An Infrastructure for Cyber Agility Defense

Presenter
Md Mazharul Islam
Advisor: Dr. Ehab Al-Shaer
Motivation – Limitations of the state of the art of cyber defense

• Cyber defense techniques are mostly non adaptive and takes long time to respond (hours, months and days).

• Cyber defense techniques are rigid and does not provide agility capability to mitigate threat proactively.

• Cyber defense policies (such as firewalls and IDS) requires too specific description of the attacks signature and lack of the ability to investigate and hypothesize.

• Developing agility on cyber defense is highly complex tasks and it requires significant effort in management and orchestration.
Research Hypothesis

• ActiveSDN/CLIPS can actively investigate and respond to cyber attacks reactively and proactively within seconds.

• ActiveSDN/CLIPS can defend against reconnaissance attack with at least 70% (approx.) effectiveness, meaning 70% of the data collected by the attacker in the reconnaissance phase will be wrong.

• ActiveSDN/CLIPS has the capability to mitigate different class of stealthy DDoS attacks through investigation and configuration actions.
Project Goals: Enabling High-Assurance Auto-Resiliency Policies

The goal of this project is to provide an extensible and verifiable framework for enabling rich and safe auto-resiliency policies. The proposed framework/tool provides the following unique capabilities:

• It provides an expressive and extensible high-level reactive language that allows for integrating investigative, proactive and corrective actions in flexible and composable manner, with full concurrency.

• It allows for "seamless" integration of sensing, sense-making and decision-making actions in a logical workflow to ensure effectiveness of CoA.

• It comprehensively detects and resolves any rule or action conflict due to concurrent and sequential actions execution.

• It provides static and dynamic techniques to ensure that the mission integrity of cyber systems will never be jeopardized by auto resiliency.
Proactive Resiliency has-A Mutation Technique has-A Mutation Parameter is-A Key Based Random Distribution is-A Uniform Distribution is-A Trigger Event triggers Mutation State Change induce Mutation Constraints has-A Mutation Function has-A Mutation Strategy Configuration Parameter dependentOn Routes IP is-A is-A is-A is-A IND Fingerprinting induce Mutation Constraints is-A Cyber Agility Policy for Proactive Defense
Adaptive Cyber Defense for Reactive Resiliency
CLIPS with ActiveSDN

Third Party/Phantom
Policy
CLIPS Policy Interface
CLIPS Translator
CLIPS Verifier
ActiveSDN
ActiveSDN API
OpenDaylight API
Virtual Network
Network
GUI
Policy
Manual
Policy
What is ActiveSDN?

• ActiveSDN is a *decision-making controller* having open interface and engine, that enables implementation of adaptive cyber defense and cyber agility capabilities rapidly and safely on Software Defined Networks.

• ActiveSDN enables users to define the specification of arbitrary sensing and actuation defense functions.

• ActiveSDN translates these specifications into *OpenFlow* configurations that are provably correct, and it automates the deployment of the configuration in the most appropriate devices for enforcement.

• ActiveSDN provides built-in cyber agility NFVs (Network function virtualization) such as IP mutation, route mutation, service migration, continuous monitoring etc.
ActiveSDN Infrastructure

• **ActiveSDN consists of two main components:**
  • **ActiveSDN API:** It provides cyber agility primitives (NFVs) and OpenFlow management functions using the OpenDayLight controller.
  • **Framework:** Decision making engine that is capable of solving computation hard problems using Constraint Satisfaction Solvers and Game Theory to optimize defense actions (Future work).
ActiveSDN (in depth...)

• ActiveSDN Implementation of various CoA:
  • **Configuration CoA:**
    • Flow Access control actions: block, limit, inspect,
    • Cyber Agility actions: reroute, redirect, mutate IP, mutate path, migrate
  • **Investigation CoA:**
    • DDoS actions: CheckElephantTCP, ChekUDPRate, CheckNewComers, etc
    • Monitoring actions: Built in statistics functions for packet drop and network statistics

• Implementation of Case Studies
  • Multi-strategy Active DDoS Mitigation -- reactive (Done)
  • Service Migration with Network Mutation (Done)
  • Risk-based Continues Monitoring and Mitigation (Done)
  • Mutate IP – proactive (Done)
  • Mutate Route – proactive (Done)

• Mapping CLIPS to ActiveSDN API (on-progress)
• Implementation of the CLIPS GUI (paused)
ActiveSDN Basic Features

• Adaptive learning
  • It learns the topology of the network.
  • Dynamically updates the topology and keeps track of any changes in the topology.

• Adaptive Discovery
  • It learns about the hosts connected to the network.
  • Creates a path on the fly when two hosts try to communicate.

• Adaptive Failure Recovery
  • It detects link failures in the network automatically. In the case of any link failure, it identifies all the paths that are affected (become unreachable).
  • It proactively migrates all the affected network paths away from the failed link with minimum packet loss.
  • During this migration, it handles the in-flight packets if any and forwards these to the destination successfully.

• Continues Traffic Monitoring and Measurement
  • Automated Statistics reporting (see all functions in the tables) such as packet drops, delays.

• Topology Querying Analytics
  • Inquiring configuration.
  • Optimize path selection.
  • Multi-homing.
  • Others.
## ActiveSDN Cyber Defense Actions -- Summary

<table>
<thead>
<tr>
<th>Defense Actions</th>
<th>Parameters</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipMutate()</td>
<td>&lt;rIP&gt;</td>
<td>List of initial real IPs, e.g. &lt;192.168.10.20/32, 192.168.10.21/32, 192.168.55.99/32,...&gt; etc.</td>
</tr>
<tr>
<td></td>
<td>when</td>
<td>-1: Deactivate IP mutation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: One time mutation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x: Time based mutation. Mutate IP after x seconds.</td>
</tr>
<tr>
<td></td>
<td>other-mutation</td>
<td>For future use.</td>
</tr>
<tr>
<td></td>
<td>how</td>
<td>specific vIP: User defined virtual IP, e.g. 192.168.60.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>randomFunction(): A function which will provide random vIPs in a specific time window x.</td>
</tr>
<tr>
<td>Example</td>
<td>ipMutate(10.0.0.2/32, 5, uniform())</td>
<td></td>
</tr>
</tbody>
</table>
## ActiveSDN Cyber Defense Actions -- Summary

<table>
<thead>
<tr>
<th>Defense Actions</th>
<th>Parameters</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>pathMutate()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;src&gt;</td>
<td>List of source Host IPs, e.g. &lt;192.168.10.20/32, 192.168.10.75/32, 192.168.55.99/32, ...&gt; etc.</td>
<td></td>
</tr>
</tbody>
</table>
| <dst>           | List of destination Host IPs, e.g. <192.168.10.20/32, 192.168.10.75/32, ...> etc.  
**Note:** (src[i], dst[i]) must be the end hosts of a complete path, where src[i] and dst[i] is the i'th element of each list. |
| pathProfile     |            |              |
| pathProfile, P = (v,n,B,R) | Example:  
\[ v = \text{overlap} \]  
\[ n = \text{number of links in a path} \]  
\[ B = \text{Maximum bandwidth} \]  
\[ R = \text{Maximum Risk threshold} \]  | overlap  
0: No overlap.  
percentage: How much overlap is acceptable.  
excludeSpecificLinks: Exclude any specific link form the path  
includeSpecificLinks: Include any specific link form the path |
| maxPathLength   | User can provide the maximum path length which can be used as new path. |
| availableBandwith | User can provide the maximum bandwidth of each link in a path. |
| maxRisk = R, where (1 - (1-p)^n) < R \( (p: \text{prob that a link is under attack}) \) | For Each link of any path containing n links, the probability of that link is under attack is p. |
| trigger         | -1: Deactivate path mutation.  
0: Immediate mutation.  
x: Temporal mutation. Mutate path after every x seconds.  
conditional: For future use. |
| Example         | pathMutate(<192.168.10.20/32>, <192.168.60.35/32>, P, 0.4) |
ActiveSDN Case Studies
ActiveSDN Example 1: NFV Support

**F₁** Forward Traffic and Keep Statistics

**F₂** Forward and inspect for Events

**F₃** Filter/Split attack traffic

Create new path & Isolate traffic
ActiveSDN Example 2: Defense against Crossfire Attack

Identifies the critical links for a VN that should be migrated to avoid indirect DDoS attack.
Multi-strategy Active DDoS Mitigation--Revised

Event
Link L is flooded

High rate UDP flows?

Block the UDP

No

High rate TCP Elephant flows?

Block all elephant flows

Yes

High # of New comers?

No

White-listed?

Yes

Limit non-whitelisted

No

Replicate/migrate Services and Absorb

Yes

Reroute, or Forward to Low Priority Queue
DDoS Mitigation Policy: CLIPS to SDN

(Link-Flooded(T,L)) →

IF (DO CheckICMPUDP ON flows BY IDS USING rate > 50%
   OUTCOME P && P <> 0) THEN
   DO Block ON flows BY Switch<1.1.1.1>
   USING proto=ICMP or UDP FOR PREVENT

ELSE IF (DO CheckerForElephant ON flows BY IDS-App USING rate>90%
   FOR DETECT OUTCOME E && E <> 0) THEN
   DO Block ON flows BY FIREWALL<1.5.6.4, “admin”>
   USING flow ∈ E FOR PREVENT

ELSE IF (DO CheckNewComers ON flows BY IDS-App USING window < 1
   OUTCOME N, rate && rate > 75%) THEN
   DO Re-reoute ON flows BY ROUTER USING src_ip ∈ N

ELSE IF (DO CheckWhiteListed ON flows BY IDS USING src_ip ∈ WHITE-LIST
   OUTCOME W && W <> 0) THEN
   DO limit ON flows BY ROUTER USING src_ip ∈ W

ELSE
   DO replicate ON services USING loc /* create new cluster*
   DO reroute ON flows USING Switch /* reroute some traffic to it */
Cyber Agility – IP Mutation and Path Mutation

Requirements:
- Safe
- Transparent
- Random/high Entropy
- Fast
For $H_3$
\[ rIP = H_3 \]
\[ vIP = U - H_3 \]
where $U = \{\text{All IP Spaces}\}$

For $H_6$
\[ rIP = H_6 \]
\[ vIP = U - H_6 \]
where $U = \{\text{All IP Spaces}\}$

Path in between $H_3$ and $H_6$ created

IP Mutation Example

Set of hosts

Controller

$\langle \text{Src} = H_3, \text{Dst} = H_6 \rangle$

$\langle \text{www.xyz.com, H}_6 \rangle$

$\langle \text{www.xyz.com, H}_6 \rangle$
For $H_3$
- $rIP = H_3$
- $vIP = U - H_3$
where $U = \{\text{All IP Spaces}\}$

For $H_6$
- $rIP = H_6$
- $vIP = U - H_6$, where $U = \{\text{All IP Spaces}\}$

Installing flow rules:
- $<\text{Src }= H_3, \text{Dst }= (H_8 \rightarrow H_6), \text{Action }= \text{Forward}>$
- $<\text{Src }= (H_6 \rightarrow H_8), \text{Dst }= H_3, \text{Action }= \text{Forward}>$
- $<\text{Src }= *, \text{Dst }= H_6, \text{Action }= \text{Deny}>$

$<\text{Src }= H_3, \text{Dst }= H_8>$
$<\text{Src }= H_*, \text{Dst }= H_6>$
$<\text{Src }= H_3, \text{Dst }= H_8>$

IP Mutation Example

Dropped