Embark: Securely Outsourcing Middleboxes to the Cloud

Chang Lan, Justine Sherry, Raluca Ada Popa, Sylvia Ratnasamy, Zhi Liu

UC Berkeley
Tsinghua University
Background

➢ Middleboxes are prevalent and problematic
  ■ Number of Middleboxes ≈ Number of Routers (APLOMB [SIGCOMM ‘12])
  ■ Lots of Problems:
    ■ MB Manifesto [HotNets ‘11], CoMb [NSDI ‘12],
      Honda et al. [IMC’11], DOA [OSDI ’04], ETTM [NSDI ’11], ...

➢ A Promising Solution: Outsourcing
  ■ APLOMB [SIGCOMM ‘12]
  ■ Aryaka, Zscaler
  ■ AT&T NFV/CORD
New Challenge: Confidentiality and Privacy

➢ The middleboxes sees the traffic unencrypted.

➢ Strawman: End-to-end Encryption (e.g. TLS):
  ■ Some middleboxes cannot process traffic (e.g. Deep Packet Inspection).
  ■ Unencrypted packet fields still leak information
Even with end-to-end encryption, Cloud can still infer the user profile.
Problem Statement

Can we outsource middleboxes without compromising privacy?

Embark

the first system that allows middlebox outsourcing, while keeping traffic confidential.
Overview

➢ Approach
  ■ Middleboxes process encrypted traffic without decrypting it

➢ Crypto Primitives
  ■ **KeywordMatch**: For Signature Matching
    ■ BlindBox [SIGCOMM ‘15]: Prohibitive Setup Time Per Flow
    Contribution: System Design + Implementation without Per-flow Setup Time
  ■ **PrefixMatch**: Prefix/Range Matching
    Contribution: A fast, secure encryption scheme for prefix matching
Overview

➢ Approach

■ Middleboxes process encrypted traffic without decrypting it

➢ Crypto Primitives

■ KeywordMatch: For Signature Matching

■ BlindBox [SIGCOMM ‘15]: Prohibitive Setup Time Per Flow

  Contribution: System Design + Implementation without Per-flow Setup Time

■ PrefixMatch: Prefix/Range Matching

  Contribution: A fast, secure encryption scheme for prefix matching
Outline

1. Service Model of Embark
2. PrefixMatch: Two Functions
   - EncryptRanges
   - EncryptValue
3. Evaluation
4. Conclusion
Service Model

Enterprise → Cloud
Service Model

Enterprise

Gateway
Encrypt / Decrypt traffic to/from the cloud

Cloud
Service Model

Middlebox Rules

IP firewall rules, IDS signatures, etc.

Enterprise

Cloud
Initialization

Enterprise encrypt rules using \textit{EncryptRanges}.
Initialization

Enterprise

Middleboxes deploy encrypted rules.
Packet Flow

1. Outgoing traffic are sent to Gateway.
Packet Flow

2. Encrypt the traffic
   - Encrypt packet headers field by field using `EncryptValue`
   - Encrypt payloads using stream cipher

Implication: no change to packet structure
Packet Flow

3. Forward to Cloud
Packet Flow

4. Middleboxes process encrypted traffic.

No change to algorithms: E.g., LPM, multi-dimensional classifiers, etc.
Packet Flow

5. Back to Gateway
Packet Flow

6. Decrypt and Forward
Outline

1. Service Model of Embark
2. PrefixMatch: Two Functions
   - EncryptRanges
   - EncryptValue
3. Evaluation
4. Conclusion
PrefixMatch

➢ Property

■ Answer if a value $V$ matches a range $R_i$ from $[R_1, R_2, ...]$

➢ Security

■ Do not reveal the value of $V$ and $R_i$
■ If both $V_1$ and $V_2$ match $R_i$, do not reveal the ordering between $V_1$ and $V_2$
PrefixMatch vs. OPE

➢ Order-preserving Encryption
  ■ Preserve the ordering of values after encryption

➢ PrefixMatch is better than OPE in this scenario
  ■ More secure (No relative ordering)
  ■ Faster (10000x)
    ■ Compare with the state-of-the-art OPE schemes (BCLO and mOPE)

<table>
<thead>
<tr>
<th>Operation</th>
<th>BCLO</th>
<th>mOPE</th>
<th>PrefixMatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt, 10K rules</td>
<td>9333 us</td>
<td>6640 us</td>
<td>0.53 us</td>
</tr>
<tr>
<td>Encrypt, 100K rules</td>
<td>9333 us</td>
<td>8300 us</td>
<td>0.77 us</td>
</tr>
<tr>
<td>Decrypt</td>
<td>169 us</td>
<td>0.128 us</td>
<td>0.128 us</td>
</tr>
</tbody>
</table>
EncryptRanges

➢ Firewall Rules

block from 192.168.1.0/24 to 205.203.224.0/19
block from 192.168.0.0/16 to 223.254.0.0/16
block from 10.1.0.0/16 to 223.201.0.0/16
EncryptRanges

Assign Random Prefixes
EncryptRanges

192.168.1.0/24 -> 3.0.0.0/8
192.168.0.0/16 -> 3.0.0.0/8
162.0.0.0/8
10.1.0.0/16    -> 62.0.0.0/8
EncryptRanges

Source IP
192.168.1.0/24 -> 3.0.0.0/8
192.168.0.0/16 -> 3.0.0.0/8
10.1.0.0/16 -> 62.0.0.0/8

Destination IP
205.203.224.0/19 -> 12.0.0.0/8
223.254.0.0/16 -> 241.0.0.0/8
223.201.0.0/16 -> 163.0.0.0/8
Encrypt each field independently

- Source IP, Destination IP,
  Source Port, Destination Port...
Encrypt each field independently

- Source IP, Destination IP,
  Source Port, Destination Port...
Src IP = 10.1.1.1

10.1.0.0/16

192.168.1.0/24

192.168.0.0/16

3.0.0.0/8

62.0.0.0/8

162.0.0.0/8
Src IP = 10.1.123.123

Enc (Src IP) = 62.0.0.0 + Rand(0, 2^24)
Problem 1: How to support NAT and Load Balancers?

- **Deterministic:** The value from the same flow will be mapped to the same value
- **Injective:** Values from different flows will be mapped to different values
- **Sufficient condition**

<table>
<thead>
<tr>
<th>Sufficient condition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let</td>
</tr>
<tr>
<td>( v = (sip, dip, sp, dp, proto) )</td>
</tr>
<tr>
<td>( v' = (sip', dip', sp', dp', proto') )</td>
</tr>
<tr>
<td>( v = v' ) if and only if</td>
</tr>
<tr>
<td>( Enc(v) = Enc(v') )</td>
</tr>
</tbody>
</table>

**Src IP = 10.1.123.123**

\[ Enc (Src IP) = 62.0.0.0 + \text{Rand}(0, 2^{24}) \]
Problem 1: How to support NAT and Load Balancers?

- Use pseudorandom function, seeded by 5-tuple
- Use IPv6 to avoid collisions

\[
\text{Src IP} = 10.1.123.123 \\
\text{Enc (Src IP)} = 62.0.0.0 + \text{Rand}(0, 2^{24}) \\
\text{Src IP} = ::FFFF:10.1.123.123 \\
\text{Enc (Src IP)} = 3e00::/8 + \text{PRF(Src IP)}
\]
Problem 1: How to support NAT and Load Balancers?

Problem 2: How to decrypt?

- Store AES(Source IP) in IP Options
- Decrypt AES(Source IP)
Outline

1. Service Model of Embark
2. PrefixMatch: Two Functions
   - EncryptRanges
   - EncryptValue
3. Evaluation
4. Conclusion
Evaluation

➢ What kinds of middleboxes does Embark support?
  ■ Performance of each type of middleboxes

➢ How much does PrefixMatch increase the number of rules?

➢ Microbenchmarks
  ■ How does PrefixMatch compare with OPE?
  ■ How well does PrefixMatch scale with the number of rules?

➢ Performance
  ■ How fast is the gateway (with PrefixMatch and with KeywordMatch)
  ■ How much does the service model increase the page load time?
Supported Middleboxes

<table>
<thead>
<tr>
<th>Supported Middleboxes</th>
<th>Linux iptables</th>
<th>PrefixMatch</th>
<th>ECMP</th>
<th>HAProxy</th>
<th>Embark vs Squid</th>
<th>Embark vs Snort</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Firewall</td>
<td>Linux iptables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAT</td>
<td>Linux iptables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3 Load Balancer</td>
<td>ECMP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4 Load Balancer</td>
<td>HAProxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP Proxy</td>
<td>Embark vs Squid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parental Filter</td>
<td>Embark vs Squid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intrusion Detection</td>
<td>Embark vs Snort</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(excluding scripts and other statistical techniques)
How much does PrefixMatch increase Firewall rules?

➢ Upper bound
  - $O(n^d)$, $d$ is the number of fields

➢ Empirically
  - Rulesets
    - 3 firewall rulesets from campus network at UC Berkeley
    - 1 firewall ruleset from Emerging Threats
  - Result
    - UCB rulesets: No increase
    - Emerging Threats: from 1363 to 1370
  - Intuition
    - Most firewall rules don’t overlap
How fast is the gateway (without KeywordMatch)?

With KeywordMatch enabled:

240 Mbps per core
(Pkt size: 1400 B)

Performance with 1k rules:
- 1.2 Gbps (min-size)
- Line rate (other cases)
See the paper for ...

- How we design and implement middleboxes
- Formal proof of sufficient conditions for NAT and L3/TCP Load Balancers
- Limitations
- More in-depth evaluation
- ...

39
Conclusion

Middleboxes can be outsourced in a way that still keeps the traffic confidential with Embark.

Paper: changlan.org/papers/embark.pdf

Contact: clan@eecs.berkeley.edu

Thanks!
Related work

Middlebox Outsourcing:
- Making Middleboxes Someone Else’s Problem. SIGCOMM 2012

Data Confidentiality
- Shielding Applications from an Untrusted Cloud with Haven, OSDI 2014
- Hails: Protecting Data Privacy in Untrusted Web Applications, OSDI 2012
- Building Web Applications on Top of Encrypted Data Using Mylar, NSDI 2014
- CryptDB: Protecting Confidentiality with Encrypted Query Processing, SOSP 2011
- BlindBox: Deep Packet Inspection over Encrypted Traffic, SIGCOMM 2015
- Multi-Context TLS (mcTLS): Enabling Secure In-Network Functionality in TLS, SIGCOMM 2015

Trace Anonymization and Inference
- A High-level Programming Environment for Packet Trace Anonymization and Transformation, SIGCOMM 2003

Encryption Schemes
- Order preserving encryption for numeric data, SIGMOD 2004
- Order-Preserving Symmetric Encryption, EUROCRYPT 2009
- An Ideal-Security Protocol for Order-Preserving Encoding, Oakland 2013
- Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys, EUROCRYPT 2016
How much does the service model inflate the page load time? (Alexa Top-500)

**Median-case increase**

**ISP/Central Office**: < 50ms

**CDN**: < 100 ms

**Public Cloud (EC2)**: < 720 ms
How does PrefixMatch scale with # of rules?

Single-core Performance of PrefixMatch
Packet size: 64 B
KeywordMatch

➢ Primitive for Deep Packet Inspection
  ▪ Detect if a keyword presents in a bytestream

➢ Searchable Encryption
  ▪ Leverage the scheme in BlindBox (SIGCOMM ‘15)

➢ Reconstruct TCP Bytestream at the Gateway
  ▪ Middleboxes receive bytestreams from a separate, reliable channel
  ▪ Process packet flows based on bytestream content