TIME-BASED VERSIONING

Conceptual Overview

Abstract

This paper provides an overview of Time-based Versioning concepts in the context of supporting OASIS CTI TC discourse and exploration of options to meet community requirements.

Time-based versioning separates Structure from State which allows versioning of Structure independently of its State.

Acknowledgement - This paper incorporates concepts and content derived from:
Time-Based Versioned Graphs May 13th, 2014
Published in Cypher, Graph Databases, Neo4j by Ian Robinson
http://iansrobinson.com/2014/05/13/time-based-versioned-graphs/

Patrick Maroney
pmaroney@specere.org

Time-Based Versioning

OVERVIEW

Time-based Versioning provides a universal scheme that satisfies the following stated community objectives/requirements for versioning:

(1) I only need/want the latest version of the model.
(2) I need to establish the state of the model (1) at a given point or (2) over a range in time.
(3) I need to request a copy of the model and all of its states (1) at a given point or (2) over a range in time.

This paper seeks to provide an overview of the Time-based Versioning concept in the context of supporting OASIS CTI TC discourse and exploration of options to meet community requirements.

REQUIREMENTS FOR TIME BASED VERSIONING

Time-based Versioning requires relatively minor changes to existing constructs and the definition of two new Top Level Object and Relationship Types.

(1) The following Top Level Object and Relationship constructs in the CTI Data Model:

Identify Nodes - contain one or more immutable properties, which together constitute an entity's identity
Structural Relationships - Time stamped structural relationships interconnecting Identity nodes
State Nodes - An immutable snapshot of an entity's state.
State Relationships – Time stamped state relationships connecting State nodes to Identity nodes.

Note that the terminologies in this model:

> ***Identify Nodes, State Nodes,*** *and* ***State Relationships*** *essentially map to Top Level Objects.*
> ***Structural Relationships*** *essentially map to Top Level Relationships.*

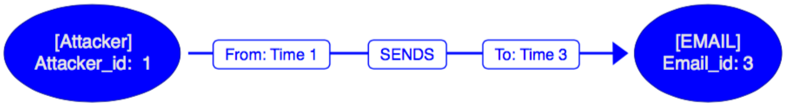
(2) The addition of "**From**" and "**To**" timestamp fields to the following Relationship constructs:

State Relationships
Structural Relationships

(2.1) The definition, use, and recognition of a Timestamp Keyword ("**EOT**", "**~**", or similar convention) in the "**To**" Timestamp field to provide a shorthand notation that represents the concept of **"the end of time".**

*It is important to note that these "**From**" and "**To**" times represent when the Node or Relationship was Created, Revised, or Deleted.*

So in the example below: [Attacker]➔ SENDS→[EMAIL] from Time 1 to Time 3 does not represent that the Attacker sent the Email between Time 1 and Time 3.



Time-Based Versioning

CONCEPTS

Separate **Structure** From **State**

The key to time-based versioning is separating **Structure** from **State**. This allows us to version the **Structure** independently of its **State**.

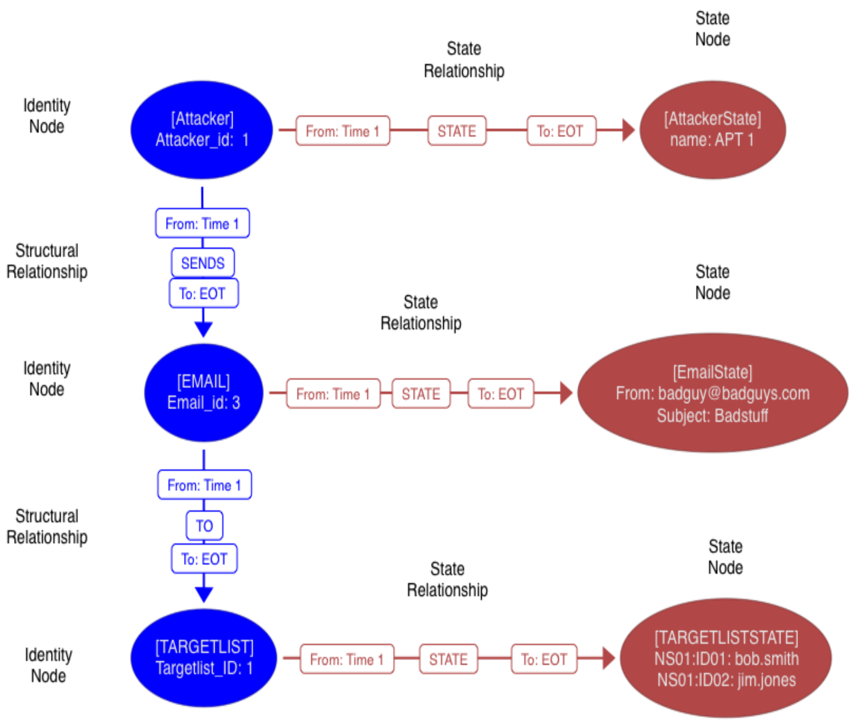
Identity Nodes

Each identity node contains one or more immutable properties, which together constitute an entity's identity.

Identity nodes serve only to identify an entity and locate it in a network structure.

Structural Relationships

Identity nodes are connected to one another using time stamped structural relationships.



State Nodes

Connected to each identity node are one or more state nodes.

Each state node represents an immutable snapshot of an entity's state.

State Relationships

State nodes are connected to identity nodes using time stamped state relationships.

The changes we track in a version-aware model are (1) changes to state and (2) changes to structure:

(1) Changes to state involve adding, removing and modifying node properties.

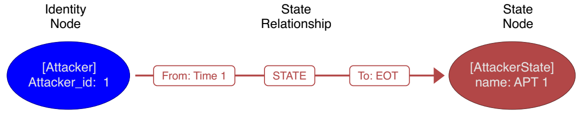
(2) Changes to structure involve adding and deleting relationships, as well as modifying the strength, weight or quality of relationships by changing one or more of their properties.

We establish the current structural and state relationships of a version-aware model by simply matching the "**To**" property on relationships against our "**EOT**" value.

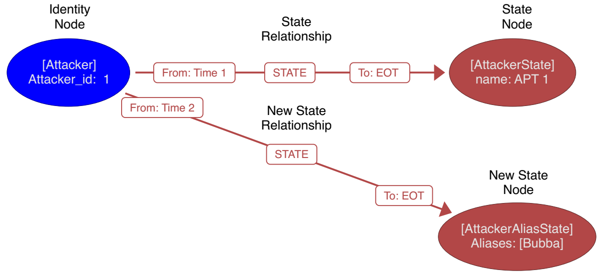
Time-Based Versioning

EXAMPLE 1 : ADDING & UPDATING NODES AND RELATIONSHIPS

@Time 1 - Add Attacker 1:
Identity Node: [Attacker: 1]
State Node [Name: APT1]
State Relationship [From: Time 1, To: EOT]



@Time 2 - Add Alias Bubba to Attacker 1:
State Node: [Aliases: (Bubba)]
State Relationship [From: Time2, To: EOT]
to Identity Node: [Attacker: APT1]



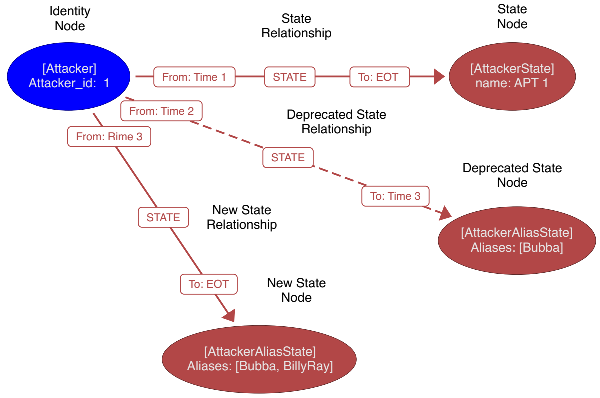
@Time 3 - Add alias BillyRay to Attacker 1:
State Node: [Aliases: (Bubba, BillyRay)]
State Relationship [From: Time 3, To: EOT]

Those who need/want to establish the state of the model at any given point or range in time:

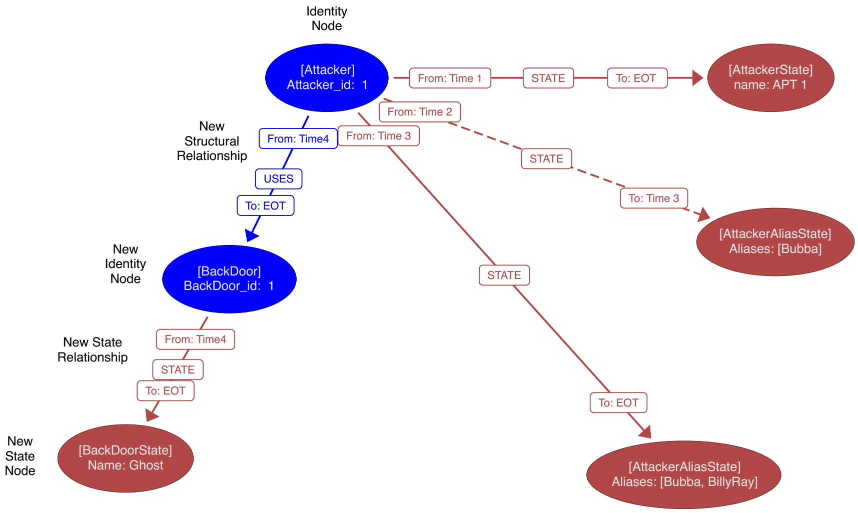
Retain State Node: [Aliases: (Bubba)]
Change Alias State Relationship to
[From: Time 2, To: Time 3]

Those who only need/want the latest version of the model:

Delete State Node: [Aliases: (Bubba)]
Delete Alias State Relationship [From: Time 2, To: EOT]



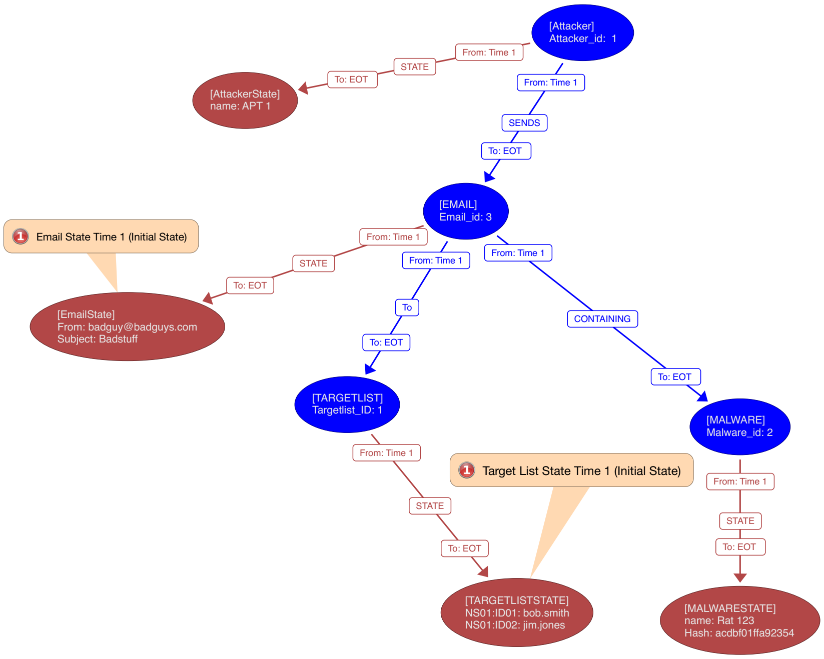
@Time 4: Add Backdoor Ghost to Attacker 1
Identity Node: [Backdoor: 1]
Structural Relationship: [From: Time 4,To: EOT]
State Node: [BackdoorName: Ghost]
State Relationship: [From: Time 4, To: EOT]


Identity Node
[Attacker] Attacker_id: 1
From: Time 1
STATE
To: EOT
[AttackerState] name: APT 1
From: Time 2
STATE
To: Time 3
[AttackerAliasState] Aliases: [Bubba]
New Structural Relationship
From: Time4
From: Time 3
USES
To: EOT
New Identity Node
[BackDoor] BackDoor_id: 1
STATE
New State Relationship
From: Time4
STATE
To: EOT
To: EOT
New State Node
[BackDoorState] Name: Ghost
[AttackerAliasState] Aliases: [Bubba, BillyRay]

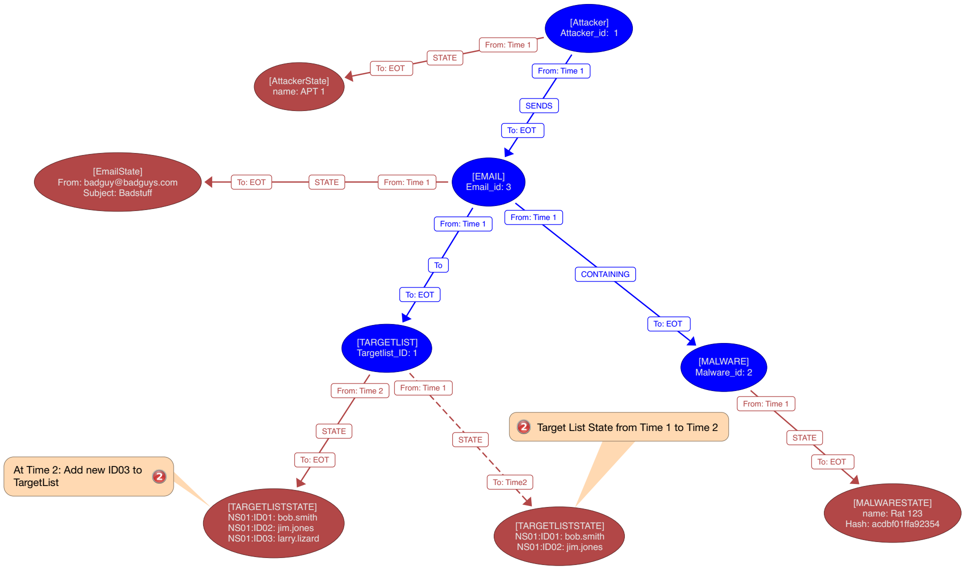
Time-Based Versioning

EXAMPLE 2 : MODELING ATTACKER, EMAILS, MALWARE, AND TARGET LISTS

[TIME 1] CREATE INITIAL MODEL



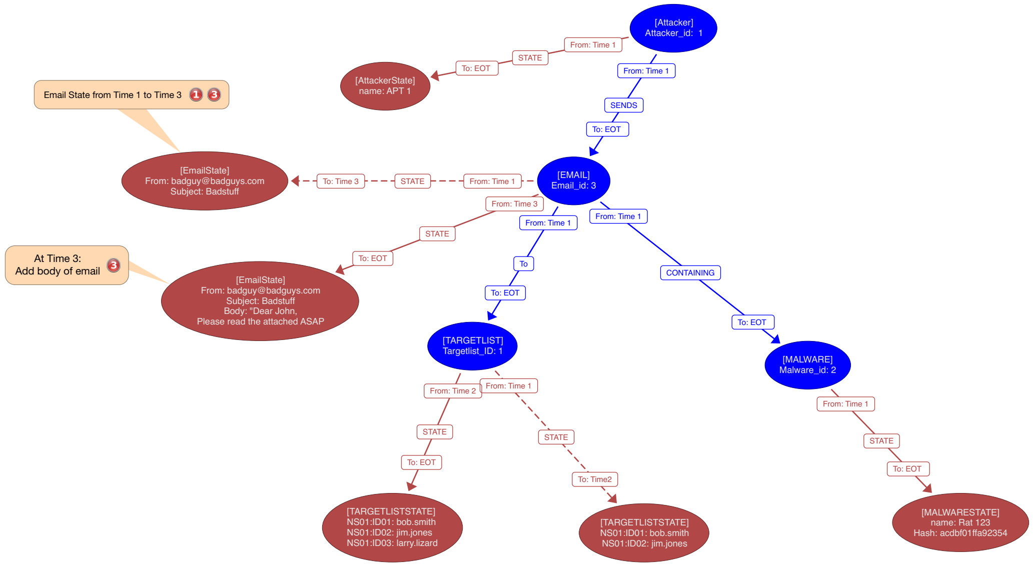
[TIME 2] ADD NEW TARGET TO [TARGET LIST: 1]



Time-Based Versioning

EXAMPLE 2 : MODELING ATTACKER, EMAILS, MALWARE, AND TARGET LISTS

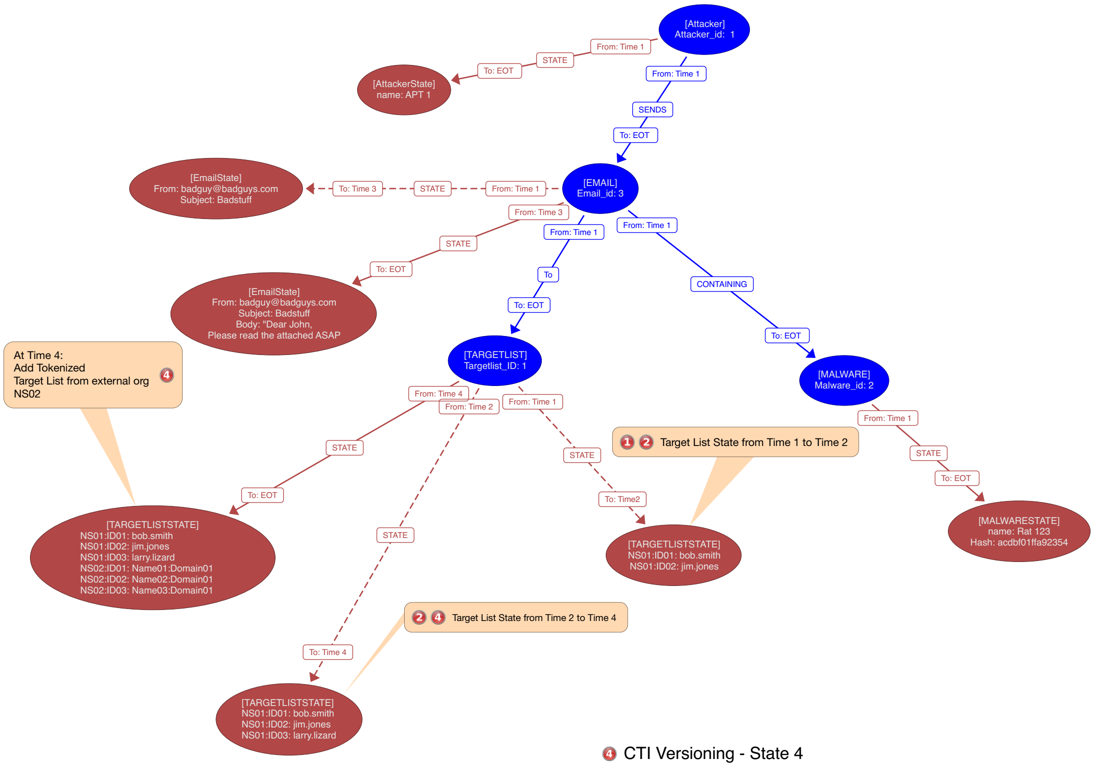
[TIME 3] ADD ADDITIONAL DETAILS TO [EMAIL: 3]



Email State from Time 1 to Time 3 ① ③

At Time 3:
Add body of email ③

[TIME 4] ADD TOKENIZED TARGET LIST FROM EXTERNAL ORG TO [TARGET LIST: 1]



At Time 4:
Add Tokenized
Target List from external org
NS02 ④

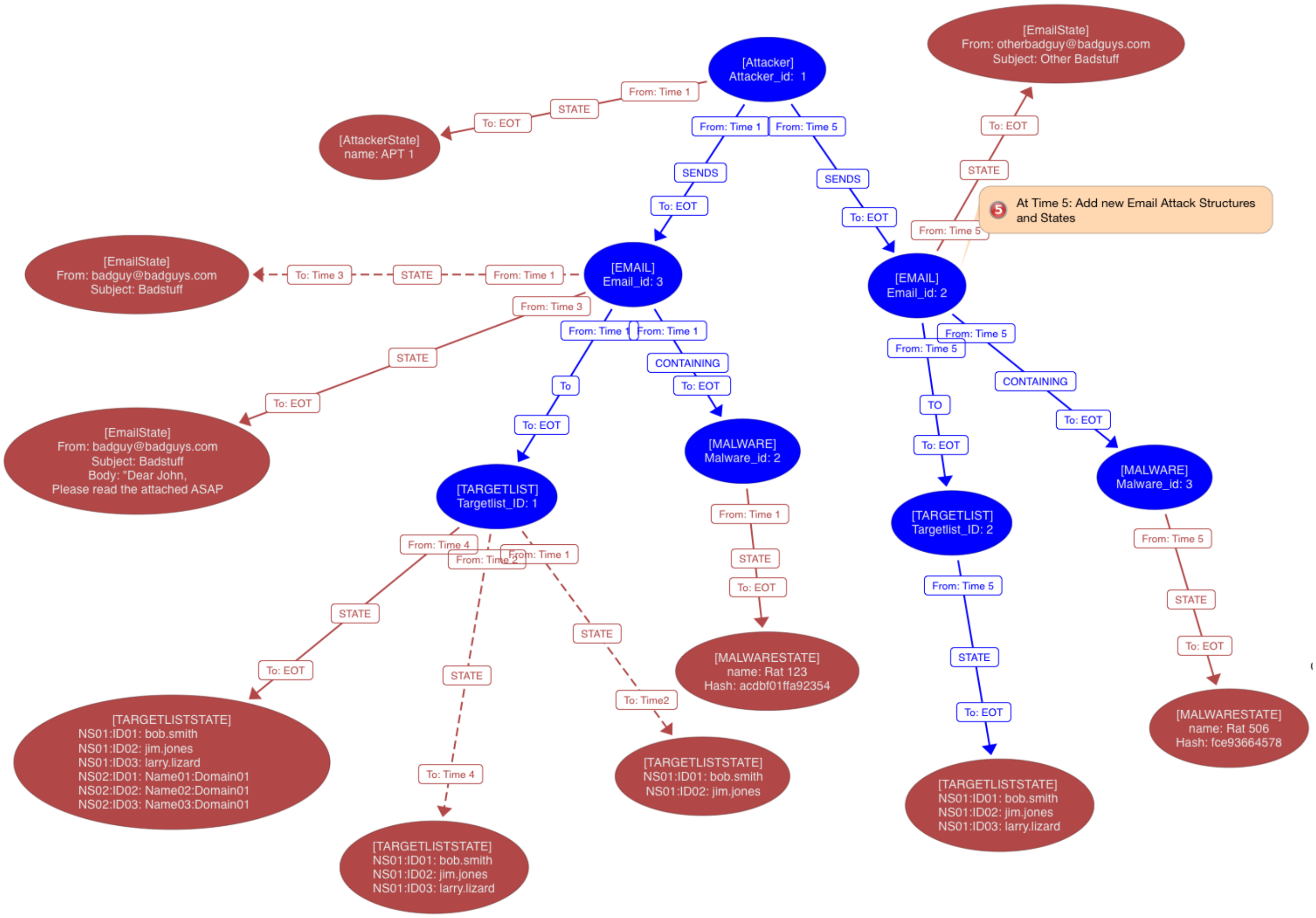
① ② Target List State from Time 1 to Time 2

② ④ Target List State from Time 2 to Time 4

④ CTI Versioning - State 4

EXAMPLE 2 : MODELING ATTACKER, EMAILS, MALWARE, AND TARGET LISTS

[TIME 5] ADD NEW ATTACK EMAILS AND MALWARE FROM [ATTACKER: 1]



At Time 5: Add new Email Attack Structures and States