BlindIDS: Market-Compliant and Privacy-Friendly Intrusion Detection System over Encrypted Traffic

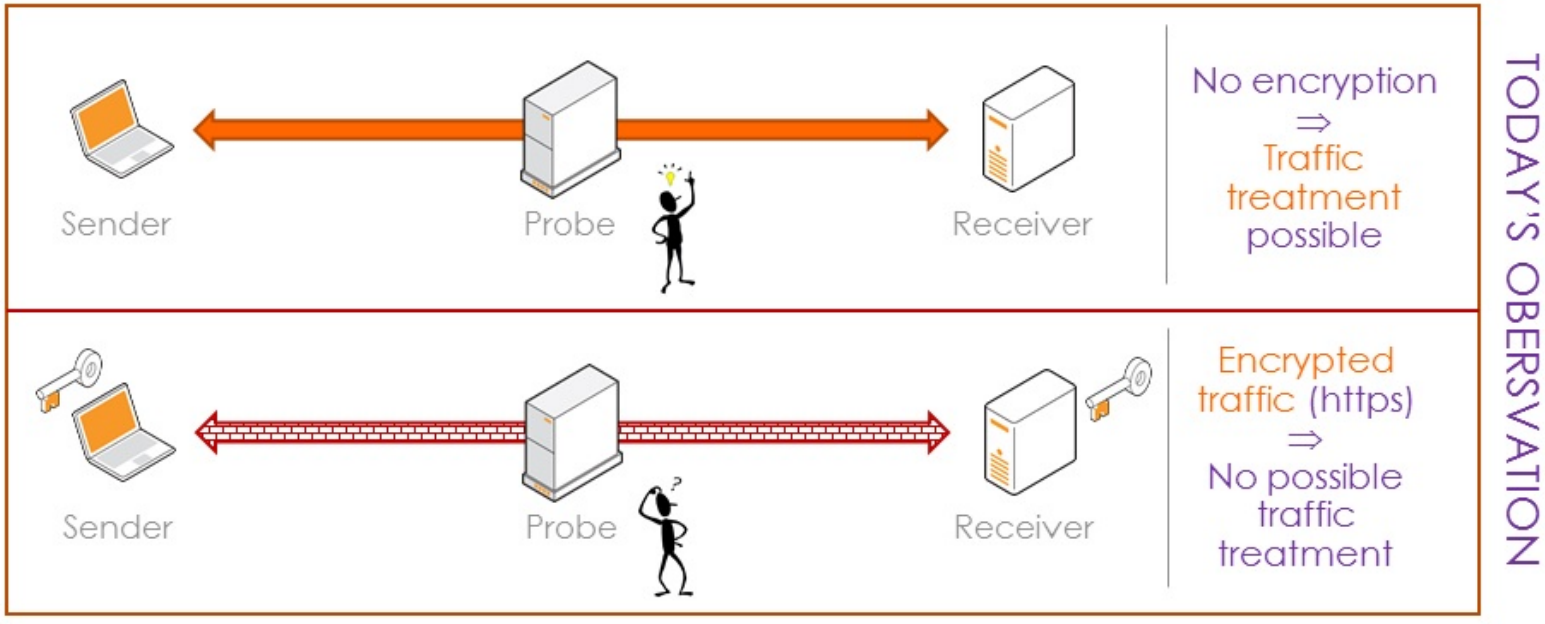
Sébastien Canard (Orange)
Joint work with Aïda Diop (Orange, Telecom SudParis), Nizar Kheir (Thales), Marie Paindavoine (Orange) and Mohamed Sabt (IRISA)

CAEN 2018, June, 2018

Encryption is our future

- IETF HTTPbis working group that is in charge of designing the next generation http 2.0 specification proposes that encryption be the default way data is transferred over the open Internet
- According to a joint study by Ponemon institute, along with Thales and Vormetric Data Security, encrypted Internet traffic has grown up from 15% of world-wide traffic in 2005 until up to 40% in 2015. The proportion of encrypted Internet traffic is expected to reach up to 80% by 2020
- OTTs are moving forward towards full end-to-end encryption, including recent example such as whatsapp, Google both for end-to-end email encryption and for Internet browsing, etc.
- European Community, through its Horizon H2020 program, and in particular the joint cPPP on cybersecurity, is advocating for more privacy guarantees in terms of traffic encryption for end users
- ...

Confidentiality $\Longrightarrow$ full security?

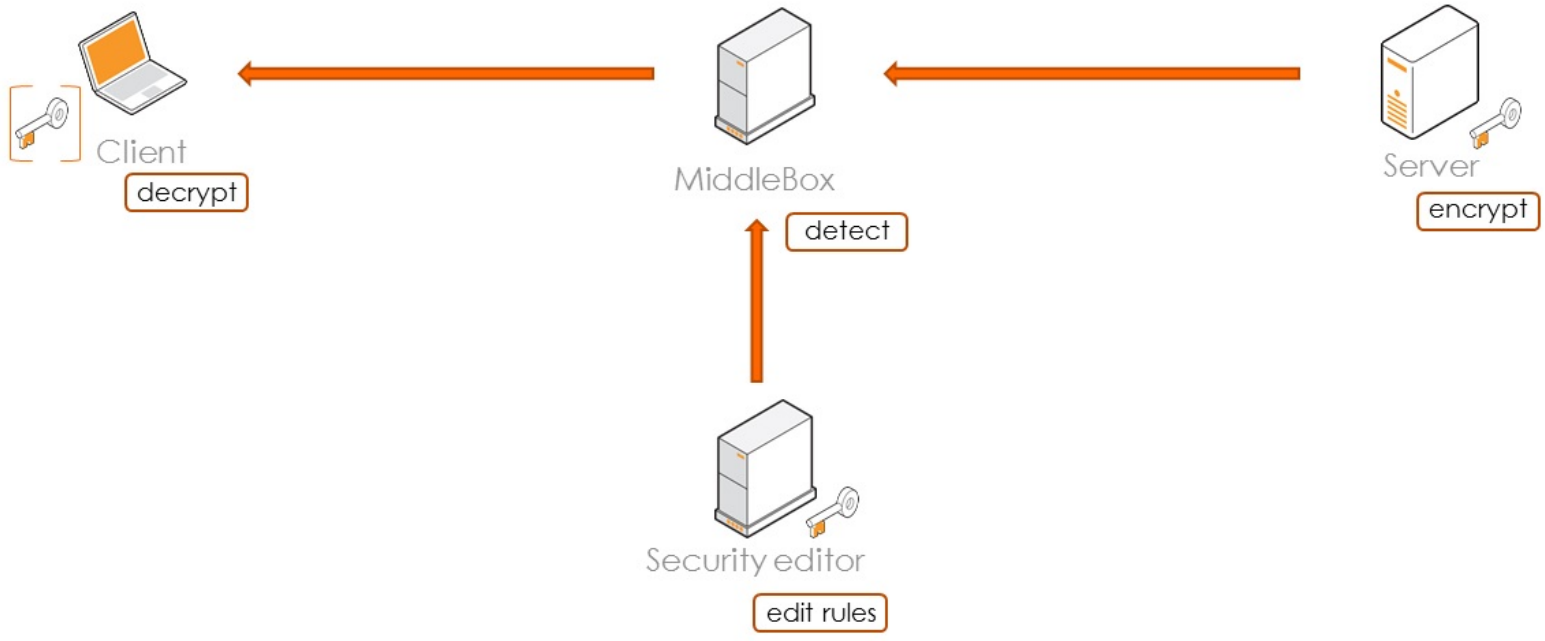


With current standards, difficult choice between
data confidentiality and usability/security!!

Impacted use cases

- Parental control over the traffic
- Security Information and Event Management
- Detecting compromising SSH requests
- Quality service probes
- Intrusion Detection Systems (IDS)
- ...

Architecture



- Deep Packet Inspection on the content of the packet
- Use detection rules to analyse the content of the traffic
  - Behavior-based detection: mostly done on meta-data that are not encrypted (CISCO approach)
  - Signature-based detection: intrusions detection using signatures

$\Longrightarrow$ How to manage an encrypted traffic?

Agenda

- Security model
- Cryptographic components
- Implementation and results
- Conclusion

Security model

Requirements and assumptions

- High level properties
  - Privacy-friendly: no access is possible to the clear-text content of encrypted traffic
  - Security-aware: it supports DPI over encrypted traffic
  - Practical: achieving both good performances and real-world market requirements (including rule secrecy, know-how of the Security editor)
- Assumptions on players
  - MiddleBox is honest-but-curious on both the traffic and the rules
  - Collusion between MiddleBox and Security editor cannot be handled, due to dictionary attack
  - Collusion between Client and Server cannot be handled, due to over-encryption possibility (as in a non-encrypted form!!)

Detection property

- Any malicious traffic (that is a traffic considered as malicious when not encrypted) must be detected by the MiddleBox

Experiment $\mathsf{Exp}^{det}_{\pi,\mathcal{A}}(\lambda)$

$(\mathsf{param}, \mathsf{sk}_{\mathsf{SE}}, \mathsf{sk}_{\mathsf{R}}) \leftarrow \mathsf{Setup}(1^{\lambda})$;
$\mathcal{B} \leftarrow \mathsf{RuleGen}(\mathsf{param}, \mathsf{sk}_{\mathsf{SE}}, \mathcal{R})$;
$E \leftarrow \mathcal{A}(1^{\lambda}, \mathsf{param})$;
if $\mathsf{Detect}(\mathsf{param}, E, \mathcal{B}) = 1$, then return 0;
$T \leftarrow \mathsf{Decrypt}(\mathsf{param}, \mathsf{sk}_{\mathsf{R}}, E)$;
if $\mathsf{Detect}(T, \mathcal{R}) = 0$, then return 0;
return 1.

Traffic indistinguishability

- It is not feasible for the MiddleBox to learn any information about the traffic, other than it is malicious or safe

Experiment $\mathsf{Exp}^{tr-ind}_{\pi,\mathcal{A}}(\lambda)$

$b \leftarrow \{0,1\}$;
$(\mathsf{param}, \mathsf{sk}_{\mathsf{SE}}, \mathsf{sk}_{\mathsf{R}}) \leftarrow \mathsf{Setup}(1^\lambda)$;
$T_0, T_1 \leftarrow \mathcal{A}(1^\lambda, \mathsf{param})$;
if $\mathsf{type}(T_0, T_1) = 0$, return 0;
$E_b \leftarrow \mathsf{Encrypt}(\mathsf{param}, T_b)$;
$b' \leftarrow \mathcal{A}(E_b)$;
return $(b = b')$.

Definition (Traffic Type)

Let T_0 and T_1 be two traffics and let $\mathcal{R}$ be a set of rules. We say that T_0 and T_1 are of the *same type*, denoted $\mathsf{type}(T_0, T_1) = 1$, iff $\mathsf{Detect}(\mathsf{param}, T_0, \mathcal{R}) = \mathsf{Detect}(\mathsf{param}, T_1, \mathcal{R})$, including the auxiliary information aux.

Rule indistinguishability

- It is not feasible for the MiddleBox to learn any information about the rules

Experiment $\mathsf{Exp}^{rul-ind}_{\pi,\mathcal{A}}(\lambda)$

$b \leftarrow \{0,1\}$;
$(\mathsf{param}, \mathsf{sk}_{\mathsf{SE}}, \mathsf{sk}_{\mathsf{R}}) \leftarrow \mathsf{Setup}(1^\lambda)$;
$\mathcal{R}_0, \mathcal{R}_1 \leftarrow \mathcal{A}_f(1^\lambda, \mathsf{param})$;
$\mathcal{B}_b \leftarrow \mathsf{RuleGen}(\mathsf{param}, \mathsf{sk}_{\mathsf{SE}}, \mathcal{R}_b)$;
$b' \leftarrow \mathcal{A}_g(\mathsf{sk}_{\mathsf{R}}, \mathcal{B}_b)$;
return $(b = b')$.

Definition (Min-entropy)

A probabilistic adversary $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ has *min-entropy* μ if $\forall \lambda \in \mathbb{N}$, $\forall r \in \mathcal{R}$: $\Pr\left[r' \leftarrow \mathcal{A}_f(1^\lambda, b) \ : \ r' = r\right] \leq 2^{-\mu(\lambda)}$. $\mathcal{A}$ is said to have *high min-entropy* if it has min-entropy μ with $\mu(\lambda) \in \omega(\log \lambda)$.

Cryptographic components

Signature-based detection

- Simple use case based on SQL injection
- Other use cases work similarly
- Example

```
http://localhost:9080/login?username=seb&password=1234' or a' = a'
```

- Example of rule

```
alert tcp any any -> HOMENET PORTHTTP (msg: "SQL Injection Attempt - or a=a"; content: "GET"; httpmethod; uricontent: or a' = a; nocase; classtype:web-application-attack; sid:3000001; rev:1;)
```

- The idea is then to search for a specific pattern inside the message
  - simple case: pattern matching
  - complex case: regular expression
- How to proceed if the traffic is encrypted?

Requirements on encryption

- Server performs encryption and client performs decryption
- MiddleBox performs matching
  - Taking as input an encrypted traffic and a pattern

  $\Rightarrow$ We need an encryption scheme with searchable capacity
- But the pattern should not be known to the MiddleBox
  - Due to the rule indistinguishability property

  $\Rightarrow$ We need trapdoor-based searchable encryption

  $\Rightarrow$ Given T_w and Encrypt(w'), test whether $w = w'$ or not

Decryptable searchable encryption (i)

- Based on a work by Fuhr and Paillier 2007
- F, G, H be three hash functions
- $(q, \mathbb{G}_1, g_1, \mathbb{G}_2, g_2, \mathbb{G}_t, e(.,.))$ be a bilinear environment
- Security editor generates $\mathsf{tk} = x' \leftarrow \mathbb{Z}_q$ and publishes $\mathsf{pk}_{\mathsf{SE}} = g_1^{x'}$ and $a \in \mathbb{Z}_q^*$
- Receiver generates $\mathsf{sk}_R = x \leftarrow \mathbb{Z}_q$ and publishes $\widetilde{\mathsf{pk}}_{\mathsf{R}} = g_1^x$
- Key independence between $\mathsf{pk}_{\mathsf{SE}}$ and $\widetilde{\mathsf{pk}}_{\mathsf{R}}$

Decryptable searchable encryption (ii)

- Rule generation: for any word w_i, computes $T_i = F(w_i)^{x'}$
- Traffic encryption: for each token t_i in the traffic, computes

$$\begin{aligned} c_{1,i} &= g_1^{r_i}; \\ (s_1, s_2)_i &= G(\widetilde{\mathrm{pk}_{\mathrm{R}}^{r_i}}); \\ c_{2,i} &= s_{1,i} \oplus t_i; \\ c_{3,i} &= g_1^{s_{2,i}}; \\ u_i &= e(\mathrm{pk}_{\mathrm{SE}}^{s_{2,i}}, F(t_i)); \\ c_{4,i} &= H(u_i) + a \mod q. \end{aligned}$$

- Detection: computes $u_i = e(c_{3,i}, T_j)$ and $a' = c_{4,i} - H(u_i) \mod q$. If $a \neq a'$, then the token is safe.
- Traffic decryption: for each ciphertext, computes

$$\begin{aligned} (s_1, s_2)_i &= G(c_{1,i}^x); \\ t_i &= c_{2,i} \oplus s_{1,i} \end{aligned}$$

Obtained security

- The scheme is detectable provided that there is no collision in the trapdoor generation function
- The scheme is traffic-indistinguishable under the CDH and the GDDHE assumptions in the random oracle model
- The scheme is rule-indistinguishable for rules of high min-entropy, in the random oracle model

- GDDHE assumption: given polynomials P, Q, f and given $H(x_1, \ldots, x_n) = (g_1^{P(x_1,\ldots,x_n)}, g_2^{Q(x_1,\ldots,x_n)}) \in \mathbb{G}_1^s \times \mathbb{G}_2^s$ and $T \in \mathbb{G}_T$, a probabilistic polynomial-time adversary has a negligible probability to successfully decide if $T = e(g_1, g_2)^{f(x_1,\ldots,x_n)}$.

Implementation and results

Details about the implementation

- Encrypted pattern matching implies exact pattern matching
  - Sliding window: every character is encrypted multiple times
  - Delimiter-based: rules and traffic are split according to specified symbols
- Implemented in Java 8, using the Herumi library in C for pairings
- Intel(R) Xeon(R) with a E5-1620 CPU with 4 cores running at 3.70GHz under a 64-bit Linux OS

Obtain performances

- % of detected rules: 75% (only matching)
- Client time: 600 μs for each token
- Server time: 700 μs for each token
- Detection time: 700 μs for each couple (token,rule)

$\Longrightarrow$ 70 s for 3K rules and 1.5KB packet

- Traffic expansion ($|C|/|M|$): 7

Conclusion and perspectives

- A new solution for intrusions detection over encrypted traffic
- Formalization of a security model
- Better performances than BlindBox [Sherry et al., SIGCOMM 2015]
  - Quite similar detection time
  - Better RAM usage: 0.5 MB RAM used vs 512 GB for 100 parallel connections
  - Enough for a practical usage...?
- Managing regular expressions (in submission by IRISA and Orange)

- More efficient DSE? Symmetric cryptography?
- Better tokenisation?
- Additional properties? Forward Secrecy?

thank you