Αυτοματοποιημένη Κατανομή Κανόνων Ελέγχου Πρόσβασης σε Κατάλληλα Σημεία Ακαδημαϊκών Δικτύων

Αδάμ Παυλίδης

Εργαστήριο Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων Τηλεματικής

NETwork **M**anagement & **O**ptimal **DE**sign Laboratory - **NETMODE**

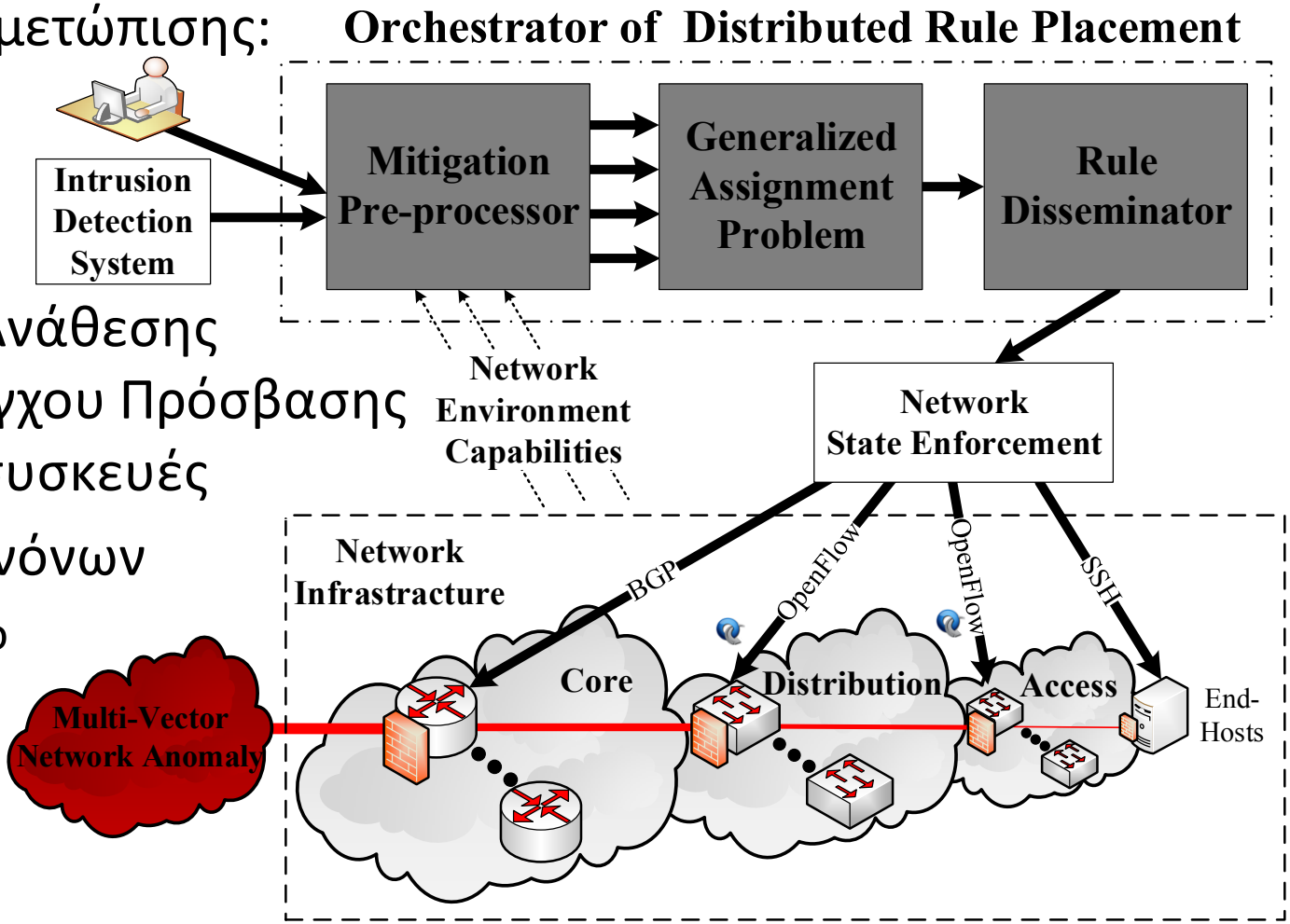
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ)

Συμπόσιο Ψηφιακής Τεχνολογίας «20 χρόνια ΕΔΕΤ», Νοέμβριος 2018

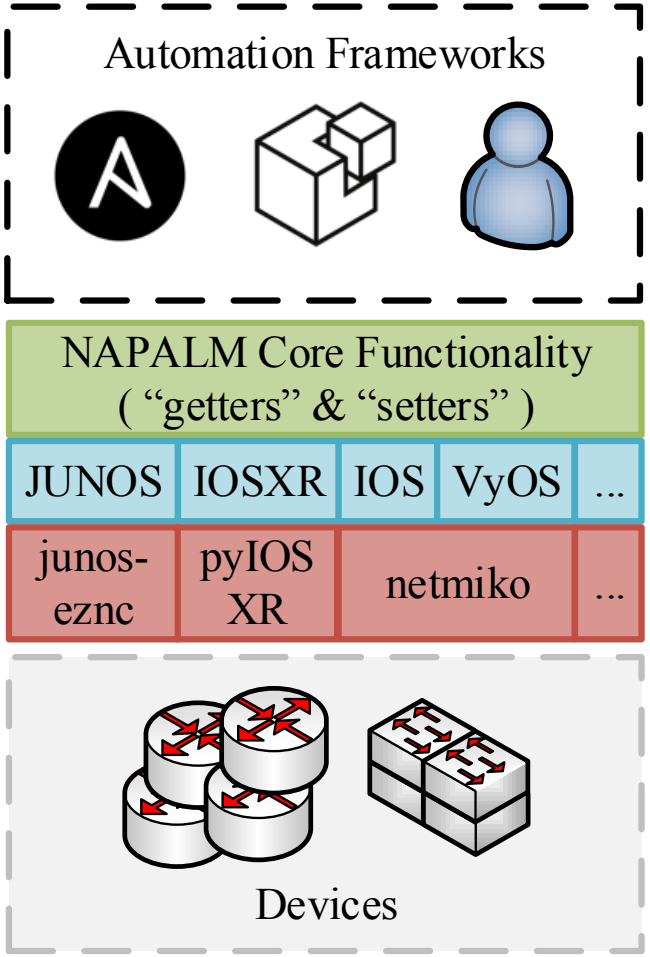
Εισαγωγή

- On-premise Αντιμετώπιση Multi-Vector επιθέσεων DDoS
- Επίπεδα Αντιμετώπισης:
  - Core
  - Distribution
  - Access
- Αλγόριθμος Ανάθεσης Κανόνων Ελέγχου Πρόσβασης σε επίπεδα/συσκευές
- Εισαγωγή Κανόνων
  - Πρωτόκολλο
  - API



Network **A**utomation and **P**rogrammability **A**bstraction **L**ayer with **M**ultivendor support - **NAPALM**

- Ενιαία διαχείριση συσκευών ανεξαρτήτως κατασκευαστή
  https://napalm-automation.net/
  - Python-based
  - Υποστηριζόμενα OS
    - EOS, Junos, IOS-XR, NX-OS, IOS
    - vyos, cumulus, asa, dellos10, ros, fortios
- Λειτουργίες
  - “getters” (e.g. bgp, routes, interfaces)
  - “setters” (configuration templates)
- Χρήση σε:
  - Ansible
  - Salt

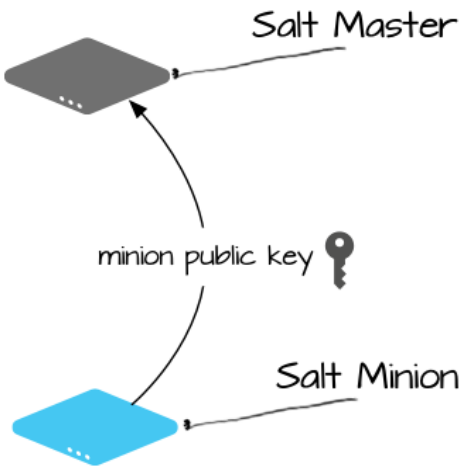


Πλατφόρμα SaltStack – Salt (1/4)

- Πλατφόρμα για αυτοματοποίηση & Configuration Management
  https://docs.saltstack.com/en/getstarted/
  - Python-based, Open Source και Enterprise
- Master – Minion
  - "Proxy" Minions: δικτυακές συσκευές (π.χ. **NAPALM**)
- Βιβλιοθήκες
  - Execution Modules: Χρήση από CLI
  - State Modules: Χρήση σε **S**a**L**t **S**tate Files - **SLS**
- Δεδομένα
  - "Grains", "Pillar", "Mine"

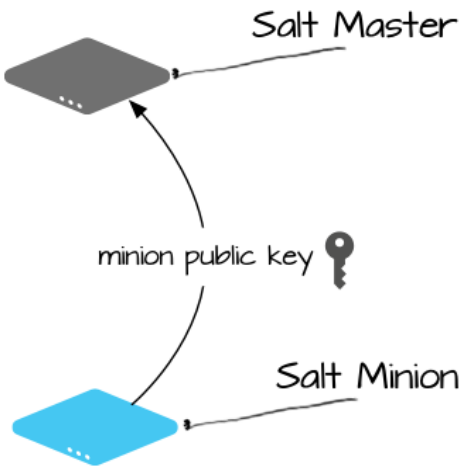
Πλατφόρμα SaltStack – Salt (2/4)

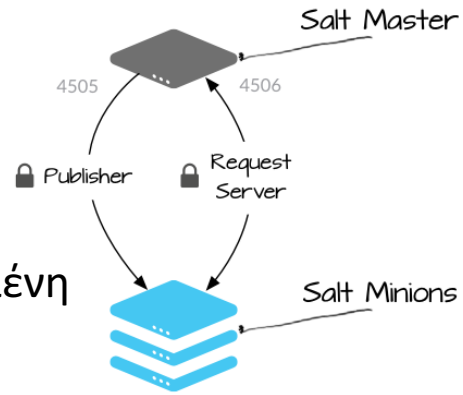
- Πλατφόρμα για αυτοματοποίηση & Configuration Management
  https://docs.saltstack.com/en/getstarted/
  - Python-based, Open Source και Enterprise
- Master – Minion
  - "Proxy" Minions: δικτυακές συσκευές (π.χ. **NAPALM**)
- Βιβλιοθήκες
  - Execution Modules: Χρήση από CLI
  - State Modules: Χρήση σε **S**a**L**t **S**tate Files - **SLS**
- Δεδομένα
  - "Grains", "Pillar", "Mine"



Πλατφόρμα SaltStack – Salt (3/4)

- Πλατφόρμα για αυτοματοποίηση & Configuration Management
  https://docs.saltstack.com/en/getstarted/
  - Python-based, Open Source και Enterprise
- Master – Minion
  - "Proxy" Minions: δικτυακές συσκευές (π.χ. **NAPALM**)
- Βιβλιοθήκες
  - Execution Modules: Χρήση από CLI
  - State Modules: Χρήση σε **Sa**L**t** **S**tate Files - **SLS**
- Δεδομένα
  - "Grains", "Pillar", "Mine"

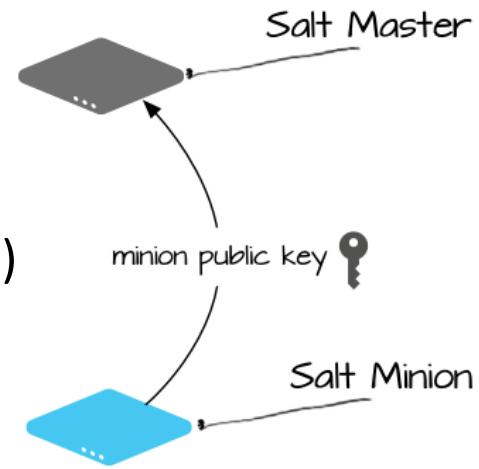


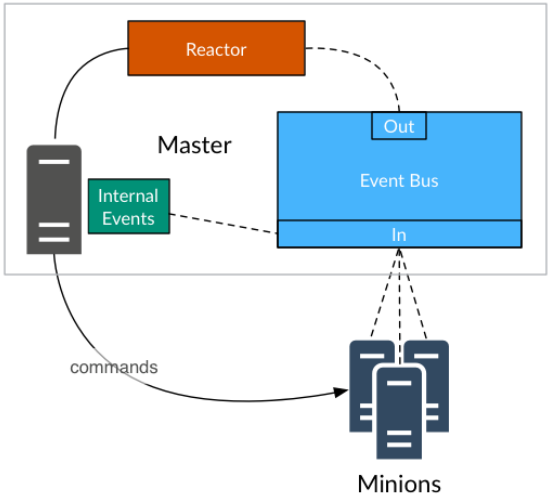


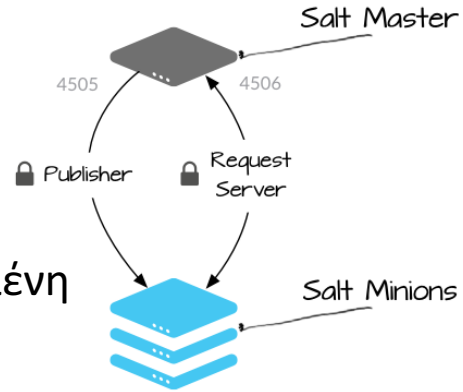
Κατανεμημένη Εκτέλεση

Πλατφόρμα SaltStack – Salt (4/4)

- Πλατφόρμα για αυτοματοποίηση & Configuration Management
  https://docs.saltstack.com/en/getstarted/
  - Python-based, Open Source και Enterprise
- Master – Minion
  - "Proxy" Minions: δικτυακές συσκευές (π.χ. **NAPALM**)
- Βιβλιοθήκες
  - Execution Modules: Χρήση από CLI
  - State Modules: Χρήση σε **Sa**L**t** **S**tate Files - **SLS**
- Δεδομένα
  - "Grains", "Pillar", "Mine"



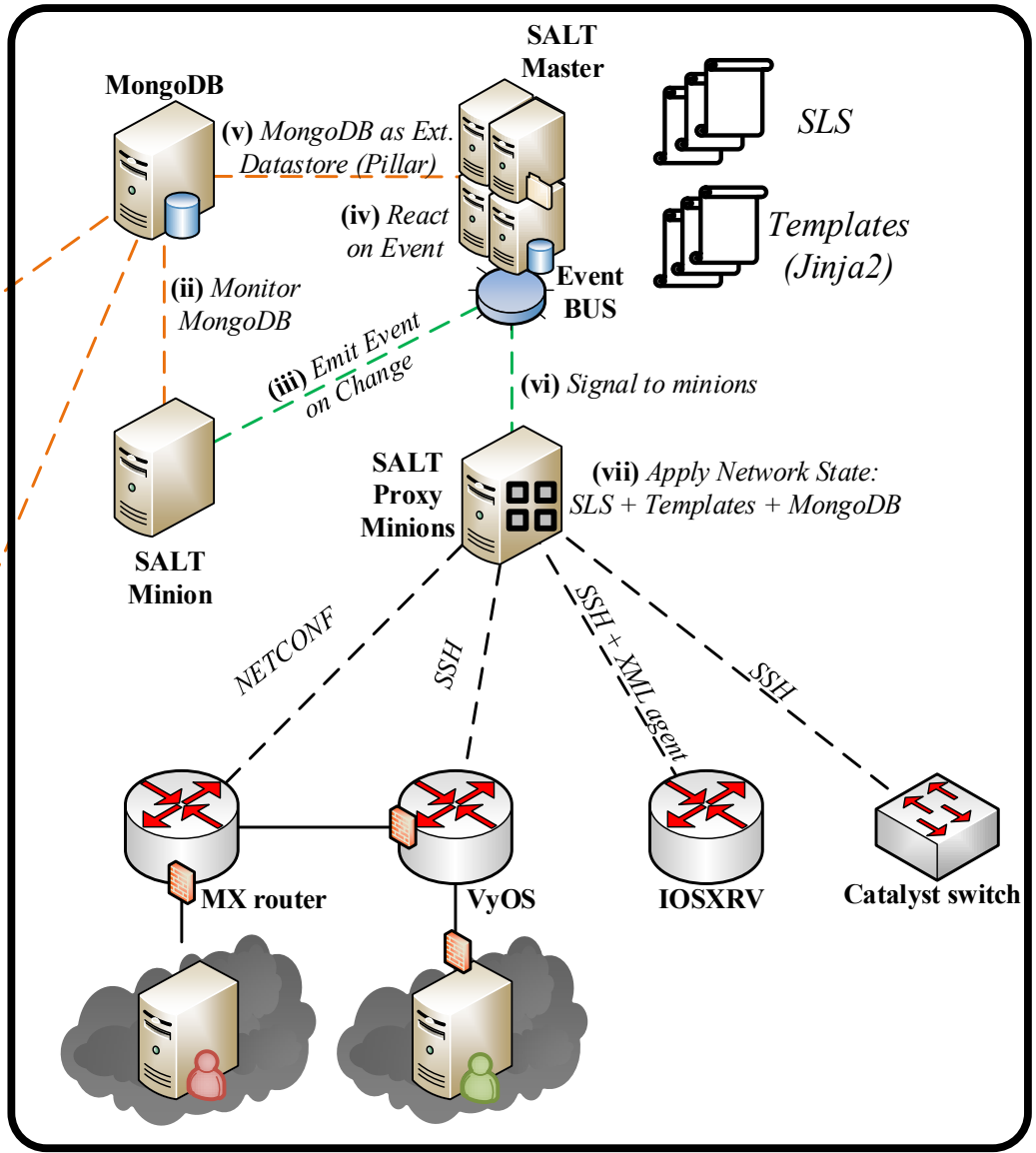




Κατανεμημένη Εκτέλεση

Κατανομή Κανόνων Ελέγχου Πρόσβασης: SALT + NAPALM

- Διευθύνσεις IP -> MongoDB
  - Βάση Δεδομένων για Salt
- "Beacon"
  - *Poll MongoDB*
  - *Αλλαγή: Event*
- "Reactor"
  - *Event*: Apply SLS
- SLS
  - Μαζική διανομή Configuration για έλεγχο πρόσβασης



Workflow Κατανομής Κανόνων: Salt + NAPALM (1/5)

Event -> Reactor -> Apply State -> Template -> Config

```
salt/beacon/eyrie/mongo_beacon/  {
    "_stamp": "2018-10-29T12:21:23.565440",
    "data": "trox,mx80,vyos3,dna1,",
    "id": "eyrie"
}
```

Workflow Κατανομής Κανόνων: Salt + NAPALM (2/5)

Event -> **Reactor** -> Apply State -> Template -> Config

```
reactor:
  - 'salt/beacon/eyrie/mongo_beacon/':
    - salt://reactor/odrp_update.sls

apply_odrp:
  local.state.apply:
    - tgt: {{ data['data'] }}
    - tgt_type: list
    - arg:
      - odrp.update
```

Workflow Κατανομής Κανόνων: Salt + NAPALM (3/5)

Event -> Reactor -> **Apply State** -> Template -> Config

```
update_prefix:
  netconfig.managed:
    - template_name: salt://salt_templates/odrp/update_prefix/{{
grains['os'] }}.jinja
    - prefix_list: {{ pillar.get('prefix_list', '') }}
    - prefixes: {{ pillar.get('malicious', []) | tojson }}
    - filter_name: {{ pillar.get('filter_name') }}
```

Workflow Κατανομής Κανόνων: Salt + NAPALM (4/5)

Event -> Reactor -> Apply State -> **Template** -> Config

```
no ipv4 access-list {{ filter_name }}
ipv4 access-list {{ filter_name }}
{%- for prefix in prefixes %}
 {{ loop.index }} deny ipv4 host {{ prefix }} any
{%- endfor %}
 permit ipv4 any any
end
```

```
IOSXR          delete policy-options prefix-list {{ prefix_list }}
               {%- for prefix in prefixes %}
               set policy-options prefix-list {{ prefix_list }} {{ prefix }}
               {%- endfor %}
               JUNOS
```

Workflow Κατανομής Κανόνων: Salt + NAPALM (5/5)

Event -> Reactor -> Apply State -> Template -> **Config**

```
no ipv4 access-list acl-malicious
ipv4 access-list acl-malicious
 1 deny ipv4 host 1.1.1.1 any
 2 deny ipv4 host 4.4.4.4 any
 3 deny ipv4 host 2.2.2.2 any
 permit ipv4 any any
end
```

IOSXR

```
delete policy-options prefix-list malicious
set policy-options prefix-list malicious 1.2.3.4
set policy-options prefix-list malicious 5.6.7.8
set policy-options prefix-list malicious 8.7.6.5
```

JUNOS

Κατανομή Κανόνων Ελέγχου Πρόσβασης: Αξιολόγηση

- Χρόνος απόκρισης για διάφορα OS
  - **Event -> Reactor -> Apply State -> Template -> Config -> Device**

OS / # rules	JUNOS	IOSXR	IOS	VyOS
50	17,6 s	8,6 s	55,3 s	34,2 s
500	22 s	10,6 s	55,2 s	61,7 s
5000	49,1 s	32,4 s	55 s	**266,4 s**

- Εναλλακτικές προσεγγίσεις:
  - Salt Minion εγκατεστημένα σε συσκευές (ARISTA EOS)
  - Πρωτόκολλα / Μηχανισμοί (π.χ. BGP / BGP FlowSpec)

Αδάμ Παυλίδης

apavlidis@netmode.ntua.gr