IEEE SecDev | 2020

IEEE COMPUTER SOCIETY

Active Deception Framework: An Extensible Development Environment for Adaptive Cyber Deception

Md Mazharul Islam* and Ehab Al-Shaer+

*University of North Carolina at Charlotte

+Carnegie Mellon University

September 28-30, 2020

#IEEESecDev https://secdev.ieee.org/2020

Motivation

- In cyber warfare, there is an **asymmetry** between the adversary and the defender
  - Defenders need to protect all susceptibilities into the infrastructure.
  - Adversary needs one vulnerability to exploit.
- Existing attack *prevention and detection* techniques have major limitations:
  - The ***time window*** between discovering a vulnerability and applying patches is long, sometimes around 16 days to 6 months.
  - Patching is hard, on average at most 14% of the vulnerable hosts get patched.
  - Skilled attackers can easily avoid static signature-based detection.
- Therefore, a **proactive approach** must be used by defenders to break the game.
- Cyber deception is a promising technology to achieve this goal.

Cyber Deception

- Cyber Deception is an *intentional misrepresentation* of real systems' ground truth to manipulate adversary's course of actions under the premises of the defender's rules.
- Deception can be used to
  - *Divert* adversary away from their target to false or no target.
  - *Distort* their perception of the infrastructure by adding ambiguity and decoys into the network.
  - *Deplete* adversary by consuming their computational power to delay attack propagation.
  - *Discover* their hidden tactics and techniques, by letting them run into honey environment.
- It is expected that the global cyber deception market's expense will grow up to $2.3 billion by 2022.

Problem Description

- Developing cyber deception techniques in real networks is a highly complex task.
- It requires significant effort in implementation and network configuration management.
- Efficient and adaptive cyber deception needs
  - Continuous network monitoring to observe adversary activities.
  - Optimal planning for feasible implementation.
  - Safe deployment without breaking the integrity of the system.
- As a result, few deception frameworks are developed and validated in the real-life operational environment.

Our Approach

- We develop an **Active Deception Framework (ADF)** to build sophisticated cyber deception applications.
- The goal of ADF is to make deception infrastructure as services through high-level APIs to abstain deception architects from intricate details of low-level deception primitives:
  - **Implementation.**
  - **Orchestration.**
  - **Safe deployment.**
- ADF provides an open environment for developing deception by
  - An extensible rich API for developing deception techniques.
  - A decision-making synthesis engine for optimizing deception planning.
  - A controller for automated orchestration and deployment of deception techniques implementation.

ADF API

- The novelty of ADF is the extensible rich API sets
  - Deception APIs: create various deception functions and applications.
  - Sensor APIs: monitor adversary activities in the system.
  - Management APIs: configure cyber resources such as switches, links, hosts, services, etc. to orchestrate deception operation.
  - Constraints APIs: APIs for defining constraints to optimize honey networks such as risk, rerouting, reachability, availability constraints while deploying honey resources.

Deception API

Name	Descriptions
createHoneyNetwork()	Dynamically creates a honey network with decoy/shadow hosts and services to analyze adversary for unknown TTP discover or distort them to delay attack propagation.
reDirect()	Redirect traffics to a given destination (can be a decoy or false target) and tunnel the packet to a proxy to generated trusted response.
reRoute()	Change the old path between a source and destination pair to a new path to avoid possible link flooding or other security measures.
routeMutate()	Change the route frequently of active flow(s) to another satisfiable route based on event or time.
hostMutate()	Randomizing real src/dst IP addresses to virtual src/dst IP addresses for depletion, so that real IP is used for routing but end hosts always uses virtual IP to communicate.
migrateService()	Create new machine with same services of the current target then migrates all benign traffic to the new machine.
spatioTemporalMutation()	Randomize the real IP of given hosts so that each host reach the same destination with a different IP address. Therefore, the view of the network is different for different host.
createShadow()	Creates an identical fingerprint (shadow) of a given host in the honeypot.
createDecoy()	Creates a decoy host. If the decoy is specified for a target host without specifying any services, then arbitrary but the same type of services will be created in the decoy, e.g., an FTP server but with a different vendors.

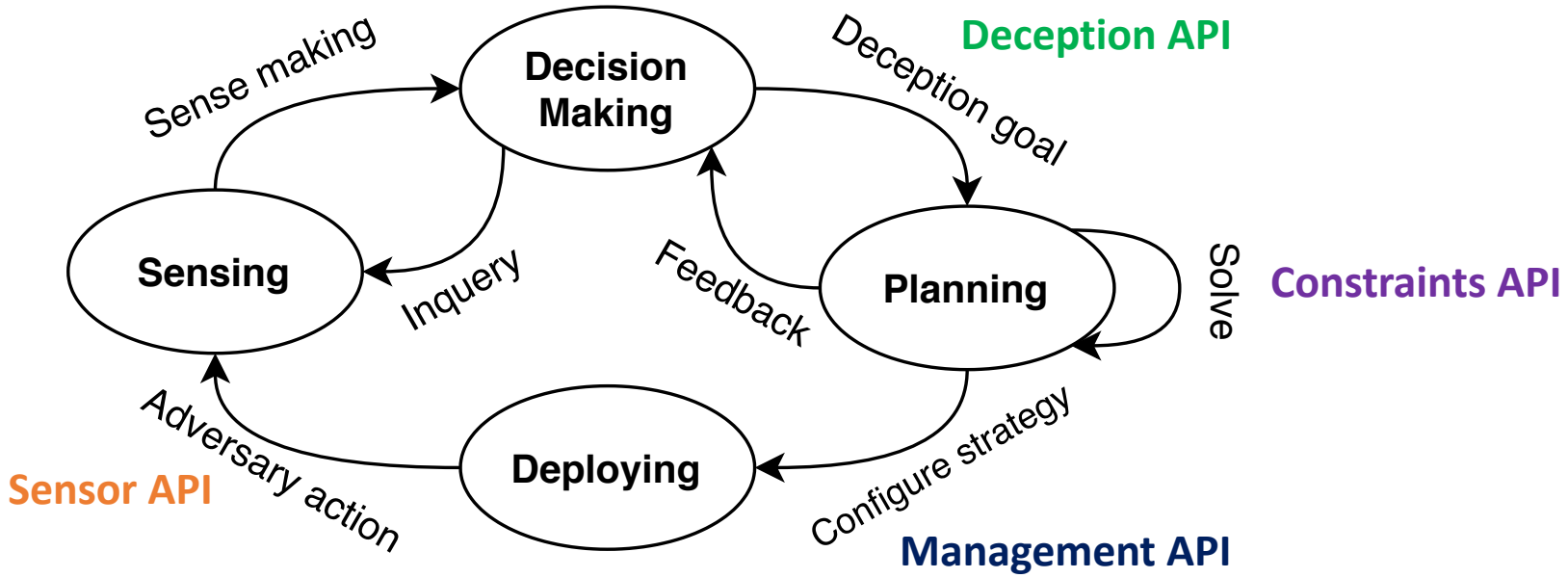
Other APIs

Sensor API	Management API	Constraints API
isHostScanning()	block()	getRouteRisk()
isLinkFlooding()	inspect()	overlap()
chekTrafficRate()	throttling()	isIncludeSwitch()
checkElephantTCP()	splitInspect()	getAvailableBandWidth()
getFlowStatistics()	priorityForwarding()	checkUniqueIP()
checkNewComers()	installFlowRule()	checkNonRepeateIP()
getCriticalLinks()	installNetworkPath()	checkSpatialCollision()
getAllFlowRules()	sendPacketOut()	getMinDetectionProb()
findNeighbors()	createTunnel()	getAttackUncertainity()
detectBot()	subscribeEvent()	canReach()
getPortID()	removeAllFlows()	getShortestPath()

ADF Framework

- We developed ADF over Software-defined networking (SDN).
- SDN provides a programmable environment over network configuration management through a centralized controller.
- Enables comprehensive diagnosis of observations and quick deception action response.

Active Deception Strategy



Case Study

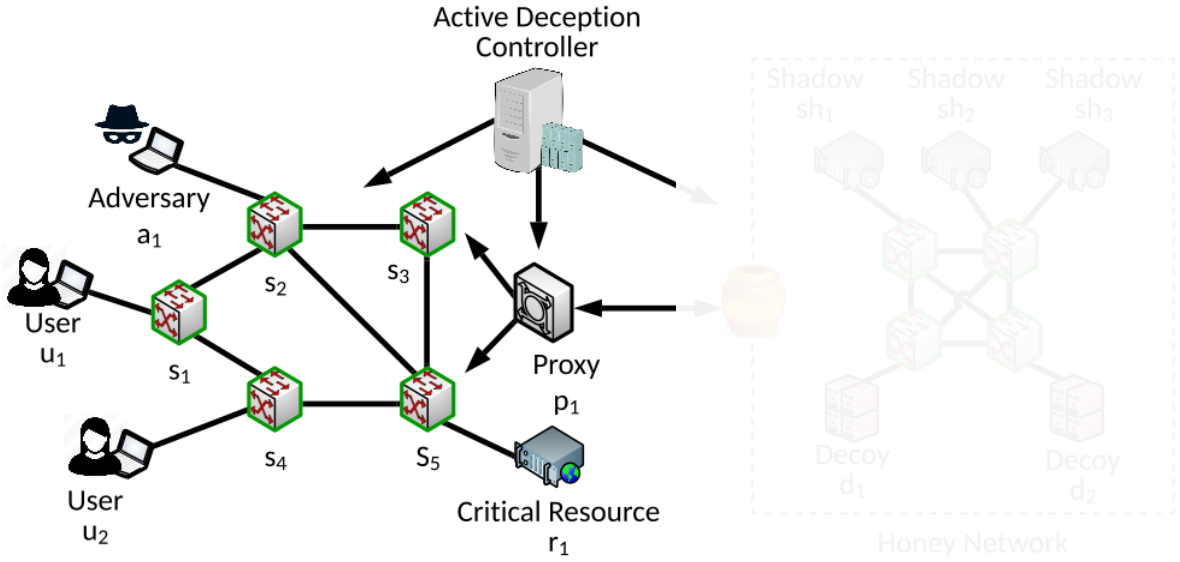
Distortion and discovery by creating HoneyNetwork

- Deception API: ***createHoneyNetwork()***
- It creates a honey network with shadows and decoys of a given target to protect it from reconnaissance attack
- API

Param	Descriptions
target	The critical resources (hosts, services, links, etc.) to defend.
impact	Impact of the critical resources. (low, medium or high).
k	To anonymize fingerprinting, *k-anonymity* places $(k-1)$ shadow host with identical fingerprinting of the target host.
l	To anonymize configuration, *l-diversity* places $(l-1)$ fake services of same software type but different versions/vendors.
trigger	*activate*: Activate generated honey network.
	deactivate: Deactivate and remove honey network.

- k-anonymization places (k−1) shadow host with identical fingerprinting of the target host.
- l-diversity places (l − 1) decoy host with fake services of the same software type but different versions/vendors.

Distortion and discovery by creating HoneyNetwork



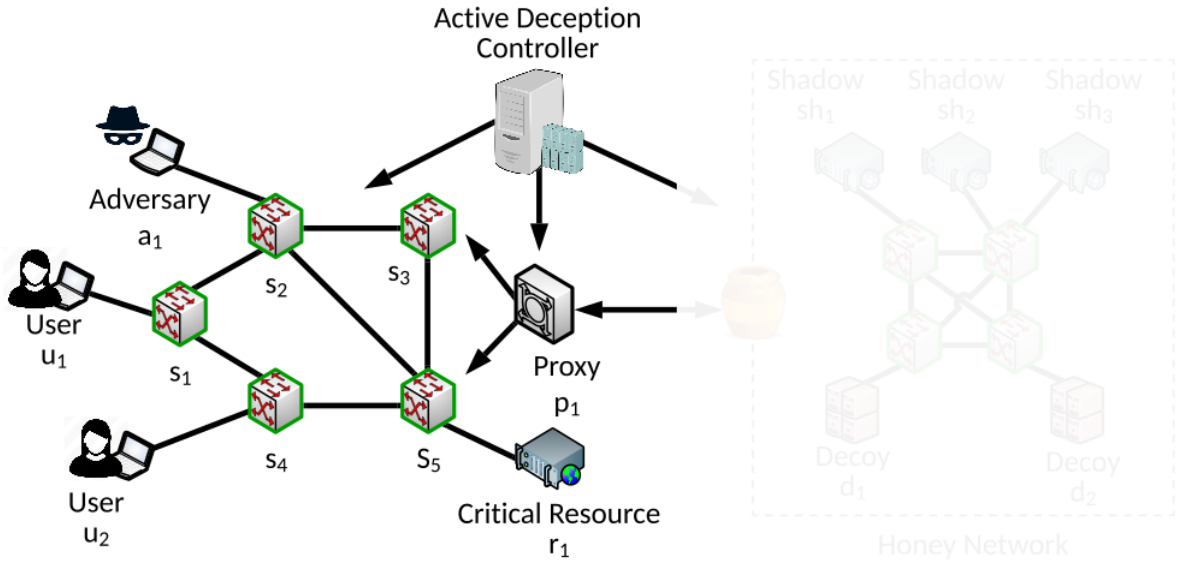

```
{"input": {
        "api": "createHoneyNetwork",
        "target": "r1",
        "impact": "high",
        "k": 2,
        "l": 3,
        "trigger": "activate"
    }
}
```

Distortion and discovery by creating HoneyNetwork

Host	OS	Services		
r_1	Ubuntu	Vsftpd-2.3.5	Apache-2.2.22	MySQL-5.5.54

```
{"input": {
        "api": "createHoneyNetwork",
        "target": "r1",
        "impact": "high",
        "k": 2,
        "l": 3,
        "trigger": "activate"
    }
}
```

2-anonymity
3-diversity



Distortion and discovery by creating HoneyNetwork

```
Vagrant.configure("2") do |config|
   config.vm.define "shadow_1" do |shadow_1|
       shadow_1.vm.box = "hashicorp/precise64"
       shadow_1.vm.network "public_network", bridge: "Ethernet",
           ip: "10.38.60.2", netmask:"255.255.224.0"
       shadow_1.vm.provision "shell", inline: "sudo apt-get -y
           install vsftpd=2.3.5"
       shadow_1.vm.provision "shell", inline: "sudo apt-get -y
           install apache2=2.2.22"
       shadow_1.vm.provision "shell", inline: "sudo apt-get -y
           install mysql-server=5.5.54"
   ...
```

Fig: Vagrant configuration script

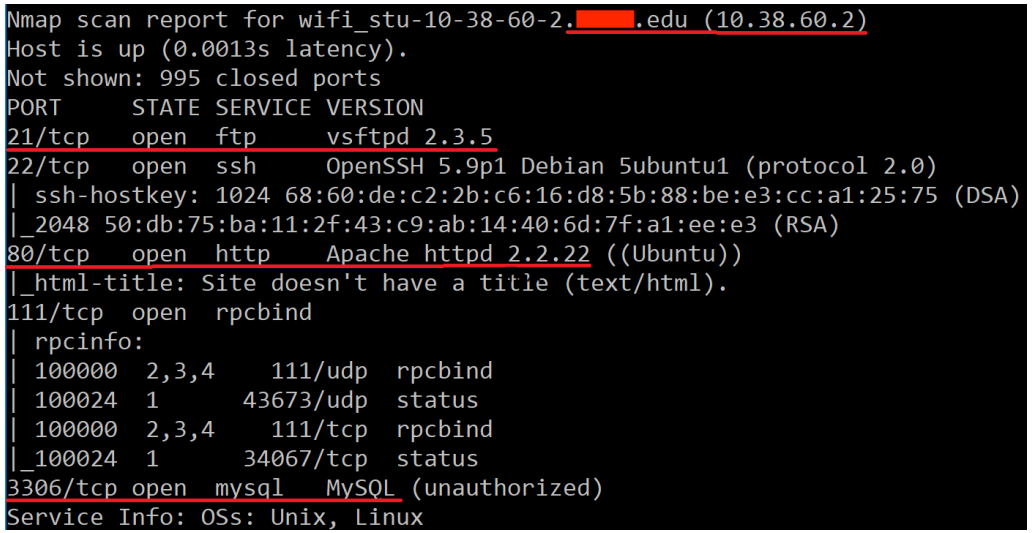


Fig: Nmap scanning result

Depletion using Spatio-temporal Mutation

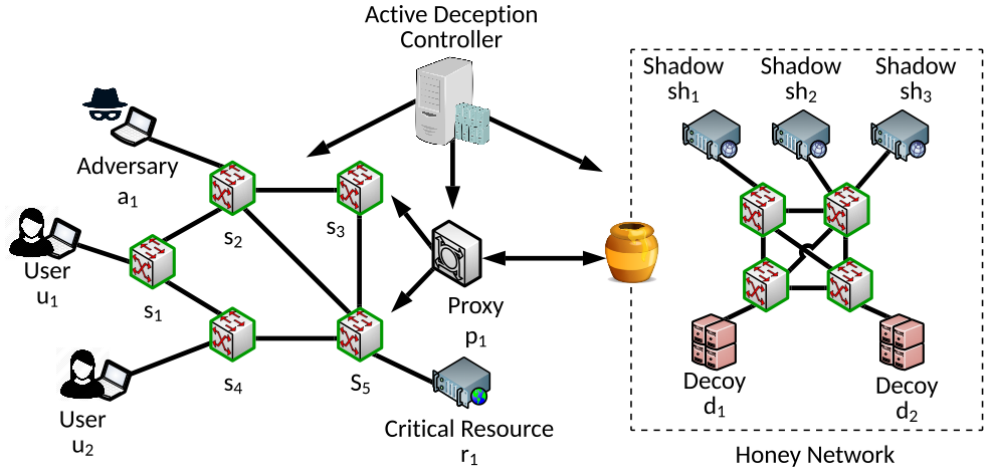
- Deception API: *spatioTemporalMutation()*
- It changes the static view of the network by periodically mutation the static real IP addresses to short lived virtual IP addresses.
- Therefore, adversary needs to increase their probing to find the target
- API

Param	Descriptions
h	Target host list for spatial mutation.
eIP	List of ephemeral IP addresses. (Optional)
m_i	***eIP*** collision rate where $i \in h$
t	Lifespan of ***eIP*** (temporal period).
how	***eIP*** distribution fucntion, can be uniform or random

Depletion using Spatio-temporal Mutation

	Real IP	eIP	
u_1	10.0.0.1	10.0.0.10	10.0.0.11
u_2	10.0.0.2	10.0.0.8	10.0.0.9
r_1	10.0.0.3	10.0.0.6	10.0.0.7

Table: Ephemeral IP assignment with real IP



	$u_1(10.0.0.1)$	$u_2(10.0.0.2)$	$r_1(10.0.0.3)$
$u_1(10.0.0.1)$	-	10.0.0.10	10.0.0.11
$u_2(10.0.0.2)$	10.0.0.8	-	10.0.0.9
$r_1(10.0.0.3)$	10.0.0.7	10.0.0.6	-

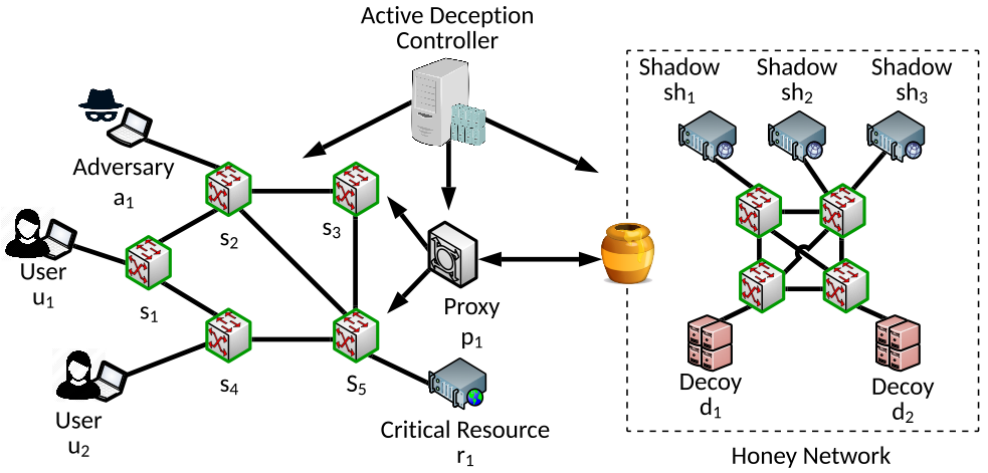
Table: Forwarding entry mapping with real IP and eIP

Depletion using Spatio-temporal Mutation

	$u_1(10.0.0.1)$	$u_2(10.0.0.2)$	$r_1(10.0.0.3)$
$u_1(10.0.0.1)$	-	10.0.0.10	10.0.0.11
$u_2(10.0.0.2)$	10.0.0.8	-	10.0.0.9
$r_1(10.0.0.3)$	10.0.0.7	10.0.0.6	-

Table: Forwarding entry mapping with real IP and eIP

Flow	Rule

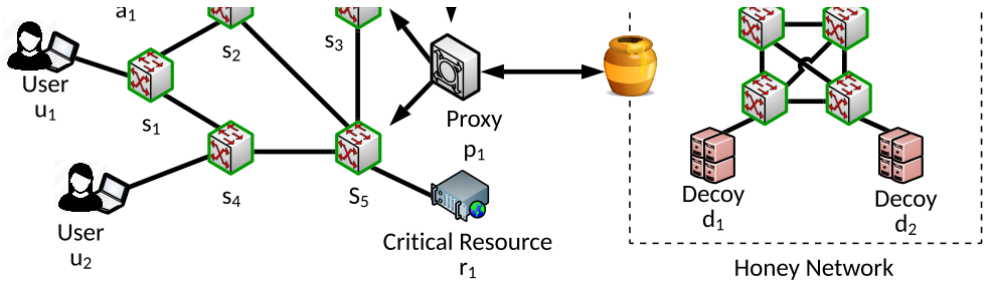



```
set_field:0->ip_dscp
cookie=0x6d, duration=1379.915s, table=0, n_packets=0, n_bytes=0, priority=400,ip,nw_src=10.0.0.3,nw_dst=10.0.0.11 actions=set_field:10.0.0.1->ip_dst,set_field:e2:b3:16:8c:34:d2->eth_dst,output:3
cookie=0x6c, duration=1379.920s, table=0, n_packets=0, n_bytes=0, priority=400,ip,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=set_field:10.0.0.11->ip_src,output:1,set_field:0->ip_dscp
cookie=0x5c, duration=1380.036s, table=0, n_packets=0, n_bytes=0,
```

Depletion using Spatio-temporal Mutation

	$u_1(10.0.0.1)$	$u_2(10.0.0.2)$	$r_1(10.0.0.3)$
$u_1(10.0.0.1)$	-	10.0.0.10	10.0.0.11
$u_2(10.0.0.2)$	10.0.0.8	-	10.0.0.9
$r_1(10.0.0.3)$	10.0.0.7	10.0.0.6	-

Table: Forwarding entry mapping with real IP and eIP



Flow	Rule
$u_1 \rightarrow r_1$	src=10.0.0.1, dst=10.0.0.3 → set_src:10.0.0.11
$r_1 \rightarrow u_1$	src=10.0.0.3, dst=10.0.0.11 → set_dst:10.0.0.1

Flow	Rule

```
set_field:0->ip_dscp
 cookie=0x6d, duration=1379.915s, table=0, n_packets=0, n_bytes=0, priority=400,ip,nw_src=10.0.0.3,nw_dst=10.0.0.11 actions=set_field:10.0.0.1->ip_dst,set_field:e2:b3:16:8c:34:d2->eth_dst,output:3
 cookie=0x6c, duration=1379.920s, table=0, n_packets=0, n_bytes=0, priority=400,ip,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=set_field:10.0.0.11->ip_src,output:1,set_field:0->ip_dscp
 cookie=0x5c, duration=1380.036s, table=0, n_packets=0, n_bytes=0,
```

```
Every 3.0s: sudo ovs-ofctl dump-flows -OOpenFlow13 s1

OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x6e, duration=1220.462s, table=0, n_packets=0, n_bytes=0, priority=400,ip,nw_src=10.0.0.3,nw_dst=10.0.0.1 actions=set_field:10.0.0.7->ip_src,output:1,set_field:0->ip_dscp
 cookie=0x6b, duration=1220.480s, table=0, n_packets=0, n_bytes=0, priority=400,ip,nw_src=10.0.0.1,nw_dst=10.0.0.7 actions=set_field:10.0.0.3->ip_dst,set_field:f2:5c:62:05:ca:c9->eth_dst,output:3
 cookie=0x5b, duration=1220.507s, table=0, n_packets=0, n_bytes=0,
```

Deflection by redirection

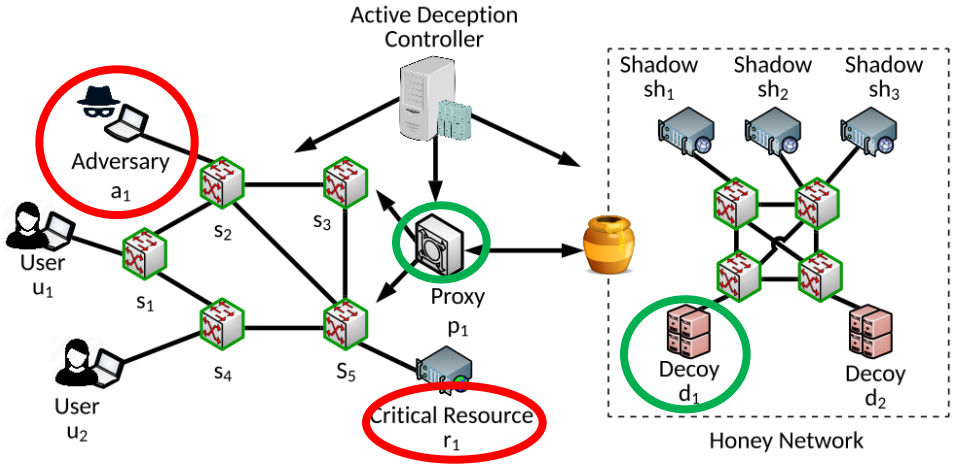
- Deception API: *reDirect() and reRoute()*
- API

	Param	Descriptions
reDirect()	***src***	Source host IP or flow ID.
	dst	Destination host IP or flow ID.
	to	The redirection destination, can be a switch, host, IDS or even the controller.
reRoute()	***src***	Source host IP or flow ID.
	dst	Destination host IP or flow ID.
	to	A new route consist of switches between ***src*** and ***dst*** e.g., s_1, s_2, s_4, s_9.

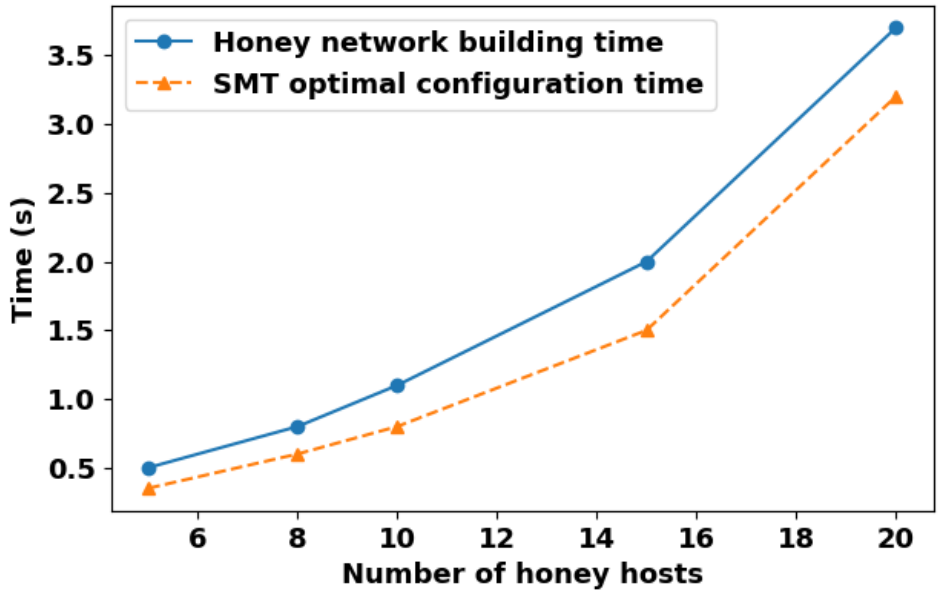
Deflection by redirection

- Redirect installs the following rules:

1. *src=*, dst=IPr1 , set_dst:IP_{p1}*
2. *(src=IP_{a1}, dst=IP_{r1}) → (src=IP_{p1} , dst=IP_{d1})*
3. *(src=IP_{d1} , dst=IP_{p1}) → (src=IP_{r1} , dst=IP_{a1})*

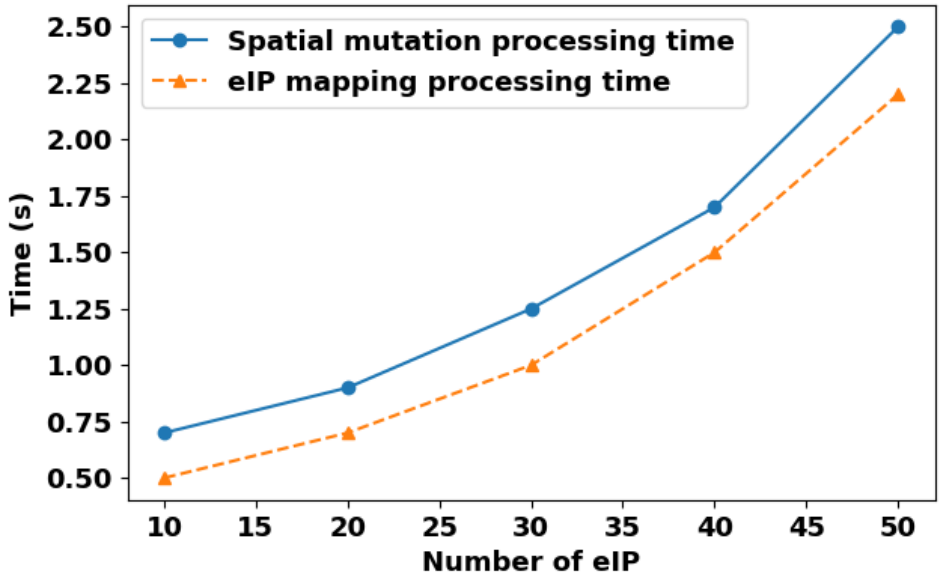


Evaluation: Honey network creation overhead



- We compare the total processing delay and the SMT solve time for Honey Network creation time.
- It takes around 3.7 seconds to create a honey network with twenty shadow and decoy hosts.

Evaluation: Spatial mutation overhead



- We calculated the total processing delay for spatio temporal mutation over different eIP.
- For a spatial mutation with fifty eIP, ADF requires 2.6 seconds to install all necessary flow rules into the network.

Conclusion & Future Works

- We present an **Active Deception Framework (ADF)** that enables an open environment for developing sophisticated cyber deception applications.
- ADF leverages an extensive deception API that can be used to build multi-strategy deception policies.
- We show different case studies by developing various goal oriented deception strategies.
- ADF incurs very little system overhead while providing proactive defense by deception.
- We plan to include more sophisticate optimization techniques such as POMDP.
- Integrate various types of honey things such as, honey applications, honey webpages, and more.
- Deceive different other classes of APT such as malware.

Thank You