GASP : *a Generic Approach to Secure Protocols*

- Projet ANR (Jeune Chercheur)
- 4 ans (2019-10-01 – 2023-09-30)

Les *parsers* et machines à états des protocoles sont complexes et mènent à de nombreuses vulnérabilités

Et si la solution était d'écrire moins de code ? Est-ce possible avec les spécifications actuelles ?

Axes de recherche

- *Network protocol observation in the field*
- *Protocol description to derive reference implementation*
- *Tests on existing implementations using a grey- or whitebox approach*

Description des messages et des automates (1/2)

Analyse des outils de génération de *parsers* existants

Sample	Category	hammer	nail	nail-pictyeye	netzob	nom	p2	parsifal	recordflux
dns-answer-A.bin	good	good	bad	good	-	good	-	good	-
dns-answer-A_PTR_c012_afterDN.bin	good	good	bad	good	-	good	-	good	-
dns-answer-scapy_PTR.bin	good	good	good	good	-	good	-	good	-
dns-request-scapy_PTR.bin	good	good	good	good	-	good	-	good	-
dns-simple-answer-A.bin	good	good	good	good	-	good	-	good	-
dns-simple-answer-A_PTR_c012_afterDN.bin	good	good	good	good	-	good	-	good	-
packet.0	good	good	good	good	-	good	-	good	-
packet.1	good	good	good	good	-	good	-	good	-
packet.2	good	good	good	good	-	good	-	good	-
packet.3	good	good	good	good	-	good	-	good	-
packet.4	good	good	good	good	-	good	-	good	-
dns-answer-A_PTR_c005_beforeDN.bin	bad	good	bad	good	-	bad	-	bad	-
dns-answer-A_PTR_c013_inversion.bin	bad	bad	bad	bad	-	bad	-	bad	-
dns-answer-A_PTR_c01c_loop.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-answer-A_PTR_c035_end.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-answer-A_PTR_c03c_out.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-answer-bad-pointer-A.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-answer-scapy_PTR_c003_malformed.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-answer-scapy_PTR_c065_end.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-answer-scapy_PTR_c06e_out.bin	bad	bad	bad	bad	-	bad	-	bad	-
dns-simple-answer-A_PTR_c005_beforeDN.bin	bad	good	good	good	-	bad	-	bad	-
dns-simple-answer-A_PTR_c013_inversion.bin	bad	bad	bad	bad	-	bad	-	bad	-
dns-simple-answer-A_PTR_c01c_loop.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-simple-answer-A_PTR_c02a_end.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-simple-answer-A_PTR_c032_out.bin	bad	good	bad	bad	-	bad	-	bad	-
dns-truncated-answer-A.bin	bad	bad	bad	bad	-	bad	-	bad	-
dns-truncated-answer-record.bin	bad	bad	bad	bad	-	bad	-	bad	-
empty	bad	bad	bad	bad	-	bad	-	bad	-

Différents outils : Hammer, Nail, Netzob, Nom, Parsifal, RecordFlux
Différents formats : DNS, SSH, IP, PNG

Bugs découverts dans des outils (*crash* dans Hammer, *integer overflow* et boucles infinies dans Nail)

Description des messages et des automates (2/2)

Exemples de descriptions (Nail et Parsifal) :

```
ip_header = {
    uint4 = 4 //version IPV4
    ihl uint4 | !0..4
    [...]
    ip_src uint32
    ip_dst uint32
}
```

```
struct ip_header = {
  version : bit_magic[4; 4];
  ihl : constrained_container
        ((at_least 5)) of bit_int[4];
  [...]
  source_ip : ipv4;
  dest_ip : ipv4;
}
```

Perspectives

- un nouveau langage de description et des compilateurs
- application à des cas compliqués comme QUIC
- extension à la description des sessions et des machines à états

Inférence des machines à états avec L* (1/2)

Travaux pré-existants sur différents protocoles

- de Ruiters and Poll sur TLS (Usenix Security 2015)
- Bossert sur H2 (SSTIC 2016)
- Fiterau-Brostean et al. sur SSH (SPIN'17)

Travaux en cours sur l'identification *automatique* de certaines classes d'erreurs

- oracles de Bleichenbacher
- chemins dangereux dans les machines à états (dans TLS)

Perspectives

- être plus précis
- prise en compte du timing
- application à SSH et QUIC

Inférence des machines à états avec L* (2/2)