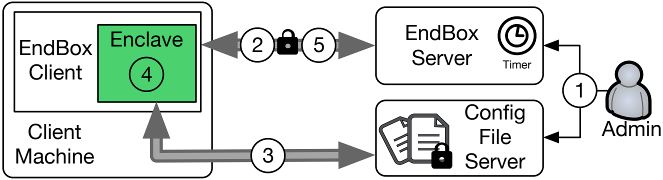
- Configuration updates are **challenging** with distributed middleboxes



1. Admin uploads **encrypted configuration** and starts grace period timer
2. New version number piggybacked on **OpenVPN ping messages**
3. If necessary, **client obtains new configuration** file
4. Configuration is **decrypted and applied**

EndBox Configuration Updates

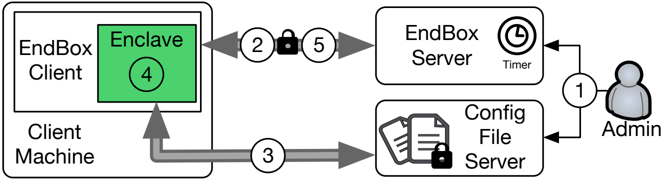
- Configuration updates are **challenging** with distributed middleboxes



1. Admin uploads **encrypted configuration** and starts grace period timer
2. New version number piggybacked on **OpenVPN ping messages**
3. If necessary, **client obtains new configuration** file
4. Configuration is **decrypted and applied**
5. Ping server with piggybacked version number to **prove application**

EndBox Configuration Updates

- Configuration updates are **challenging** with distributed middleboxes



① Admin uploads **encrypted configuration** and starts grace period timer
② New version number piggybacked on **OpenVPN ping messages**
③ If necessary, **client obtains new configuration** file
④ Configuration is **decrypted and applied**
⑤ Ping server with piggybacked version number to **prove application**

EndBox configurations are **centrally controlled and enforced.**

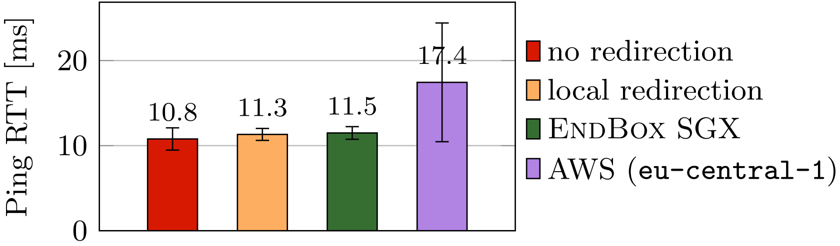
Outline

- Introduction to Middleboxes
- Design of EndBox
- **Evaluation of EndBox**
- Related Work
- Conclusion

Evaluation of EndBox

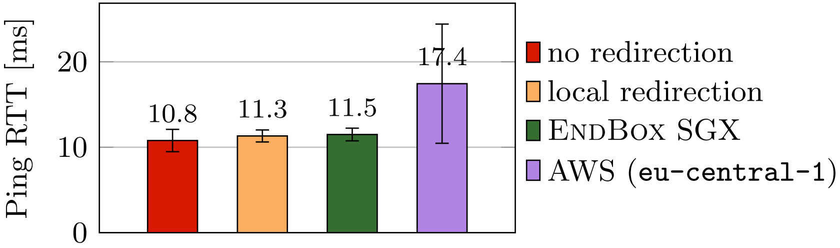
- **5 client machines** for executing many clients
  - SGX-capable 4-core Xeon v5 CPUs, 32GB RAM
- **2 server machines** as OpenVPN servers
  - non-SGX 4-core Xeon v2 CPUs, 16GB RAM
- **10 Gbps** interconnection (switched network)
- **Research questions**:
  - What is EndBox's **impact on latency**?
  - What **throughput** can EndBox achieve?
  - Does EndBox improve **scalability**?

Latency Depending on Middlebox Placement



Experiment	latency overhead
local redirection	4.6%
EndBox SGX	**6.5%**
AWS (Europe)	61%

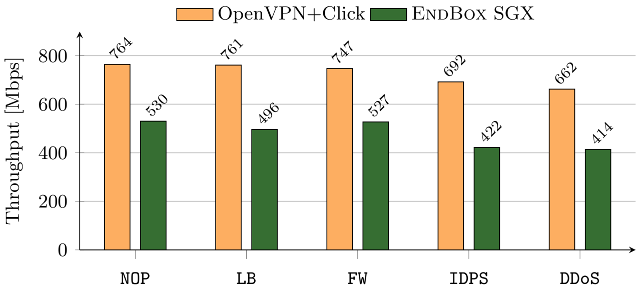
Latency Depending on Middlebox Placement



Experiment	latency overhead
local redirection	4.6%
EndBox SGX	**6.5%**
AWS (Europe)	61%

EndBox has **low impact on latency** compared to cloud-based solutions.

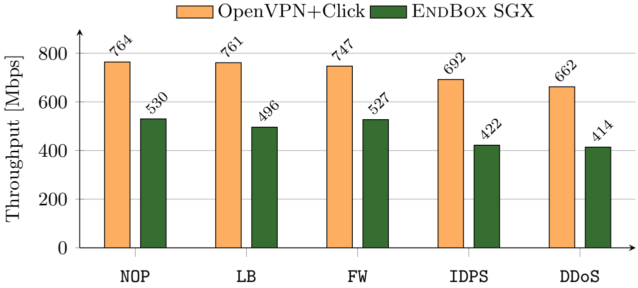
Throughput for Different Middlebox Use cases



packet size: 1500 bytes

Use case	throughput overhead
Forwarding (NOP)	30.6%
Load balancing (LB)	34.8%
Firewalling (FW)	29.5%
Intrusion prev. (IDPS)	39.0%
DDoS mitigation (DDoS)	37.5%

Throughput for Different Middlebox Use cases



packet size: 1500 bytes

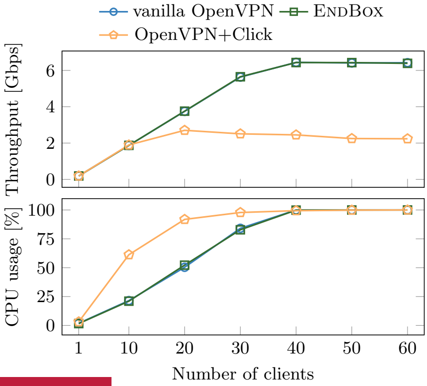
Use case	throughput overhead
Forwarding (NOP)	30.6%
Load balancing (LB)	34.8%
Firewalling (FW)	29.5%
Intrusion prev. (IDPS)	39.0%
DDoS mitigation (DDoS)	37.5%

EndBox has an **average throughput overhead** of 34,3% for multiple use cases.

Scalability on Server-side

Setup	Description
vanilla OpenVPN	unmodified OpenVPN version
EndBox	EndBox with SGX
OpenVPN+Click	OpenVPN and server-side Click instance

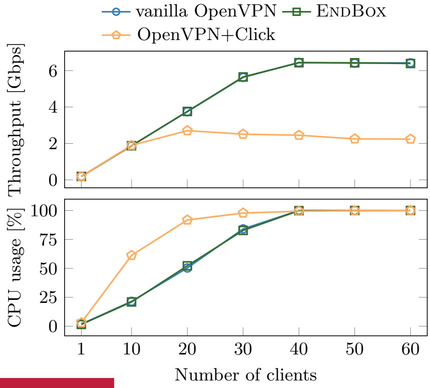
Clients generate a workload of 200 Mbps each



Scalability on Server-side

Setup	Description
vanilla OpenVPN	unmodified OpenVPN version
EndBox	EndBox with SGX
OpenVPN+Click	OpenVPN and server-side Click instance

Clients generate a workload of 200 Mbps each

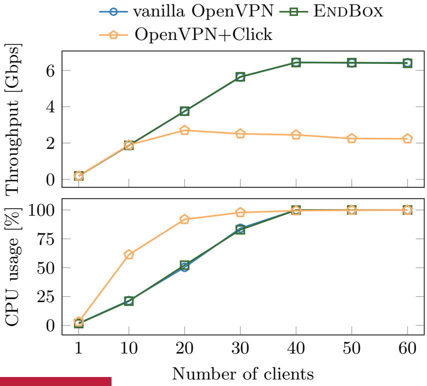


EndBox **scales linearly** with the number of clients.

Scalability on Server-side

Setup	Description
vanilla OpenVPN	unmodified OpenVPN version
EndBox	EndBox with SGX
OpenVPN+Click	OpenVPN and server-side Click instance

Clients generate a workload of 200 Mbps each



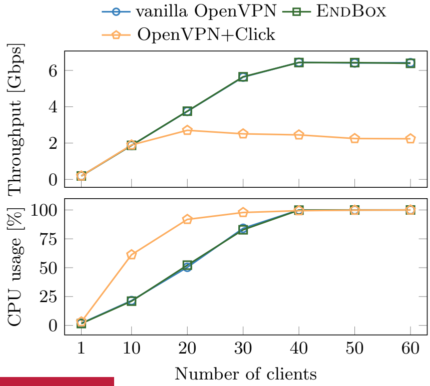
EndBox **scales linearly** with the number of clients.

EndBox has **no server-side performance penalty.**

Scalability on Server-side

Setup	Description
vanilla OpenVPN	unmodified OpenVPN version
EndBox	EndBox with SGX
OpenVPN+Click	OpenVPN and server-side Click instance

Clients generate a workload of 200 Mbps each



EndBox **scales linearly** with the number of clients.

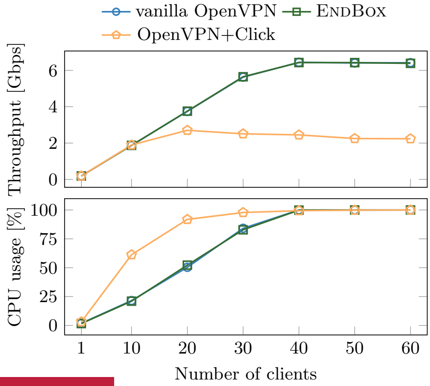
EndBox has **no server-side performance penalty.**

EndBox has a **3.8× higher throughput** compared to a traditional deployment.

Scalability on Server-side

Setup	Description
vanilla OpenVPN	unmodified OpenVPN version
EndBox	EndBox with SGX
OpenVPN+Click	OpenVPN and server-side Click instance

Clients generate a workload of 200 Mbps each



EndBox **scales linearly** with the number of clients.

EndBox has **no server-side performance penalty.**

EndBox has a **3.8× higher throughput** compared to a traditional deployment.

EndBox **saves resources** on server-side.

Outline

- Introduction to Middleboxes
- Design of EndBox
- Evaluation of EndBox
- **Related Work**
- Conclusion

Related Work

- Moving middlebox functions to clients has been proposed before
- **Trusted clients** assumed, exception: **ETTM** (Dixon et al. NSDI'11)
  - Based on **Trusted Platform Module** (TPM)
  - Large trusted computing base (TCB) includes hypervisor
  - **Paxos** applied for consensus → bad scalability
- Recent work uses SGX, but target **cloud-based trusted middleboxes**
  - **ShieldBox** (Trach et al. SOSR'18)
  - **SafeBricks** (Poddar et al. NSDI'18)

Related Work

- Moving middlebox functions to clients has been proposed before
- **Trusted clients** assumed, exception: **ETTM** (Dixon et al. NSDI'11)
  - Based on **Trusted Platform Module** (TPM)
  - Large trusted computing base (TCB) includes hypervisor
  - **Paxos** applied for consensus → bad scalability
- Recent work uses SGX, but target **cloud-based trusted middleboxes**
  - **ShieldBox** (Trach et al. SOSR'18)
  - **SafeBricks** (Poddar et al. NSDI'18)

EndBox is the first approach exploring the deployment of
client-side middleboxes with recent hardware trends like Intel SGX

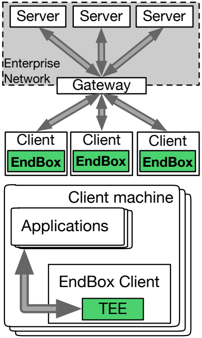
Outline

- Introduction to Middleboxes
- Design of EndBox
- Evaluation of EndBox
- Related Work
- **Conclusion**

Conclusion

EndBox's contributions:

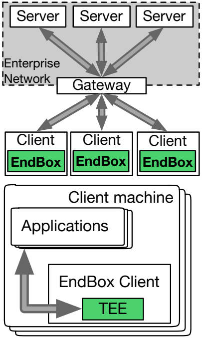
- **Secure deployment and execution** of middlebox functions on **untrusted client machines**
- **Scales linearly** with number of clients
- Up to **3.8× higher throughput**
- **Centrally controlled and enforced** configuration
- Secure **analysis of encrypted traffic** (see paper!)
- **Additional scenario:** ISP (see paper!)



Conclusion

EndBox's contributions:

- **Secure deployment and execution** of middlebox functions on **untrusted client machines**
- **Scales linearly** with number of clients
- Up to **3.8**$\times$ **higher throughput**
- **Centrally controlled and enforced** configuration
- Secure **analysis of encrypted traffic** (see paper!)
- **Additional scenario:** ISP (see paper!)



Thank you for your time! Questions?

goltzsche@ibr.cs.tu-bs.de
@d_goltzsche github.com/ibr-ds