Resource-Bounded Intruders in Denial of Service Attacks

Abraão Aires Urquiza   Musab A. Alturki
Tajana Ban Kirigin   Max Kanovich   Vivek Nigam
Andre Scedrov   Carolyn Talcott

CSF 2019 – Hoboken, NJ
June 25-28, 2019

Denial of Service attacks

Denial of Service attacks are a serious security concern:

- No service is protected against DoS attacks:
  flooding attacks can be performed by attackers with large amounts of resources;

- Some DoS attacks do not require large amounts of resources:
  asymmetric DoS attacks,
  amplification attacks,
  slow DoS attacks including SlowDroid attacks where a web server can be taken down by a single mobile phone

Resources in Security Protocols

Such slow attacks are **hard to identify** due to low volumes of traffic.

Intruders minimize their effort and **exploit protocols** used by the target service.

Resources in Security Protocols

Such slow attacks are **hard to identify** due to low volumes of traffic.

Intruders minimize their effort and **exploit protocols** used by the target service.

Intruders also exploit various **types of resources**:

- limited number of workers web-servers possess (Slowloris);
- limited amount of TCAM memory of switches (SDN TCAM exhaustion attacks);
- server's processing power (TLS renegotiation DoS attacks);
- network bandwidth (SIP forking amplification attacks on VoIP);
- ...

Resource-Bounded Intruders

- **Dolev-Yao intruder** with **unlimited resources** can trivially render any service unavailable.
- It is useful in practice and more challenging for formal verification to determine whether a service is vulnerable to intruders with **limited resources**, *i.e.*, **Resource-Bounded Intruders**,

Resource-Bounded Intruders

- **Dolev-Yao intruder** with **unlimited resources** can trivially render any service unavailable.
- It is useful in practice and more challenging for formal verification to determine whether a service is vulnerable to intruders with **limited resources**, *i.e.*, **Resource-Bounded Intruders**, *e.g.*,:
  - Bounded Traffic Intruder Model - intruder can send only a number of messages at a given rate, *e.g.*, messages per second.
  - Bounded Processing Intruder Model - according to his processing power, intruder can carry out only a bounded number of actions in a given time window.
  - Bounded Memory Intruder Models - intruder uses only a bounded amount of memory.

Key Contributions

- We refine the notion of a DoS attack taking into account the duration of the attack.

Key Contributions

- We refine the notion of a DoS attack taking into account the duration of the attack.
- We introduce Protocol Resource Theories that involve resource management and timeouts.

Key Contributions

- We refine the notion of a DoS attack taking into account the duration of the attack.
- We introduce Protocol Resource Theories that involve resource management and timeouts.
- We formalize Resource-Bounded intruders
  - similar to the standard Dolev-Yao intruder
    - able to create fresh values, compose messages, encrypt and decrypt messages with available keys, etc.;
  - amended with resource and time features
    - an intruder can only consume at most some specified amount of resources in any given time window.

Key Contributions

- We refine the notion of a DoS attack taking into account the duration of the attack.
- We introduce Protocol Resource Theories that involve resource management and timeouts.
- We formalize Resource-Bounded intruders
  - similar to the standard Dolev-Yao intruder
    - able to create fresh values, compose messages, encrypt and decrypt messages with available keys, etc.;
  - amended with resource and time features
    - an intruder can only consume at most some specified amount of resources in any given time window.
- We obtain complexity results for the related verification problems.

Key Contributions

- We refine the notion of a DoS attack taking into account the duration of the attack.
- We introduce Protocol Resource Theories that involve resource management and timeouts.
- We formalize Resource-Bounded intruders
  - similar to the standard Dolev-Yao intruder
    - able to create fresh values, compose messages, encrypt and decrypt messages with available keys, etc.;
  - amended with resource and time features
    - an intruder can only consume at most some specified amount of resources in any given time window.
- We obtain complexity results for the related verification problems.
- We automate the search for DoS attacks using Maude.

Model: Multiset Rewriting with Real Time

Timestamped Fact - An atomic FOL formula F with an associated **real** number t, $F@t$. $Time@t$ specifies the global time.

Configuration - A multiset of facts with exactly one occurrence of the fact $Time$.

Model: Multiset Rewriting with Real Time

Timestamped Fact - An atomic FOL formula F with an associated **real** number t, $F@t$. $Time@t$ specifies the global time.

Configuration - A multiset of facts with exactly one occurrence of the fact $Time$.

Timestamps of facts may denote:

- **creation/availability**:

  $N_S(A, C, m)@t$ denotes that message m was sent by agent A on transmission medium C at moment t;

  $E@t$ denotes empty memory slot available from the moment t;

- **validity/expiring**:

  $T_i@t$ denotes that the protocol state S_i times out at moment t.

Model: Multiset Rewriting with Real Time

Timestamped Fact - An atomic FOL formula F with an associated **real** number t, $F@t$. $Time@t$ specifies the global time.

Configuration - A multiset of facts with exactly one occurrence of the fact $Time$.

Formalizing **time requirements**:

- Configurations may have time constraints attached:

e.g., $Time@T, \mathsf{T}_i^{A_k}(S, \vec{X})@T_1 \mid T_1 \geqslant T + 1$

denoting that protocol state S_i will time out in 1 time unit.

Time Constraints - comparisons of time variables:

$T > T' \pm D$ and $T = T' \pm D$

where T and T' are time variables, and D is a natural number.

Model: Multiset Rewriting with Real Time

Formalization involves **explicit real time** and **time requirements**, *i.e.*, comparisons of time variables.

Tick Rule - Advances global time by any positive **real** number ε :

$$Time@T \longrightarrow Time@(T+\varepsilon)$$

Instantaneous Rule - Changes the state, but not the global time:

Time Constraints:
$T > T' \pm D$ and $T = T' \pm D$

$$Time@T, \mathcal{W}, F_1@T'_1, \ldots, F_n@T'_n \mid C \longrightarrow$$
$$\exists \vec{x}. Time@T, \mathcal{W}, Q_1@(T+D_1), \ldots, Q_m@(T+D_m)$$

where $D_1, \ldots, D_m, D$ are natural numbers and $\vec{x}$ are nonces.

Execution time of actions - rules may have explicit **duration**, specified by timestamps of created facts.

Protocol Resource Theory

Formalization also involves **resource** management and requirements.

Services allocate their available resources to each client request for some period of time.

- Resources are consumed and recovered during **protocol execution**:

$$\begin{gathered} Time@T, \mathsf{S}_i(S, S_{id}, r_i, \vec{X}_i)@T_1, \mathsf{R}(S, R + r_j - r_i + r^S_{min})@T_2, \\ \mathsf{N}(m_i)@T_3, \mathcal{W}_1, \mathcal{W} \;\mid\; T_1 \geqslant T, T_2 \leqslant T, T_3 \leqslant T \longrightarrow \\ Time@T, \mathsf{S}_j(S, S_{id}, r_j, \vec{x}_j)@(T + t_j), \mathsf{R}(S, R + r^S_{min})@T, \mathcal{W}_2, \mathcal{W} \end{gathered}$$

Special facts, variables and constants are used in the model:

$\mathsf{R}(s, r)@t$ - service s has r **resources available** at moment t;

r^S_{min} **minimal service resource** - the lower bound on resources required by the service S to work;

r^S_{ini} **initial service resources** for the service S.

Protocol Resource Theory

Formalization also involves **resource** management and requirements.

- **Service availability rules** specify:

  - service is **denied** when resources reach the minimum:

$$\begin{array}{r} Time@T,\ \mathrm{R}(S, r^S_{min})@T, \mathrm{Av}(S)@T_2 \ \longrightarrow \\ Time@T,\ \mathrm{R}(S, r^S_{min})@T, \mathrm{Den}(S)@T \end{array}$$

  - service is **available** if the resources are greater than the minimum:

$$\begin{array}{r} Time@T,\ \mathrm{R}(S, r^S_{min} + R + 1)@T, \mathrm{Den}(S)@T_2 \longrightarrow \\ Time@T,\ \mathrm{R}(S, r^S_{min} + R + 1)@T, \mathrm{Av}(S)@T \end{array}$$

$\mathrm{Av}(s)@t$ - **service** s **is available** from moment t;

$\mathrm{Den}(s)@t$ - **service** s **is unavailable** from moment t.

Denial of Service Attacks

Very short service interruptions may be tolerated in practice.

Hence, the notion of a DoS attack is refined by a duration parameter:

A DoS attack on a service is successful
if the service's resources are exhausted
for some **duration**, *mdur*.

$$Time@T,\ \mathrm{Den}(S)@T_1 \ \mid\ T \geqslant T_1 + mdur_S$$

For simplicity we model only one resource
and use natural numbers to represent resources.

Verification Problems

Verification: We look for traces representing DoS attacks!

Formalization of the verification scenario:

- Protocol resource theories - representing used services $\mathcal{A}_i$
- natural numbers $mdur_i$, specifying the minimal duration that the resources of the service $\mathcal{A}_i$ have to be consumed to represent a successful DoS attack on that service;
- Intruder theories $\mathcal{I}_j$

Resource-Bounded Intruders

- **Dolev-Yao intruder** with unlimited resources - not suitable for DoS verification, leads to many false positives.
- **Resource-Bounded Intruders** - more refined intruder model, -amended with resource and time features:
  - Intruder has a bounded total amount of resources:
    r_{max}^{id} **total amount of resources** of an intruder *id*;
  - Intruder can only consume a bounded amount of resources in any given time window;
  - Intruder obeys physical laws related to **time**:
    - obeys network transmission time restrictions;
    - intruder's actions take time to be performed.

Resource-Bounded Intruder Theory

- Actions of **Resource-Bounded Intruders** have the attached cost,

  e.g., sending a message to the network:

$$\textbf{SND:}\ Time@T, M(I,X)@T_1, \mathrm{R}(I, Z + r_R)@T_2 \mid T \geqslant T_1 \longrightarrow$$
$$Time@T, N(X)@(T+\delta_L), \mathrm{R}(I,Z)@T, \mathrm{Rec}(I, r_R)@(T+\delta_R)$$

  - sending message X takes intruder I δ_L time units;
  - it consumes r_R resources;
  - these resources can only be recovered after δ_R time units.

  $\mathrm{R}(id, r)@t$ - intruder *id* has r **resources available** at moment t;

  $\mathrm{Rec}(id, r)@t$ - intruder *id* can **recover** r **resources** at moment t.

Model: Multiset Rewriting with Real Time

Verification: We look for traces representing DoS attacks!

A **trace** of MSR rules $\mathcal{T}$ is a sequence of configurations

$\mathcal{S}_0 \longrightarrow_{r_1} \mathcal{S}_1 \longrightarrow_{r_2} \cdots \longrightarrow_{r_n} \mathcal{S}_n$,

such that for all i, $\mathcal{S}_{i+1}$ is obtained by applying $r_{i+1} \in \mathcal{T}$ to $\mathcal{S}_i$.

We are interested in traces that reach some goal.

Goal is a set $\mathcal{GS} = \{ \langle \mathcal{S}_1, \mathcal{C}_1 \rangle, \ldots, \langle \mathcal{S}_n, \mathcal{C}_n \rangle \}$ where each $\mathcal{C}_j$ is a set of time constraints and $\mathcal{S}_j$ is a multiset of timestamped facts.

A configuration $\mathcal{S}$ is a **goal configuration** if for some i there is a grounding substitution, σ, such that $\mathcal{S}_i\sigma \subseteq \mathcal{S}$ and $\mathcal{C}_i\sigma$ is true.

Model: Multiset Rewriting with Real Time

We consider traces that do not contain critical configurations, *i.e.*, non-critical traces.

Critical Configuration Specification is a set $\mathcal{CS} = \{ \langle \mathcal{S}_1, \mathcal{C}_1 \rangle, \ldots, \langle \mathcal{S}_n, \mathcal{C}_n \rangle \}$ where each $\mathcal{C}_j$ is a set of time constraints and $\mathcal{S}_j$ is a multiset of timestamped facts.

A configuration $\mathcal{S}$ is a **critical configuration** if for some i there is a grounding substitution, σ, such that $\mathcal{S}_i\sigma \subseteq \mathcal{S}$ and $\mathcal{C}_i\sigma$ is true.

Non-Critical Reachability Problem:
Given a goal $\mathcal{GS}$, a critical configuration specification $\mathcal{CS}$ and an initial configuration $\mathcal{S}_0$, is there a non-critical trace $\mathcal{P}$ of MSR rules $\mathcal{T}$ that leads from $\mathcal{S}_0$ to a goal configuration?

Complexity Results

Non-Critical Reachability Problem in MSR systems is **undecidable** in general, but is **decidable** for systems containing only balanced rules, assuming an upper-bound on the size of facts, *i.e.*, on the total number of symbols in each fact.

MSR		**Non-Critical Reachability**
Balanced	untimed	PSPACE-complete
	discrete time	PSPACE-complete
	dense time	PSPACE-complete [new]
Not necessarily balanced		Undecidable

A rule is **balanced** if the number of facts appearing in its pre-condition and its post-condition is the same.

Complexity Results

Several challenges in **timed MSR with dense time**, particularly related to the notion of non-critical traces:

- Showing that a trace of a dense time MSR is non-critical involves not only the configurations it contains, but also an **infinite number of configurations** obtained by decomposing *Tick* rules, in order to faithfully capture the continuity of time.
- Technical results involve abstractions, **circle-configurations**, and the auxiliary notion of **immediate successor configurations**, related to satisfiability of relevant time constraints.

Verification Problems

Protocol Verification: We look for traces representing DoS attacks!

DoS Problem:
Given a verification scenario, can resource-bounded intruders deny some service s_i for the duration $mdur_i$?

Verification Problems

Protocol Verification: We look for traces representing DoS attacks!

> **DoS Problem**:
> Given a verification scenario, can resource-bounded intruders deny some service s_i for the duration $mdur_i$?

DoS problem is an instance of the non-critical reachability problem!

Theorem (DoS problem)

The DoS problem is undecidable in general.

Theorem (Balanced DoS problem)

Assuming a bound on the size of facts, the DoS problem for balanced verification scenarios is PSPACE-complete.

Verification Problems

Protocol Verification: We look for traces representing DoS attacks!

DoS Problem:
Given a verification scenario, can resource-bounded intruders deny some service s_i for the duration $mdur_i$?

DoS problem is an instance of the non-critical reachability problem!

- initial configuration specifies available resources of each service and each intruder, initial knowledge etc.
- goal: $Time@T,\ \mathsf{Den}(S)@T_1 \mid T \geqslant T_1 + mdur_S$
- critical configurations related to services

Protocol Resource Theory

Protocol Critical Configuration Specification ensures protocol execution respects timeouts and resource bounds:

- **Timeout** $\mathcal{CS}$: $\langle \{ Time@T, S_i(s, S_{id}, R_i, \vec{x_i})@T_1 \}, \{ T_1 < T \} \rangle$ configurations denoting protocol sessions for which the timeout has passed are critical;

Protocols states may have **timeouts**:

- $\mathsf{S}_i(s, S_{id}, r_i, \vec{x_i})@t$ - protocol state S_i times out at moment t
- once a timeout is reached, protocol session ends.

$$Time@T, \mathsf{R}(s, R)@T_1, \mathsf{S}_i(s, S_{id}, r_i, \vec{x_i})@T \longrightarrow$$
$$Time@T, \mathsf{R}(s, r_i + R)@T$$

Protocol Resource Theory

Protocol Critical Configuration Specification ensures protocol execution respects timeouts and resource bounds:

- **Timeout** $\mathcal{CS}$: $\langle \{ Time@T, S_i(s, S_{id}, R_i, \vec{x_i})@T_1 \}, \{ T_1 < T \} \rangle$ configurations denoting protocol sessions for which the timeout has passed are critical;
- **Denied** $\mathcal{CS}$: $\langle \{ \mathsf{R}(s, r^s_{min})@T_1, \mathsf{Av}(s)@T_2 \rangle \}, \{ T_1 \leqslant T_2 \}$ configurations are critical if a service is considered available at a time its resources have been exhausted:

Protocol Resource Theory

Protocol Critical Configuration Specification ensures protocol execution respects timeouts and resource bounds:

- **Timeout** $\mathcal{CS}$: $\langle\{Time@T, S_i(s, S_{id}, R_i, \vec{x_i})@T_1\}, \{T_1 < T\}\rangle$ configurations denoting protocol sessions for which the timeout has passed are critical;
- **Denied** $\mathcal{CS}$: $\langle\{\mathsf{R}(s, r^s_{min})@T_1, \mathsf{Av}(s)@T_2\rangle\}, \{T_1 \leqslant T_2\}$ configurations are critical if a service is considered available at a time its resources have been exhausted:
- **Available** $\mathcal{CS}$: $\langle\{\mathsf{R}(s, r^s_{min} + R + 1)@T_1, \mathsf{Den}(s)@T_2\}, \{T_1 \leqslant T_2\}\rangle$ service should not be considered denied at anytime when sufficient resources are available.

Time-Based Countermeasures for DoS attacks

We extend our model to take into account **countermeasures**.

Countermeasures based on timeouts,
e.g., ReqTimeOut for mitigating the Slowloris attack.

- Trigger a timeout whenever a condition on the traffic is satisfied:

$$\begin{array}{l} \mathit{Time}@T, \mathrm{R}(s, R)@T_1, \mathrm{S}_i(s, S_{id}, r_i, \vec{x}_i)@T_2,\ \mathrm{TimeCM}(s, S_{id})@T \\ \quad \longrightarrow \mathit{Time}@T, \mathrm{R}(s, r_i + R)@T \end{array}$$

  Critical configuration: $\langle \{\mathit{Time}@T, \mathrm{TimeCM}(s, S_{id})@T_1\}, \{T_1 < T\} \rangle$

- The connection S_{id} of service s is closed;
- The service's resources r_i are made available.

Experimental Results

- We implemented the framework using Rewriting Modulo SMT in Maude.

Experimental Results

- We implemented the framework using Rewriting Modulo SMT in Maude.
- It uses symbolic search with the following symbols:
  - **Time Symbols**, that is, instead of instantiating time variable, we use time symbols that are constrained by a set of constraints. We use an SMT solver to check for the satisfiability of this set.
  - **Intruder and Service Resource Symbols:** Instead of using concrete values for intruder and service resources, we use intruder resource symbols and service resource symbols;
  - **Protocol Instance Symbols:** Instead of creating one protocol session, we allow the intruder to create several instances of a protocol session representing a burst from the intruder.

Experimental Results

We carried out bounding model checking with the following bounds:

- **Bound on the Number of Parallel Symbolic Protocols (pxs).**
- **Bound on the Number of Messages at a Time (pxs).**

	No Bounding		**Bounded msgs_1**		**Bounded pxs**		**Bounded msgs_1 and pxs**	
Attack	States	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)
SL [1]	18	0.4	16	0.4	8	0.1	7	0.1
SL [2]	409	13	277	11.2	27	0.4	17	0.4
SL [3]	–	–	–	–	228	4.7	56	2.6
STCAM [2]	17	0.3	15	0.3	16	0.3	14	0.2
STCAM [3]	387	12.5	266	9.7	361	10.3	243	9.2
STCAM [4]	–	–	–	–	12783	561	6322	474

Conclusion

- We refine the notion of a DoS attack taking into account the duration of the attack.
- We propose a framework for analyzing the security of systems against DoS attacks:
  - reasoning about service's resources;
  - reasoning about intruder's resources;
  - reasoning about service timeouts;
- We obtain complexity results for the DoS problem and a general non-critical reachability problem for real time MSR;
- We automate the verification using Rewriting Modulo SMT.

Future Work

- Capture a larger class of security problems that are closely related to DoS attacks.
- Use statistical model checking to investigate effectiveness of intruder strategies and attack defenses.
- Investigate how different resource-bounded intruder models can be compared.

Thank you!

Non-Critical Traces in MSR with Real Time

Showing that a trace of a dense time MSR is non-critical involves not only the configurations it contains, but also an infinite number of configurations obtained by decomposing *Tick* rules in order to faithfully capture the continuity of time.

Definition (Non-Critical Traces)

Let $\mathcal{R}$ be a set of timed MSR rules and $\mathcal{CS}$ a critical configuration specification. A trace $\mathcal{P}$ of $\mathcal{R}$ rules is non-critical if it contains no critical configuration and if no critical configuration is reachable along any trace obtained by matching any subtrace of $\mathcal{P}$ on the left below with the one on the right:

$$\mathcal{S}_i \longrightarrow_{Tick_\varepsilon} \mathcal{S}_{i+1} \qquad\qquad \mathcal{S}_i \longrightarrow_{Tick_{\varepsilon_1}} \mathcal{S}' \longrightarrow_{Tick_{\varepsilon_2}} \mathcal{S}_{i+1}$$

where ε_1 and ε_2 are arbitrary non-negative real numbers, such that, $\varepsilon = \varepsilon_1 + \varepsilon_2$ holds.