BlindBox: Deep Packet Inspection Over Encrypted Traffic

Raluc...
Deep Packet Inspection (DPI)

In-network devices which inspect packet payloads to enforce policies.

Intrusion detection/prevention

Exfiltration

Parental filtering
Example: Intrusion prevention

Alice

middlebox

Bob

Detect!

rule generator

rules

ATTACK

HACKS

183237

McAfee

Detect!
Observation: a lot of traffic today is sent over https
Problem: middleboxes cannot inspect traffic over https
State-of-the-art: man in the middle attack on SSL [Huang et al.’14]

No privacy!
and a lot of other security issues [Jarmoc’12]
Can we achieve both privacy and payload inspection?
The first system to enable DPI middleboxes to inspect traffic **without seeing the traffic**
Approach

inspect encrypted traffic without decrypting it!
Technical setup
Threat model

One endpoint can misbehave but at least one endpoint behaves correctly; endpoints cannot learn rules.

McAfee® rule generator

generates rules correctly

run detection functionality correctly

but curious to see traffic content (honest but curious/passive)
Goals

1. Strong privacy guarantee
2. Practical
3. Wide range of functionality

- well-studied security guarantees
- network rates are incredibly fast! (microsec)
- regexp, scripts
Strawmen:
fully homomorphic or functional encryption?

[Gentry'09, BSW'11]

• Does not fit our threat model

• Prohibitive performance: IDS detection over a single packet requires over 1 day* NO NO NO

* based on our experiments using [Katz, Sahai, Waters'08]
BlindBox’s design
System overview

**BBhttps**: enhance https

- **BBhttps**
  - SSL encrypt
- SSL traffic
- **BBhttps**
  - SSL decrypt

- BlindBox encrypt
  - split in tokens
  - encrypt each token

encrypted tokens

- run detection on encrypted tokens

SSL remains unchanged.
How do we inspect encrypted traffic efficiently?
## Step I: searchable encryption

<table>
<thead>
<tr>
<th>scheme</th>
<th>packet inspection</th>
<th>security</th>
</tr>
</thead>
<tbody>
<tr>
<td>deterministic schemes</td>
<td>fast (O(\log(#\text{rules}) \times #\text{bytes/packet}))</td>
<td>weak</td>
</tr>
<tr>
<td>randomized schemes [SongWagnerPerrig'00]</td>
<td>slow (O(#\text{rules} \times #\text{bytes/packet}))</td>
<td>high</td>
</tr>
</tbody>
</table>

Desired: fast & high security

No satisfying scheme for our setting.

---

Our new searchable encryption scheme & detection algorithm
Our new search scheme

example message

BBhttps → tokens

salt, AES

(salt)

AES

AES

AESk(rule)

Fast encryption: only AES!

But detection is slow $O(#\text{rules} \times #\text{tokens})$
Desired:

Avoid combining a salt with each rule

Build index on rules
Fast detection protocol

Use a salt schedule:

<table>
<thead>
<tr>
<th>tokens</th>
<th>encrypted tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Enc_K(1, A)</td>
</tr>
<tr>
<td>B</td>
<td>Enc_K(1, B)</td>
</tr>
<tr>
<td>A</td>
<td>Enc_K(2, A)</td>
</tr>
<tr>
<td>A</td>
<td>Enc_K(3, A)</td>
</tr>
</tbody>
</table>

Rule index:
precompute tree with salt=1
Enc_K(1, rule1),
Enc_K(1, rule2), ...
Enc_K(2, rule2)
rare operation!

For each token, one tree lookup!
\( O(\log \text{#rules} \ast \text{# tokens}) \)
But how does the middlebox obtain $\text{AES}_K(\text{rule})$?
Step 2: BlindBox’s setup phase

1. Alice sends to MB garbled circuits [Yao’86] for $\text{AES}_K()$
2. MB runs oblivious transfer [Rabin’81] with Alice to obtain encoding for rule
3. MB evaluates garbled circuit and obtains $\text{AES}_K(\text{rule})$
Security guarantee

“Principle of least privilege”: the middlebox learns only byte positions where a rule matches (well-studied guarantee in the searchable encryption literature)
So far...

complete system for equality matching

Need support for regular expressions!
How to support regular expressions?
Rules with regular expressions in IPS

Snort rule example:

```
{ content: "malicious string",
  pcre: "\r\nHost\x3As+[\^\r\n]*?[bcdfghjklmnpqrstvwxyz]{5,}[\^\r\n]*?\xEbiz Hi",
  [...] }
```

• Rule first matches string by equality
• Must be highly selective string
New privacy model: probable cause privacy

Privacy may be lost only if there is a probable cause.

If a malicious string matches a packet, middlebox can decrypt the packet, but not otherwise.
New encryption scheme for probable cause

If token = rule, middlebox obtains $K$
Middlebox can run regexp

Rule

```
{content: "malicious string",
pcre: "\r\nHost\…",
[...]
}
```

Step 1: match content string on encrypted traffic

obtain $K$

decrypt packet

Step 2: run regexp on unencrypted traffic
More details in our paper!

- Optimizations to reduce bandwidth overhead
- Details on garbled circuits + oblivious transfer
- Support for malicious middlebox
- Rule generation, regular expressions, probable cause decryption…
Implementation

- Endpoints: BBhttps - C library
- Middlebox: Click framework
Evaluation
functionality
performance
# Functionality Evaluation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Without probable cause</th>
<th>With probable cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document watermarking</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Parental filtering</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Snort community (HTTP)</td>
<td>67%</td>
<td>100%</td>
</tr>
<tr>
<td>Snort Emerging Threats (HTTP)</td>
<td>42%</td>
<td>100%</td>
</tr>
<tr>
<td>StoneSoft (McAfee) IDS</td>
<td>40%</td>
<td>100%</td>
</tr>
<tr>
<td>LastLine IDS</td>
<td>29%</td>
<td>100%</td>
</tr>
</tbody>
</table>

IDS
Performance highlights

• Three main performance figures:

• Detection Time: competitive with existing IDSes!
  • 186Mbps with BlindBox (comparable to Snort)

• Transmission Time: practical overhead
  • Page load completion time increases by 0.15-1x

• Setup Time: not yet competitive
  • 1 min for 3000 rules
    • fine for long-lived connections (cloud-enterprise)
Upcoming work: MBArk

Outsourcing middleboxes to the cloud

Support header-based computations over encrypted data: firewall, NAT, IP forwarder, load balancer, VPN, IDS, exfiltration
Conclusion

BlindBox is the first system to enable DPI middleboxes to process traffic **without seeing it**

- practical for a class of applications

Thanks!