Mario Di Mauro

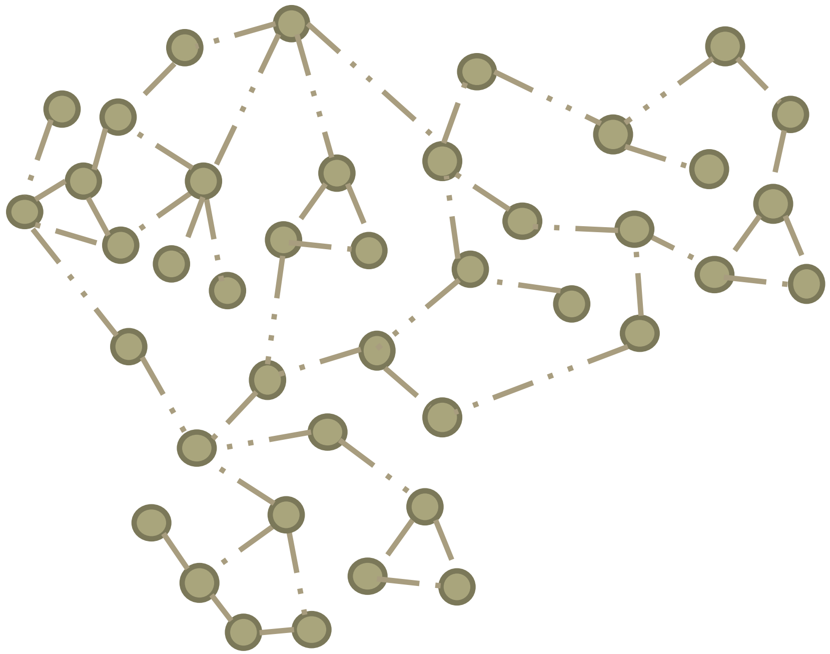
Statistical Models for the Characterization, Identification, and Mitigation of Distributed Attacks in Data Networks

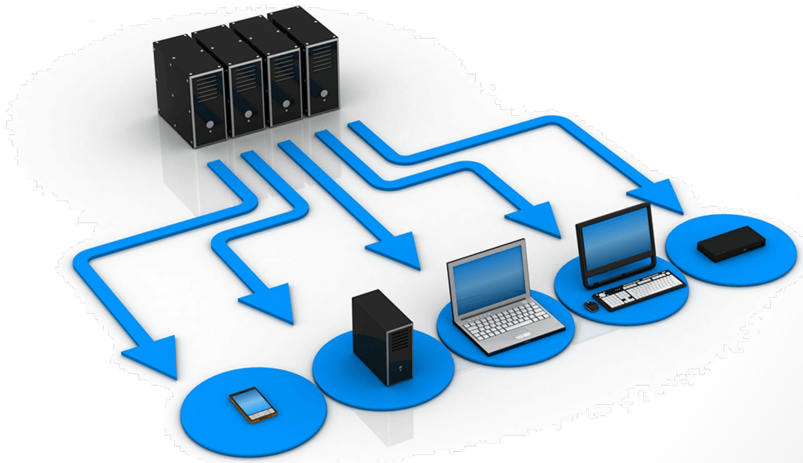
Advisor: Prof. Maurizio Longo

Mario Di Mauro – University of Salerno

Associazione Gruppo Telecomunicazioni e Tecnologie dell'Informazione

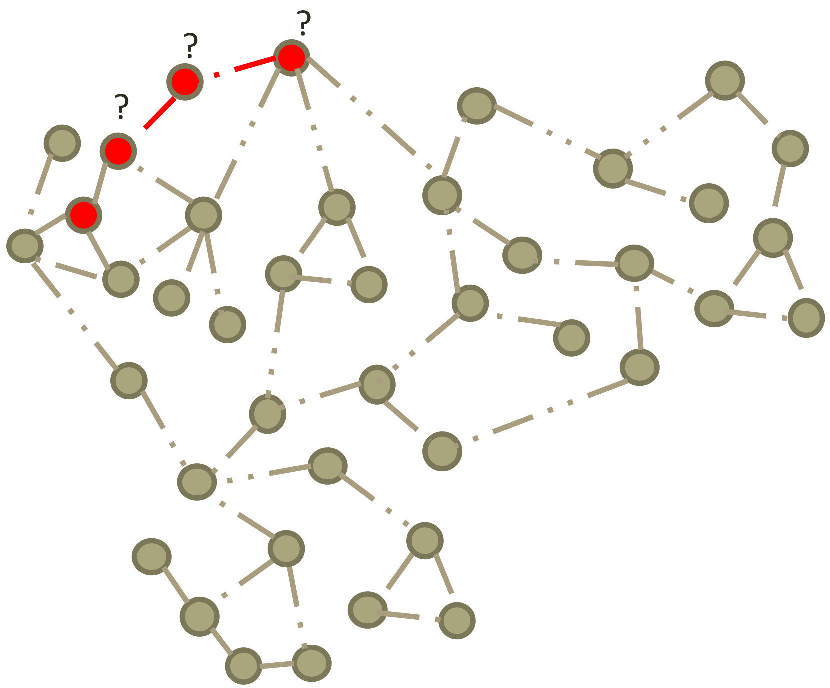
Three critical challenges of distributed cyber-attacks

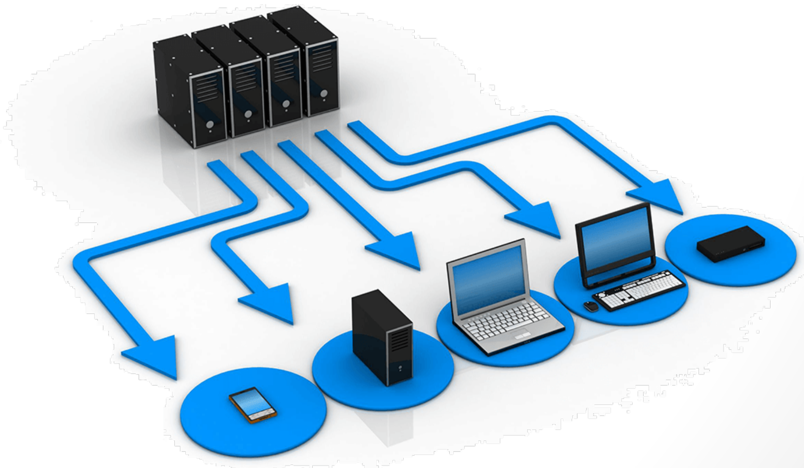




Three critical challenges of distributed cyber-attacks

1. Identifying and banning the sources of the cyber-attack (e.g., the bots in a Distributed Denial-of-Service)



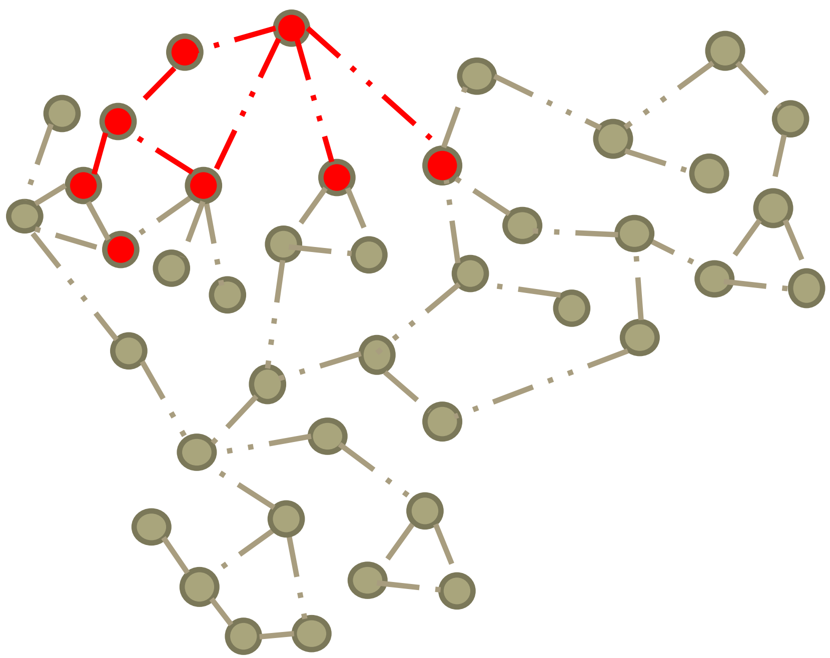


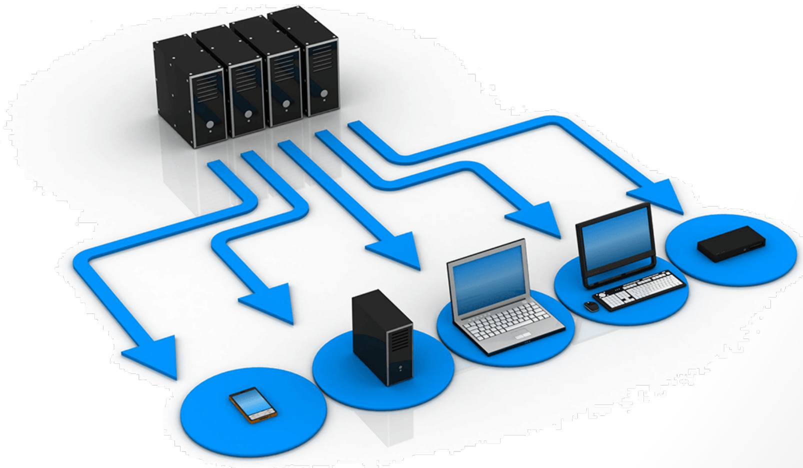
Mario Di Mauro – University of Salerno

Three critical challenges of distributed cyber-attacks

1. Identifying and banning the sources of the cyber-attack (e.g., the bots in a Distributed Denial-of-Service)

2. Containing the spreading of a cyber-threat (e.g., a virus or a malware)

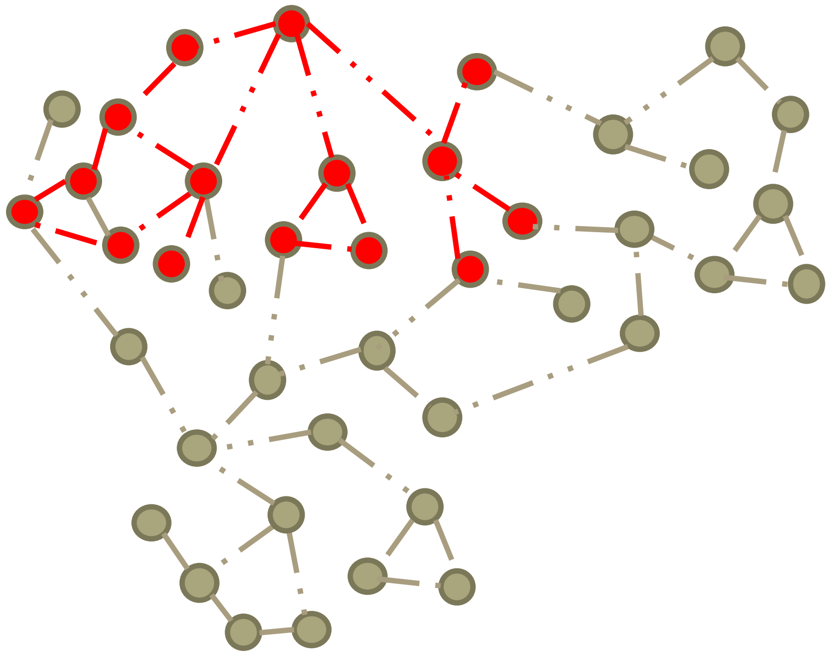


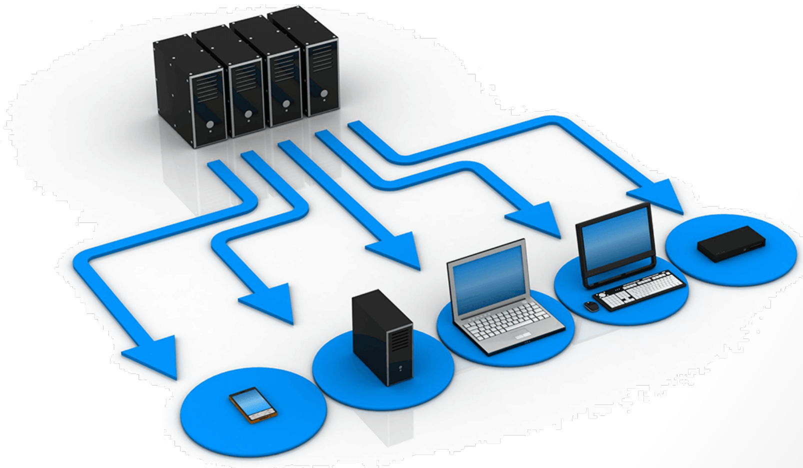


Three critical challenges of distributed cyber-attacks

1. Identifying and banning the sources of the cyber-attack (e.g., the bots in a Distributed Denial-of-Service)

2. Containing the spreading of a cyber-threat (e.g., a virus or a malware)

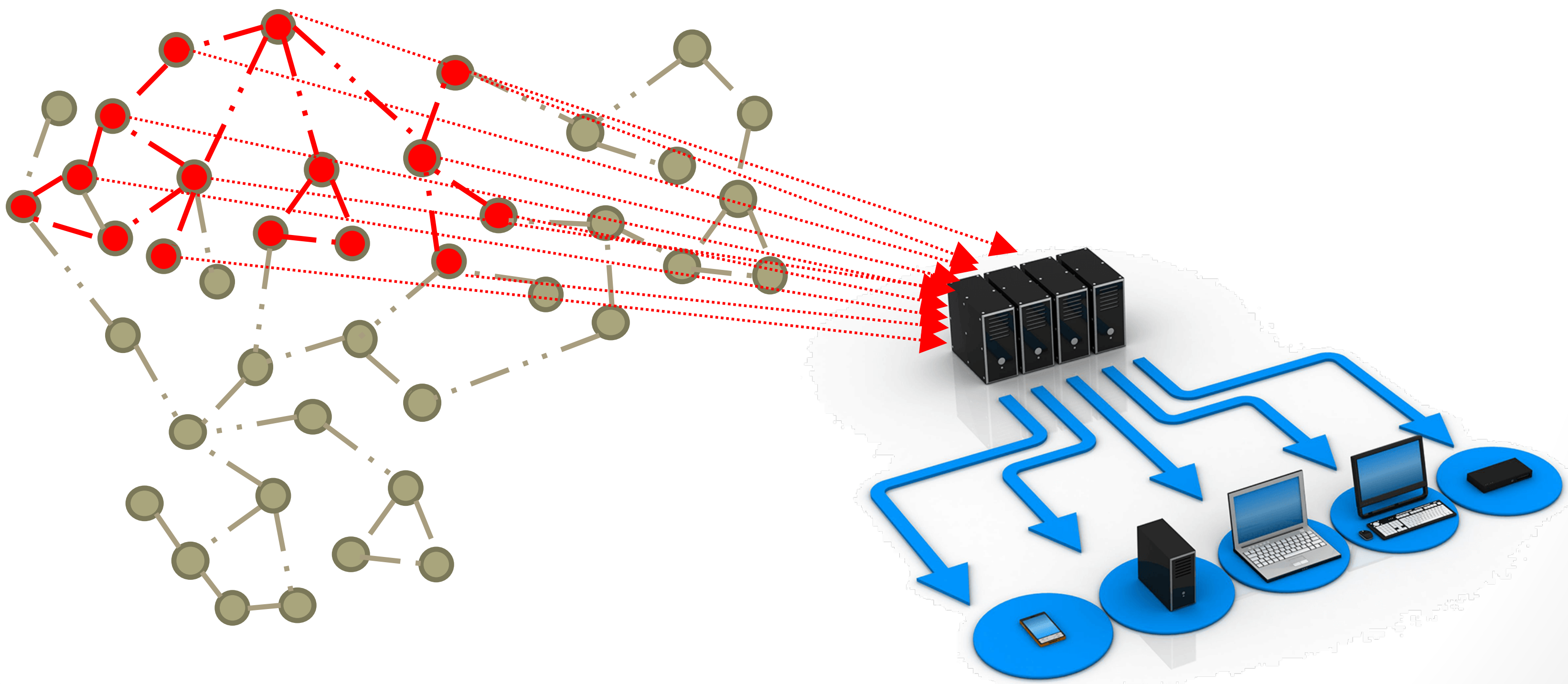




Three critical challenges of distributed cyber-attacks

1. Identifying and banning the sources of the cyber-attack (e.g., the bots in a Distributed Denial-of-Service)

2. Containing the spreading of a cyber-threat (e.g., a virus or a malware)

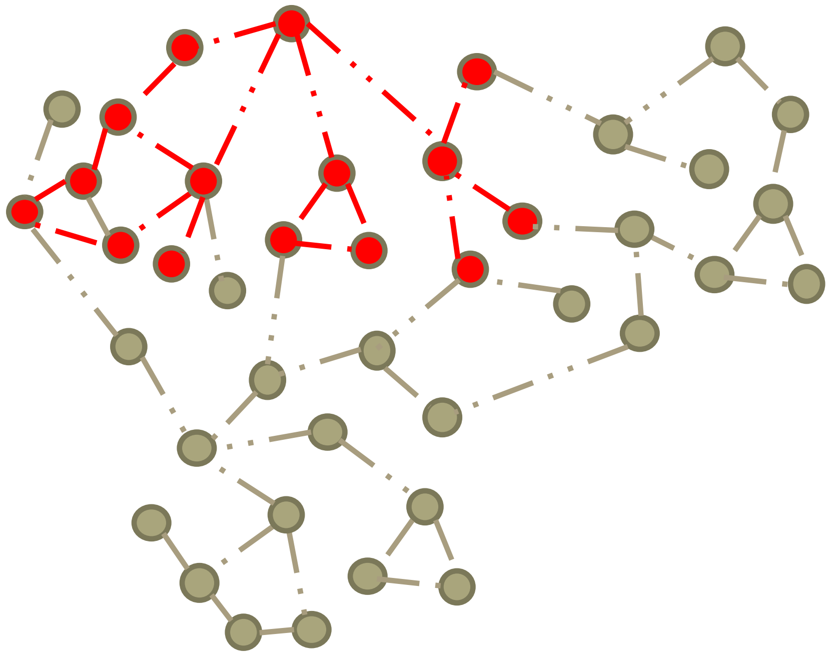


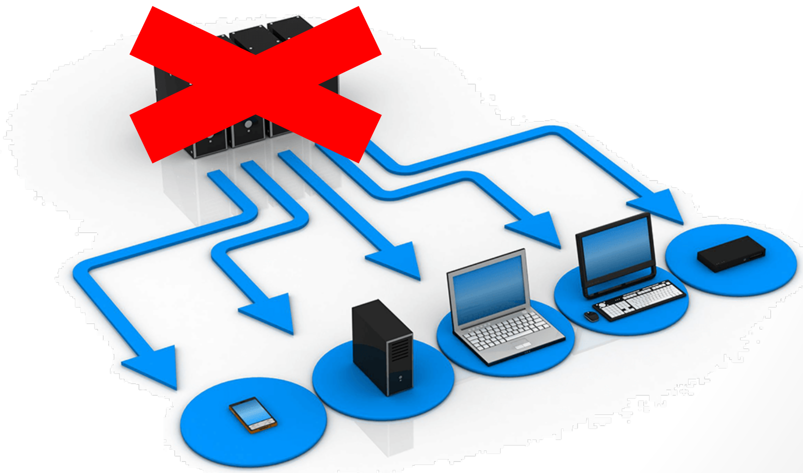
Mario Di Mauro – University of Salerno

Three critical challenges of distributed cyber-attacks

1. Identifying and banning the sources of the cyber-attack (e.g., the bots in a Distributed Denial-of-Service)

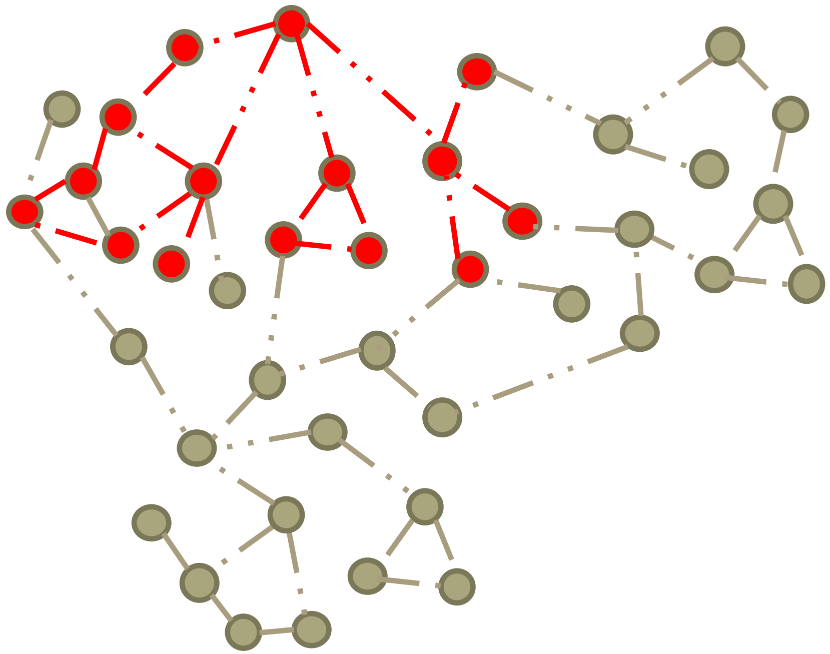
2. Containing the spreading of a cyber-threat (e.g., a virus or a malware)





Mario Di Mauro – University of Salerno

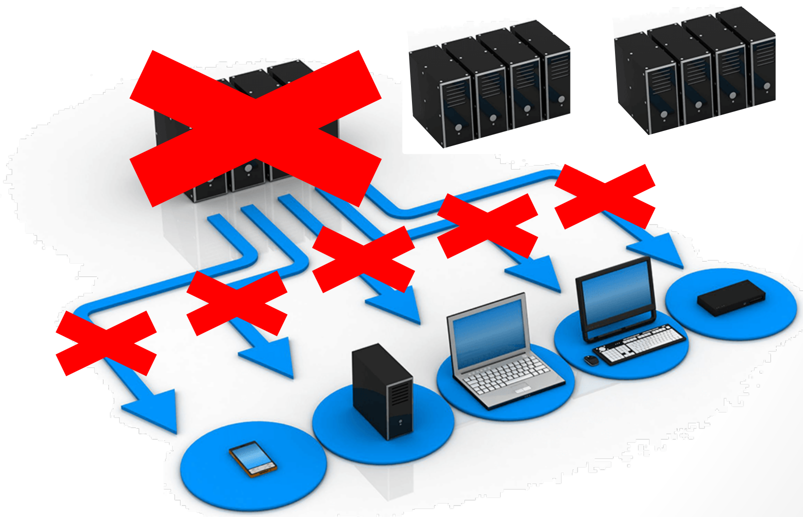
Three critical challenges of distributed cyber-attacks



1. Identifying and banning the sources of the cyber-attack (e.g., the bots in a Distributed Denial-of-Service)

2. Containing the spreading of a cyber-threat (e.g., a virus or a malware)

3. Adding controlled network redundancy in view of some defeat (e.g., a network node crashes)



Mario Di Mauro – University of Salerno

Main Contributions

Proposed solution: inferential strategies to detect, identify, and mitigate the distributed attacks

1. **Formal Characterization** of a distributed attack in a randomized setting[1]
    - Botnet model with randomized emulation of legitimate traffic
    - *Designed-from-the-scratch* algorithm for hidden botnet identification
2. **Analytical Model** of the attack spreading phenomenon[2]
    - Kendall's Birth-Death-Immigration model to formalize a spreading attack
    - Optimal *curing* resource allocation for attack mitigation
3. **Stochastic Techniques** for prevention measures
    - Modeling network resilience against attacks
    - Stochastic approaches: SRN (Stochastic Reward Nets) and original extension of UGF (Universal Generating Function) - Multidimensional UGF (MUGF)[3]

[1]Matta V., Di Mauro M., Longo M., *DDoS Attacks with Randomized Traffic Innovation: Botnet Identification Challenges and Strategies*, IEEE Transactions on Information Forensics and Security, Vol. 12, n°8, Aug.17, pp. 1844-1859

[2]Matta V., Di Mauro M., Longo M., Farina A. *Cyber-Threat Mitigation Exploiting the Birth-Death-Immigration Model,* IEEE Transactions on Information Forensics and Security, Vol. 13, n°12, Dec. 2018, pp. 3137-3152

[3]Di Mauro M., Longo M., Postiglione F. *Availability Evaluation of Multi-tenant Service Function Chaining Infrastructures by Multidimensional Universal Generating Function, submitted on* IEEE Transactions on Services Computing

I. Novel Class of Randomized DDoS Attack

DoS (Denial of Service) attack: "volumetric" attack where a target site is overwhelmed with a huge request rate by a single node.

Distributed DoS attack (**DDoS**): a huge number of apparently innocuous requests is produced in parallel by a net of robots (*Botnet*) coordinated by a Controller (*Botmaster*).

- Hard to identify single nodes of a Botnet
- It is one of the most critical threats to face

Key Idea: designing an "enhanced DDoS attack" where:

The *Botnet* emulates the regular traffic patterns (application layer) by gleaning admissible messages from an "emulation dictionary" (that becomes richer and richer as time elapses) built by the Botmaster during a collection phase to evade detection

Experiments have been carried out in a realistic testbed set up in CoRiTel (Consortium Research on Telecommunication) LAB

The Botnet Identification Condition (BIC)

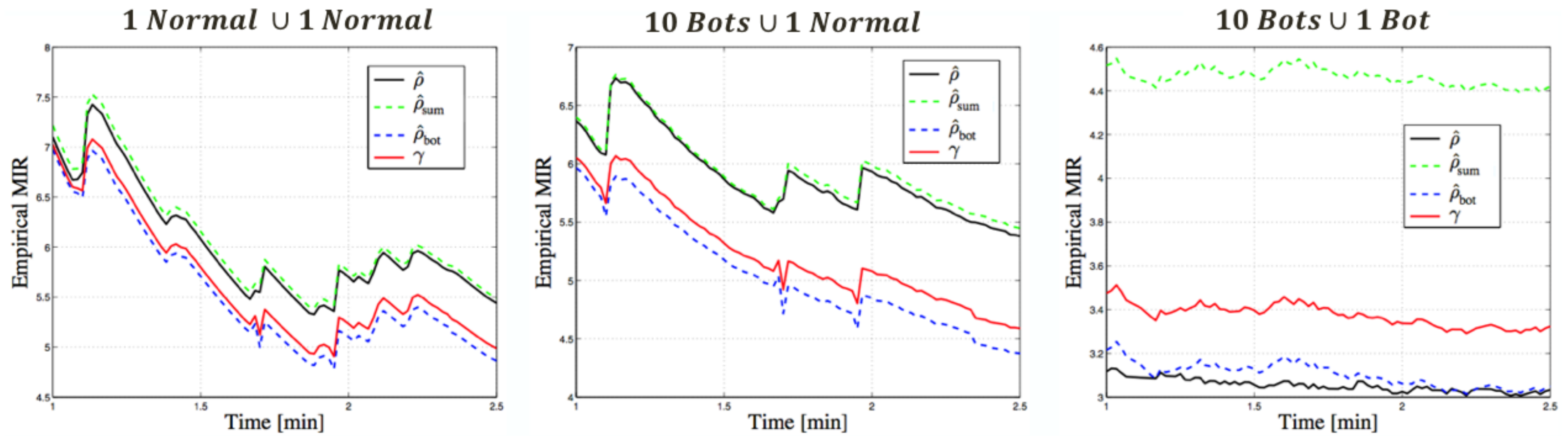
Key point: define a Message Innovation Rate (MIR) ρ defined as the number of **distinct** messages (picked from emulation dictionary) transmitted per unit time from bots.

Intuition: Botnet MIR is smaller than normal (and independent) users MIR

BIC: it is necessary to set a a threshold aimed at guaranteeing a separation between the MIR of a "trusted" Subnet and the MIR of a Botnet.

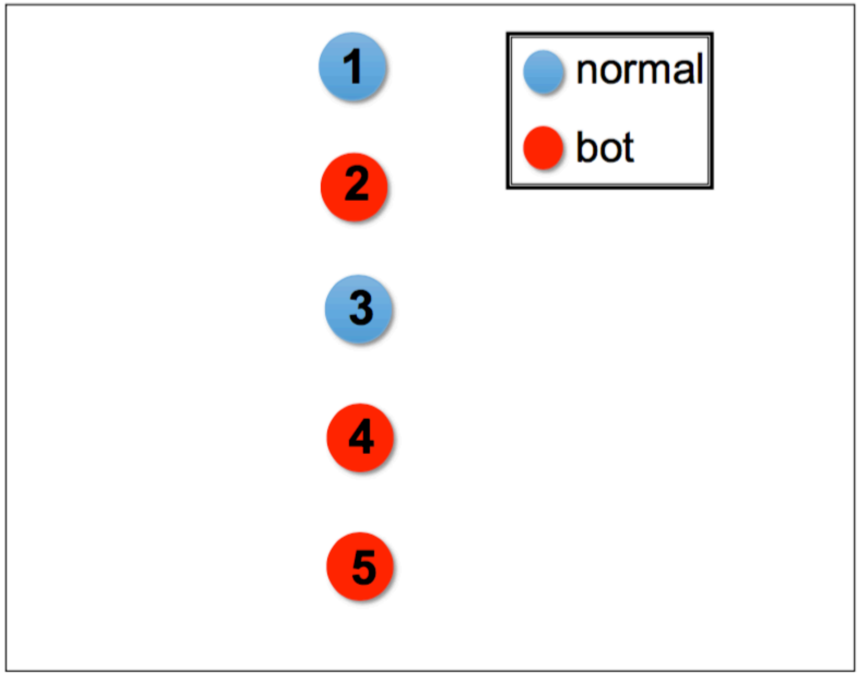
Set an intermediate threshold (tuning parameter $0 < \epsilon < 1$)

$$\rho_{\text{bot}} < \underbrace{\rho_{\text{bot}} + \epsilon(\rho_{\text{sum}} - \rho_{\text{bot}})}_{\text{Threshold } \gamma} < \rho_{\text{sum}}$$



Mario Di Mauro – University of Salerno

The BotBuster algorithm



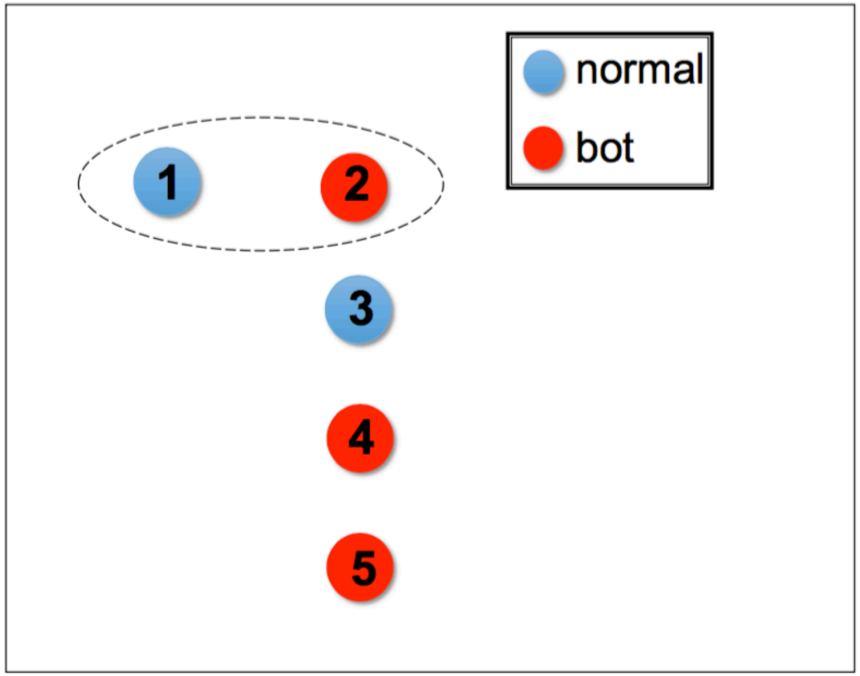
Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Set 1 as pivot

Mario Di Mauro – University of Salerno

The BotBuster algorithm

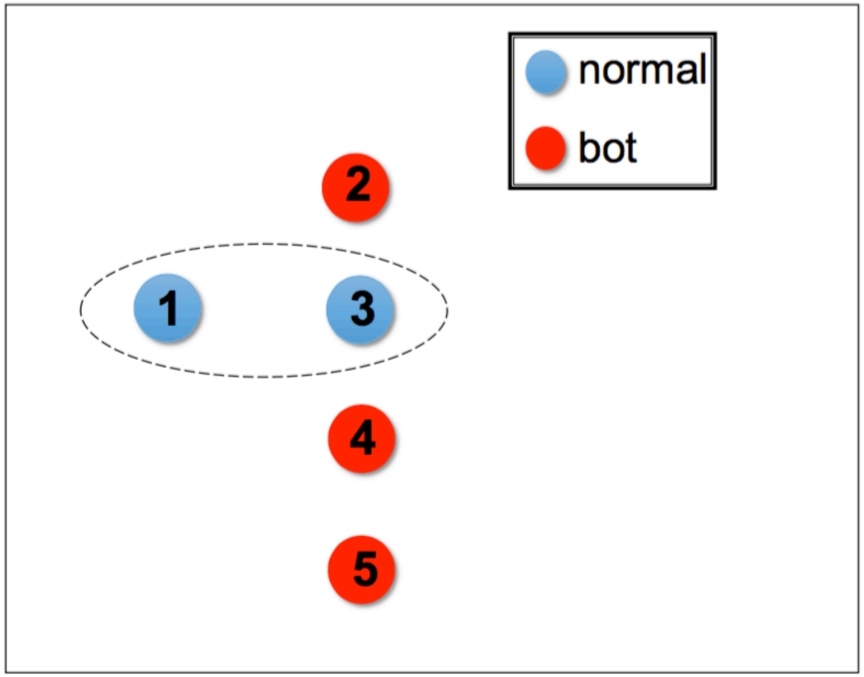


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Not botnet

The BotBuster algorithm

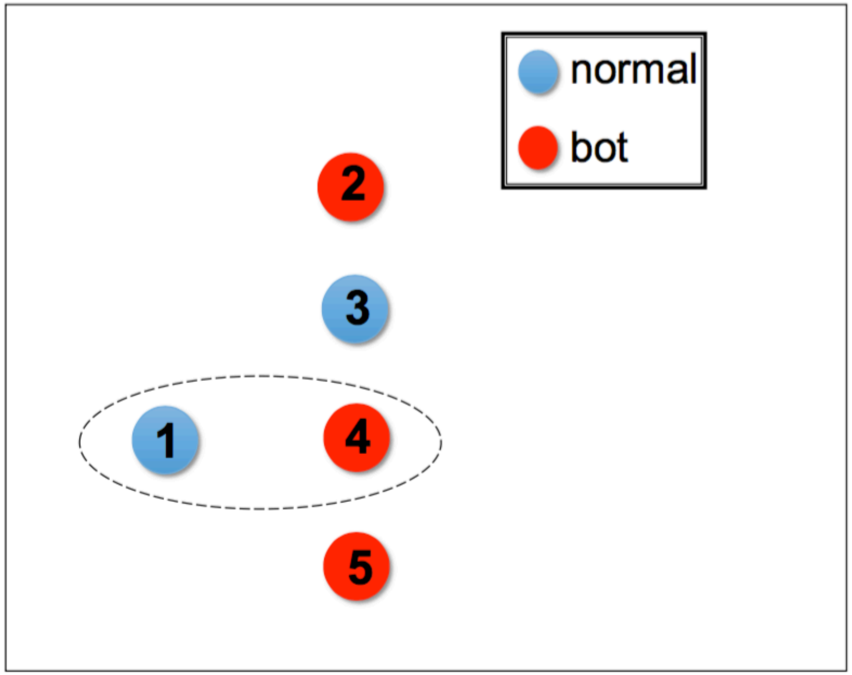


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Not botnet

The BotBuster algorithm

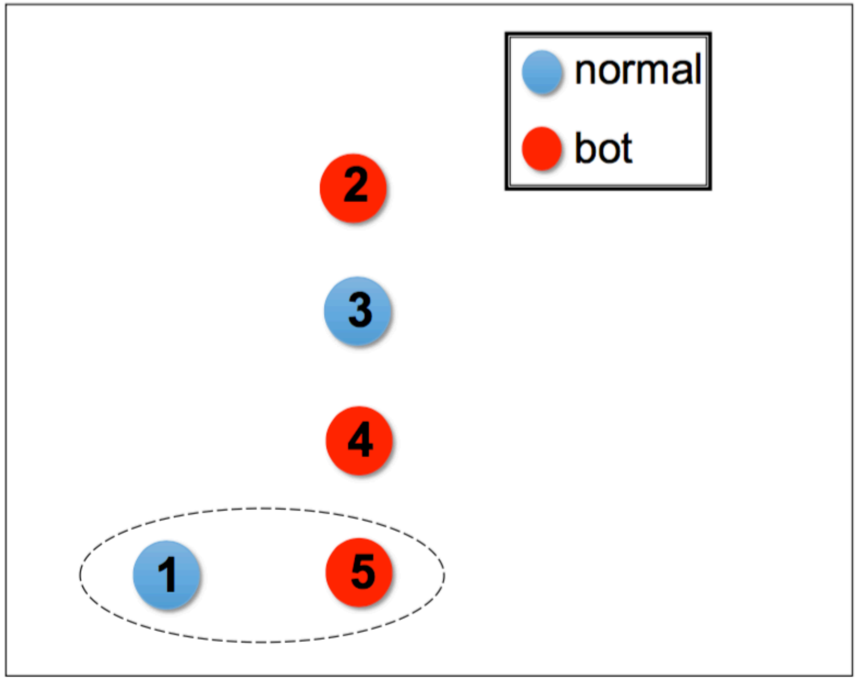


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Not botnet

The BotBuster algorithm

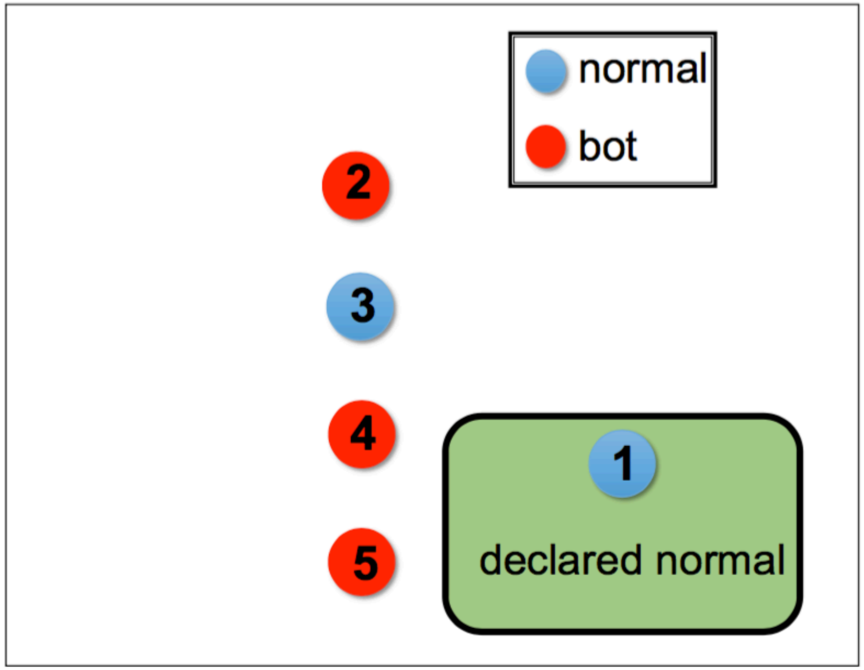


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Not botnet

The BotBuster algorithm



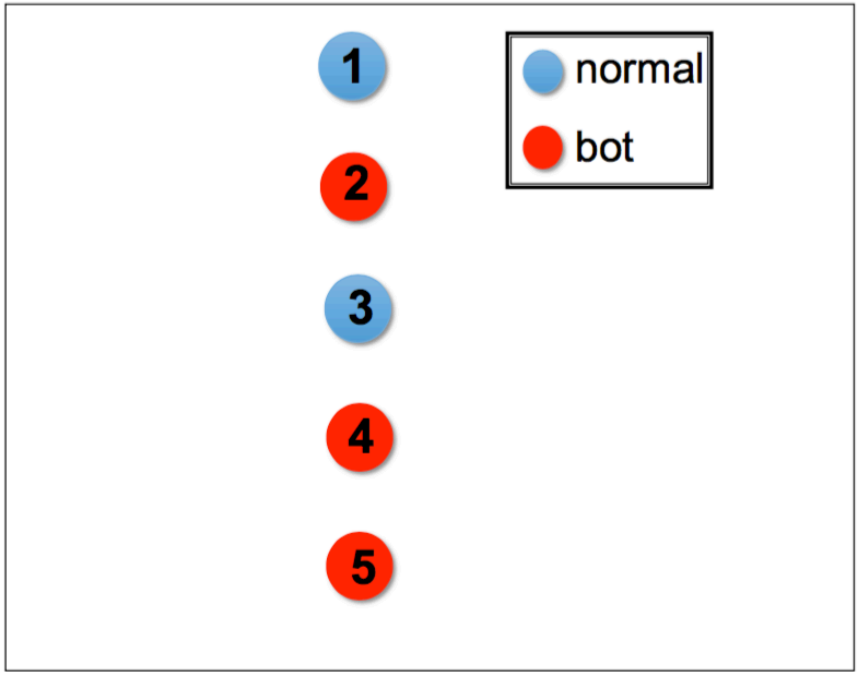
Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Estimate

Mario Di Mauro – University of Salerno

The BotBuster algorithm

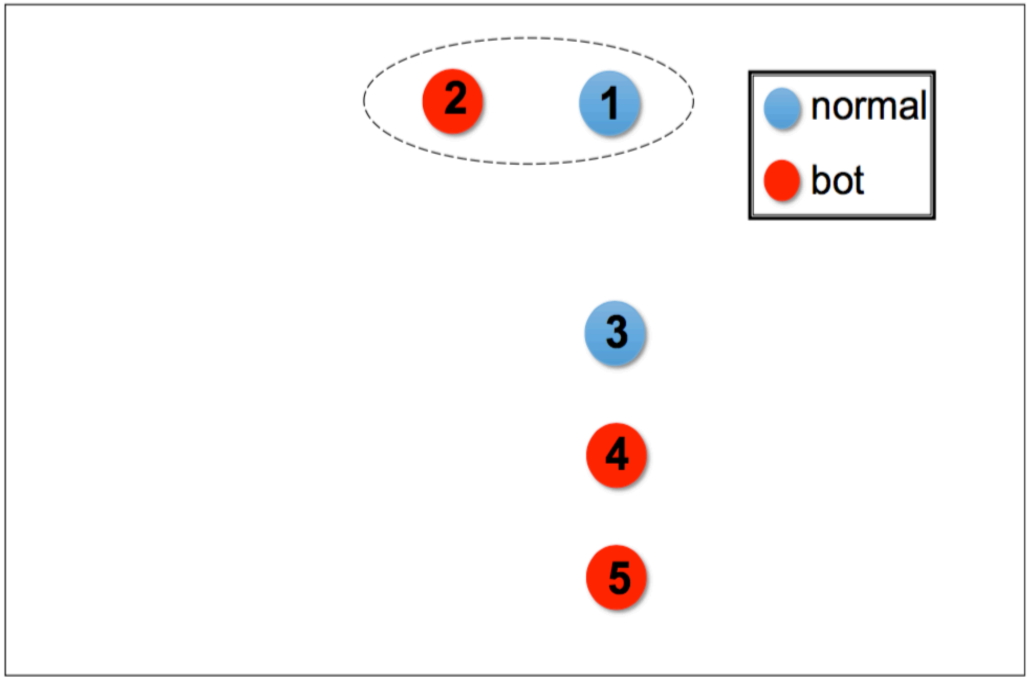


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Set 2 as pivot

The BotBuster algorithm

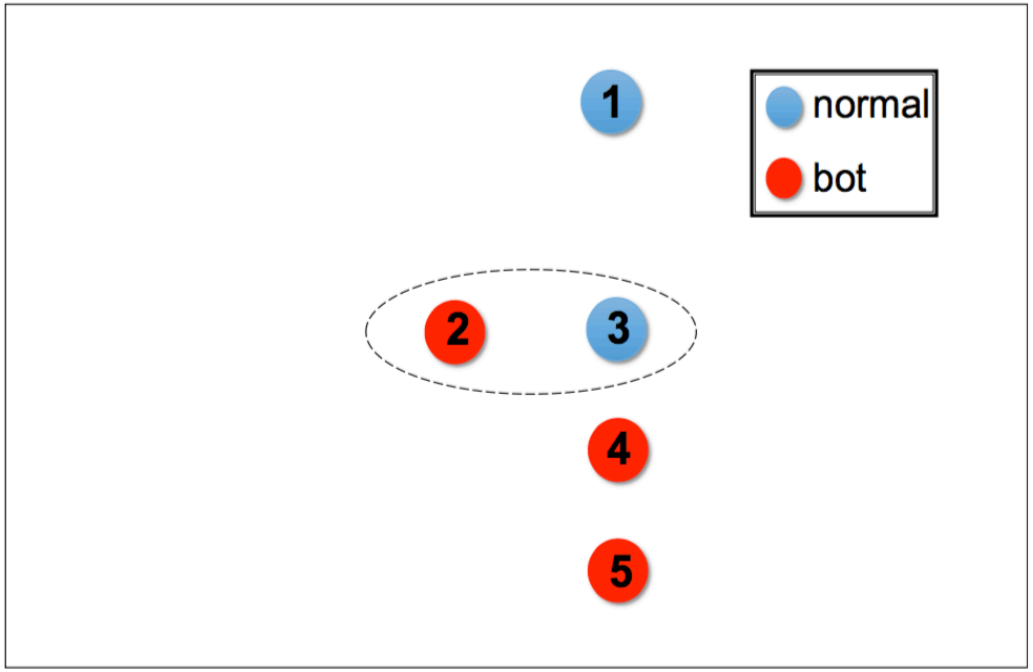


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Not botnet

The BotBuster algorithm

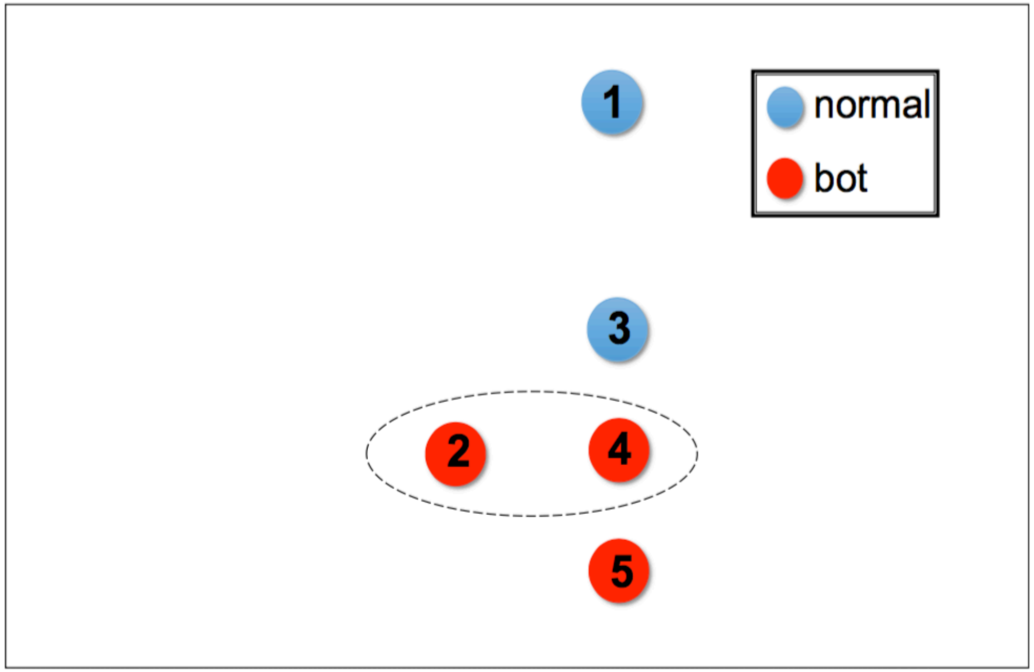


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Not botnet

The BotBuster algorithm

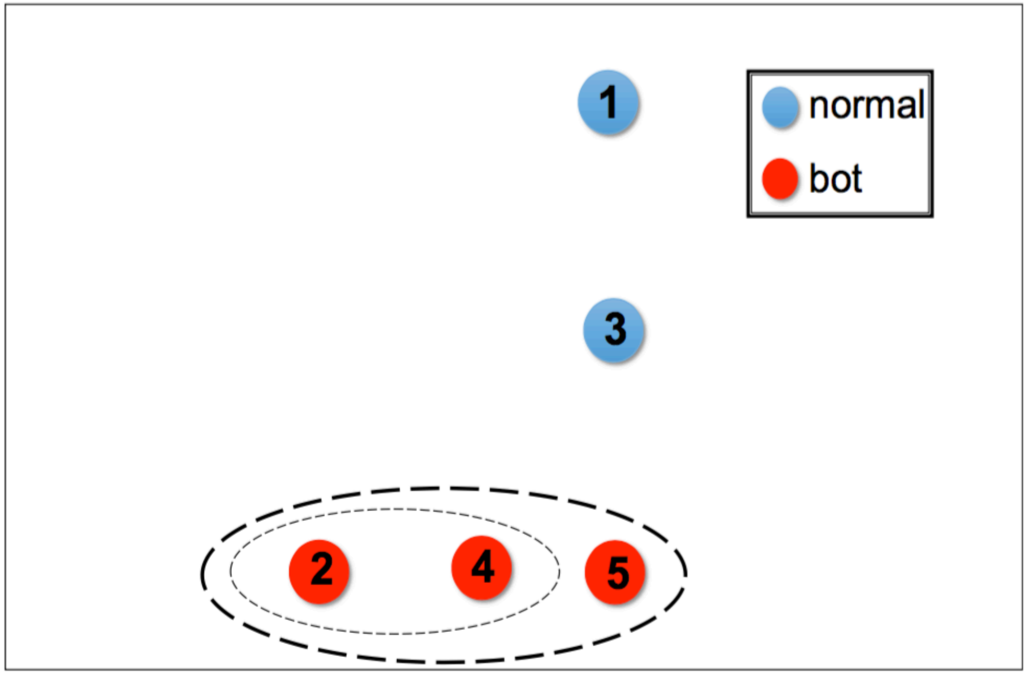


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

2 **and** 4 **bots**

The BotBuster algorithm

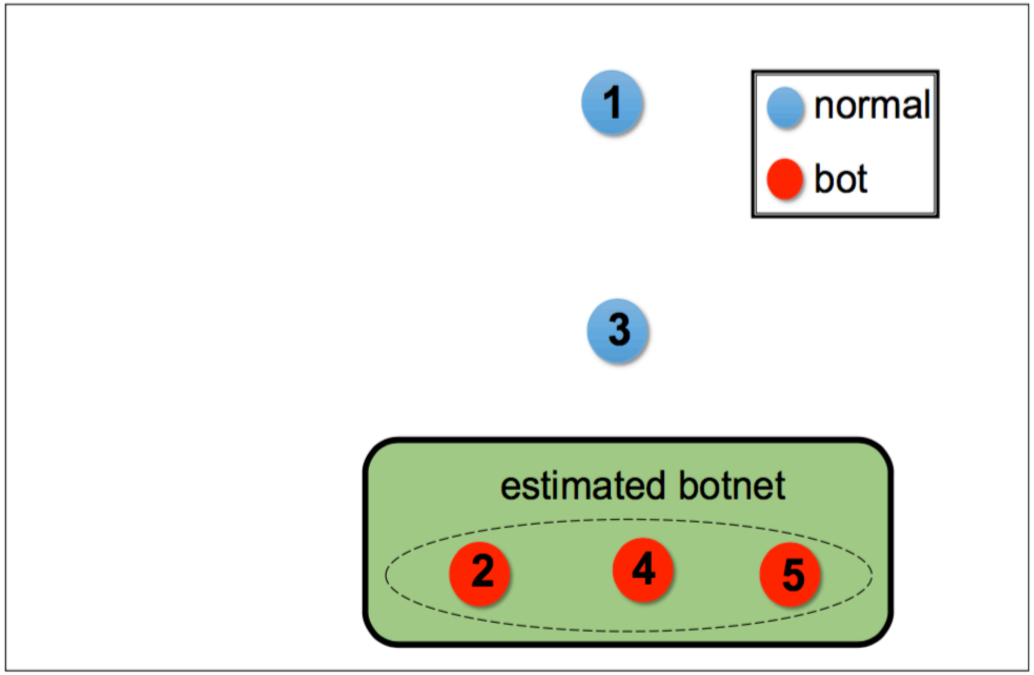


Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

$2, 4$ **and** 5 **bots**

The BotBuster algorithm



Algorithm 1: $\hat{\mathcal{B}}_{\text{new}}$=BotBuster

$\mathcal{N} = \{1, 2, \ldots, N\}$; $\hat{\mathcal{B}}_{\text{new}} = \emptyset$;
for $b_0 \in \mathcal{N}$ **do**
  $\hat{\mathcal{B}} = \{b_0\}$;
  for $j \in \mathcal{N} \setminus \{b_0\}$ **do**
    if $\hat{\rho}(\hat{\mathcal{B}} \cup \{j\}) < \gamma(\hat{\mathcal{B}}, \{j\})$ **then**
      $\hat{\mathcal{B}} = \hat{\mathcal{B}} \bigcup \{j\}$;
    end
  end
  if $|\hat{\mathcal{B}}| > \max(1, |\hat{\mathcal{B}}_{\text{new}}|)$ **then**
    $\hat{\mathcal{B}}_{\text{new}} = \hat{\mathcal{B}}$;
  end
end

Estimate

Performance indices

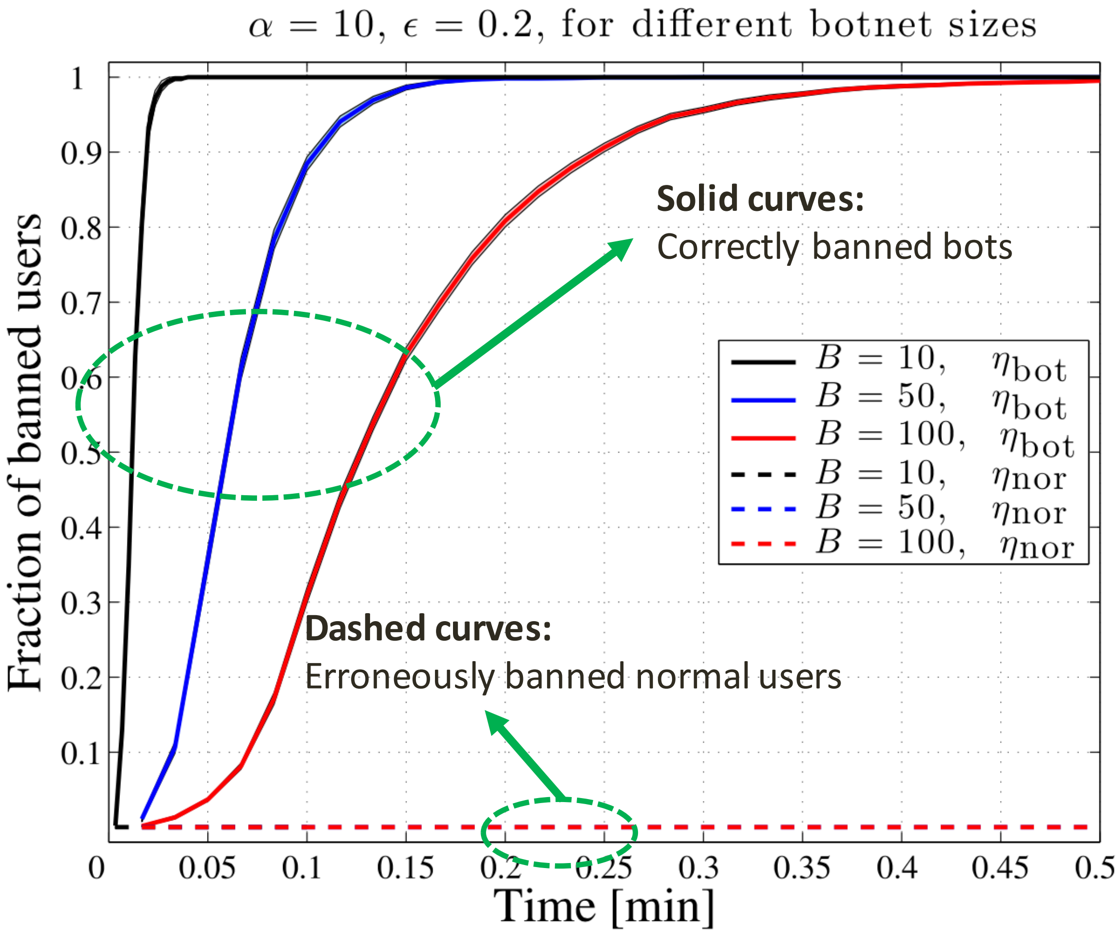
$$\eta_{bot}(t) = \frac{E[|\hat{B}(t) \cap B|]}{|B|}$$

Expected fraction of **correctly banned users**.
We want $\eta_{bot}(t) \to 1 \; as\ t\ goes\ to\ infinity$

$$\eta_{nor}(t) = \frac{E[|\hat{B}(t) \cap (N \backslash B)|]}{|N \backslash B|}$$

Expected fraction of **incorrectly banned users**.
We want $\eta_{nor}(t) \to 0 \; as\ t\ goes\ to\ infinity$

BotBuster applied to real data



- Fraction of banned users as a function of time, for different botnet sizes
- The monitored network is composed by 100 normal users
- Percentage of erroneously banned users nevers exceeds 5%
- The performance decreases as the number of bots grows

II. Analytical Model of Cyber-Threat Propagation

- Adoption of Birth-Death-Immigration model originally proposed by Kendall[1] in 1948
  - **Birth Rate** (λ): represents the number of hosts infected by another infected host per unit time (**internal** infection rate)
  - **Death Rate** (μ): represents the number of "cured hosts" per unit time
  - **Immigration Rate** (ν): represents the number of hosts directly infected by original source per unit time (**external** infection rate)

- The **Mitigation Strategy**: solution of an optimal resource allocation problem, by injecting the optimal curing vector $\boldsymbol{\mu}$

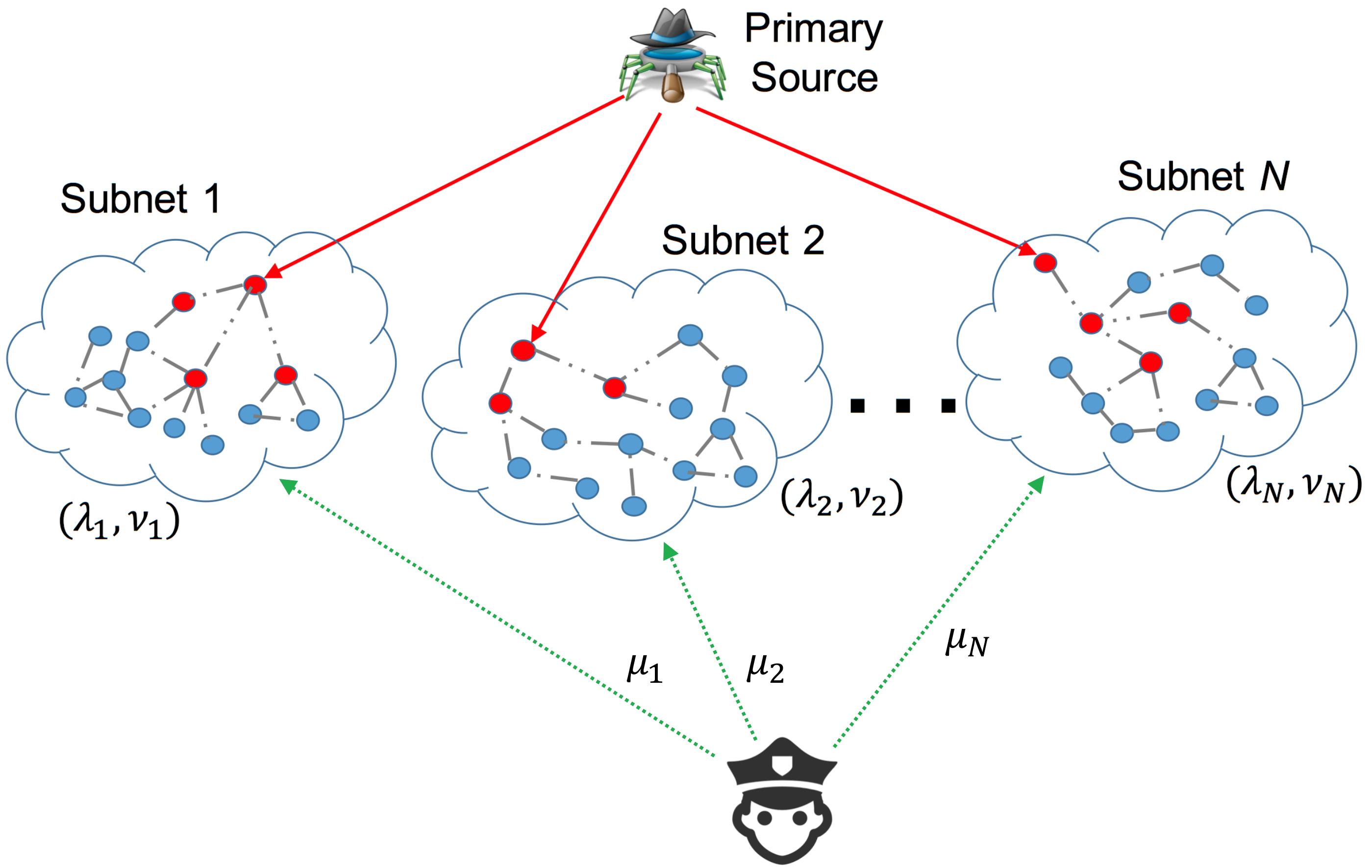
Two cases:

- Vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ perfectly known → exact solution
- Vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ unknown → Maximum Likelihood Estimation (MLE)

1. D.G. Kendall, "*On some modes of population growth leading to R.A. Fischer's logarithmic series distribution*," Biometrika, vol. 35, n°1/2, pp. 6-15, May 1948.

Mario Di Mauro – University of Salerno

II. Analytical Model of Cyber-Threat Propagation



1. N subnets (each subnet is susceptible to a specific threat)
2. The random process associated to the no. of sick nodes infected by the primary source is modeled by a Poisson counting process with rate ν
3. The random process associated to the no. of sick nodes infected by secondary source is modeled by a Poisson counting process with rate λ

Operational Regimes

Motivation: In the proposed threat propagation model, each infected node acts as a new (secondary) source of infection. The balance between infection and curing processes can originate various *operational regimes*

Definitions and adopted formalisms

$I(t)$ → Number of infected nodes (state) at time t

$p(n;t) \triangleq \mathbb{P}[I(t) = n]$ → Prob. distrib. of number of infected nodes

$\Psi(x;t) \triangleq \mathbb{E}[e^{xI(t)}]$ → Moment Generating Function (MGF) of *I(t)* at time t

$\Delta \triangleq \lambda - \mu, \quad \rho \triangleq \lambda/\mu, \quad \eta \triangleq \nu/\lambda$ → Normalized indicators

Mario Di Mauro – University of Salerno

Operational Regimes

Statistical characterization of *I(t)*

Key Idea: For the B-D-I model, it is possible to find a closed-form solution for the MGF and, then, for the corresponding probability distribution

$$\frac{\partial \Psi}{\partial t} + a(x)\,\frac{\partial \Psi}{\partial x} = b(x)\,\Psi$$

The MGF of *I(t)* obeys to this first order p.d.e.

$$a(x) \triangleq [\lambda(1-e^{x}) + \mu(1-e^{-x})], \quad b(x) \triangleq \nu(e^{x}-1)$$

$$\Psi(x;t) = \left(\frac{1-\pi_t}{1-\pi_t\, e^{x}}\right)^{\eta+n_0} \left(\frac{1-q_t\, e^{x}}{1-q_t}\right)^{n_0}$$

n_0 is the initial number of infected nodes

$$\pi_t \triangleq \frac{e^{\Delta t}-1}{e^{\Delta t}-1/\rho}, \quad q_t \triangleq \frac{e^{\Delta t}-\rho}{e^{\Delta t}-1}$$

Mario Di Mauro – University of Salerno

Asymptotic Regimes

Statistical characterization of *I(t)*

A seq. $X_1, X_2, \ldots, X_n$ of real-valued r.v. is said to *converge in distribution* to r.v. X if:

$$\lim_{n \to \infty} F_n(X) = F(X)$$

(for all $x \in \mathbb{R}$ at which F is continuous)

Key Idea: the convergence of MGF implies the convergence in distribution

$$I(t) \xrightarrow[t \to \infty]{\text{d}} \mathcal{N}_b(\eta, \rho), \qquad \text{if } \rho < 1,$$

$$\frac{I(t)}{\lambda t} \xrightarrow[t \to \infty]{\text{d}} \mathcal{G}(\eta), \qquad \text{if } \rho = 1,$$

$$I(t)\, e^{-\Delta t} \xrightarrow[t \to \infty]{\text{d}} \mathcal{Y}(\eta, \rho, n_0), \qquad \text{if } \rho > 1$$

- $\mathcal{N}_b(\eta, \rho)$: Negative binomial Random Variable (**stable case**)
- $\mathcal{G}(\eta)$: Unit-scale Gamma Random Variable (**unstable case**)
- $\mathcal{Y}(\eta, \rho, n_0)$: Generic Random Variable (**strongly unstable case**)

Optimal Resource Allocation

Key Idea: Given "infection parameter vectors" $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$, we are interested in allocating the optimal "curing vector" $\boldsymbol{\mu}$. Ideally, we would to solve the following *Optimization Problem*:

$$\min_{\boldsymbol{\mu}} \sum_{\ell=1}^{N} I_\ell(t) \quad \text{s.t.} \quad \sum_{\ell=1}^{N} \mu_\ell \leq C$$

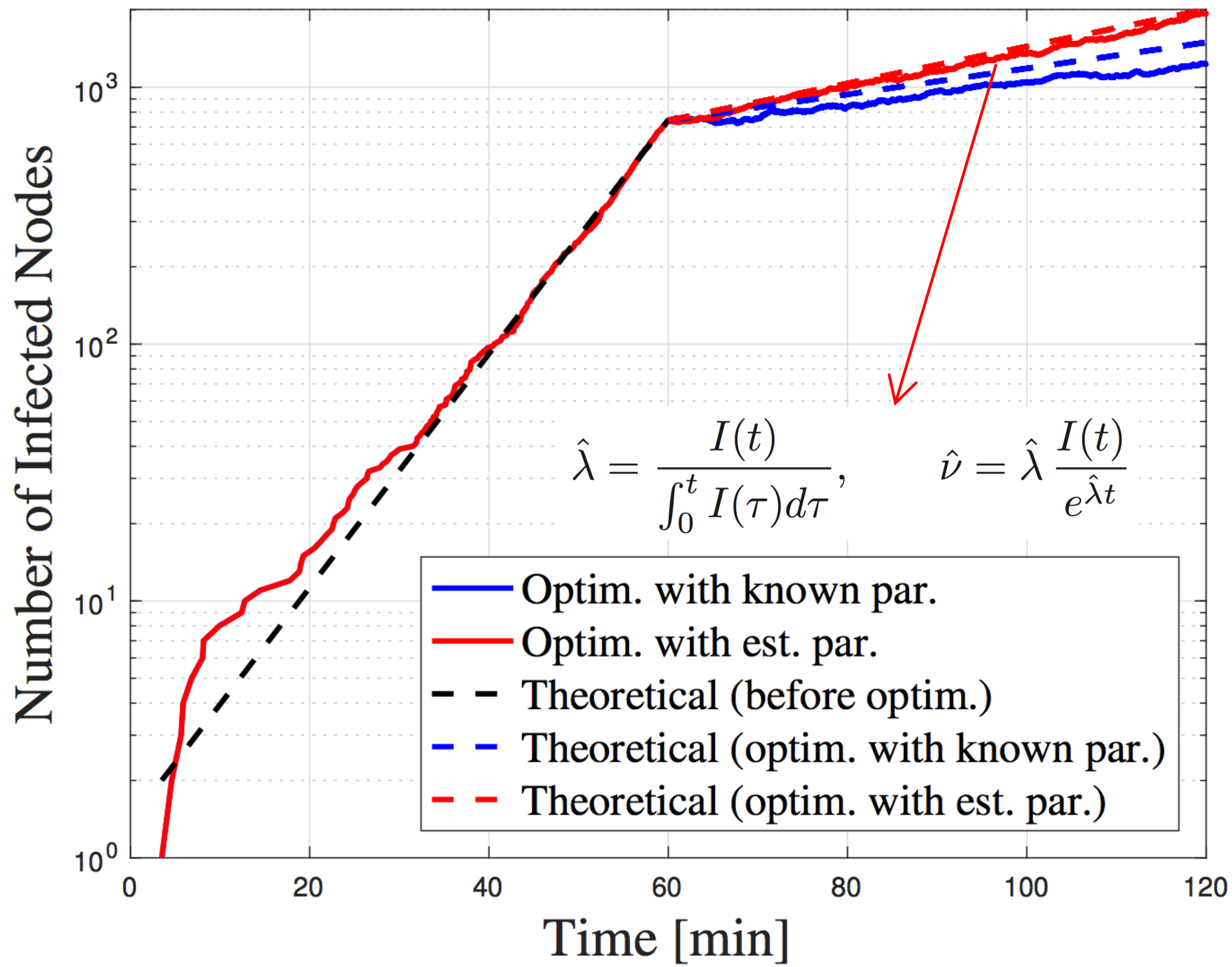
C represents the available curing capacity that determines two regimes:

$$\sum_{l=1}^{N} \lambda_l > C \longrightarrow \text{global infection rate greater than the available capacity}$$

$$\sum_{l=1}^{N} \lambda_l \leq C \longrightarrow \text{global infection rate smaller (or equal) than the available capacity}$$

Optimal Resource Allocation

Numerical Results



N° of infected nodes spreading across N=3 subnets.

$$\boldsymbol{\lambda} = [0.104, 0.052, 0.017]$$

$$\boldsymbol{\nu} = [0.104, 0.157, 0.069]$$

$$C = 0.8 \sum_{\ell=1}^{N} \lambda_\ell$$

Case 1:

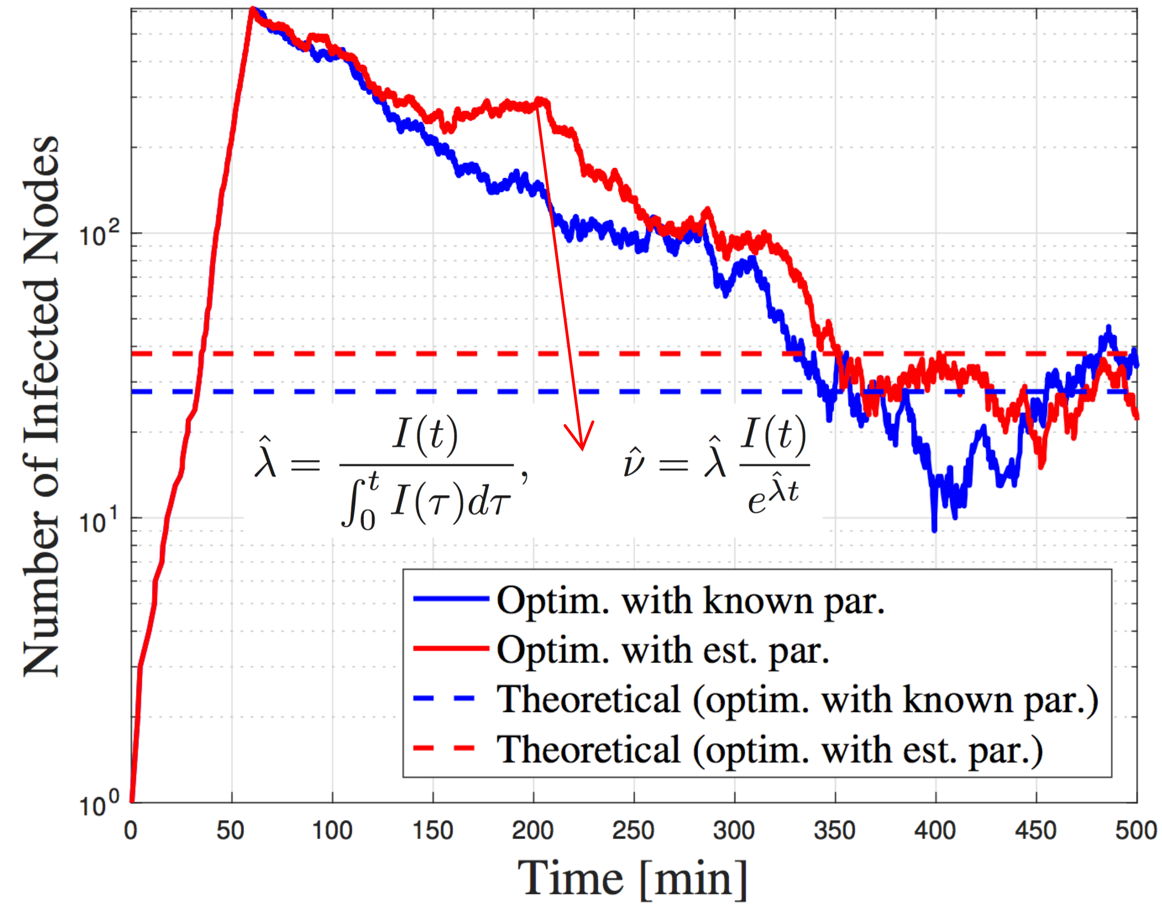
$$\sum_{\ell=1}^{N} \lambda_\ell > C$$

The optimization procedure focuses on mitigating the threat (exp) growth rate

C.C. Zou, W. Gong, D. Towsley, *Code Red Worm Propagation modeling and analysis*, IEEE/ACM Transactions on Networking, Vol. 13, n°5, Oct.05, pp. 961-974

Optimal Resource Allocation

Numerical Results



N° of infected nodes spreading across N=3 subnets.

$$\boldsymbol{\lambda} = [0.104, 0.052, 0.017]$$

$$\boldsymbol{\nu} = [0.104, 0.157, 0.069]$$

$$C = 1.1 \sum_{\ell=1}^{N} \lambda_\ell$$

Case 2:

$$\sum_{\ell=1}^{N} \lambda_\ell < C$$

⇩

The optimization procedure is able to guarantee the stability of threat growth (exp. divergence prevention)

Conclusions

1. Conceptualization of a randomized distributed network attack along with mitigation strategies.

   Ongoing work: cluster of botnets that completely/partially share emulation dictionaries

2. Characterization of threat propagation phenomenon by means of Kendalls' B-D-I- model with optimal curing solution tested over simulated data

   Ongoing work: formulation of the adversarial problem through Game Theory framework

Other (related) Publications

- Di Mauro M., Galatro G., Longo M., Postiglione F., Tambasco M. Availability Modeling of a Virtualized IP Multimedia Subsystem using non-Markovian Stochastic Reward Nets. Accepted for European Safety and Reliability Conf. (2018).

- Di Mauro M., Di Sarno C. (2018) Improving SIEM capabilities through an enhanced probe for encrypted Skype traffic detection. In: Journal of Information Security and Application (Elsevier), Vol. 38, n°PP, Pagg. 85-95. ISSN: 2214-2126.

- Di Mauro M., Longo M., Postiglione F., Tambasco M. (2017). Availability Modeling and Evaluation of a Network Service Deployed via NFV. Digital Communication: Towards a smart and secure future internet. (Springer), chapter book, pag. 31-44. ISBN: 978-3-319-67638-8.

- Di Mauro M., Longo M., Postiglione F., Tambasco M., Carullo G. (2017). Service Function Chaining deployed in an NFV environment: an availability modelling. In proc. of CSCN17, Helsinki, pag. 42-47. ISBN: 978-1-5386-3070-9.
- Matta V., Di Mauro M., Longo M., (2017). Botnet Identification in Multi-clustered DDoS Attacks. In proc. Of Eusipco1, Kos Island, August 2017. ISBN: 978-0-9928626.

- Di Mauro M., Longo M., Postiglione F., Carullo G., Tambasco M. (2017). Software Defined Storage: availability modeling and sensitivity analysis. In IEEE/SCS International Symposium on Performance Evaluation of Computer (SPECTS17), Seattle 9-12 Jul 2017, Pag. 445-451. ISBN:1-56555-362-4 .

- Carullo G., Di Mauro M., Galderisi M., Longo M., Postiglione F., Tambasco M. (2017). Object Storage in Cloud Computing environments: an availability analysis. LNCS 10232 – Green, Pervasive, and cloud computing 2017, Cetara 11-14 May 2017, Pag.178-190. ISBN:978-3-319-57185-0.

- Di Mauro M., Galatro G., Longo M., Postiglione F., Tambasco M. (2017). Availability evaluation of a virtualized IP multimedia subsystem for 5G network architectures. In: IN Esrel17. Portorose 18-22 Jun. 2017, Pag.2203-2210 LONDON, TAYLOR and FRANCIS GROUP. ISBN:978-1-138-62937-0.

- Di Mauro M., Longo M., Postiglione F., (2016). Performability evaluation of Software Defined Networking Infrastructure. In Valuetool16. Taormina, Oct 2016, Pag. 1-8. ISBN: 978-1-63190-141-6.

Other (related) Publications

- Di Mauro M., Longo M., Postiglione F., Restaino R., Tambasco M. (2016). Availability Evaluation of the Virtualized Infrastructure Manager in Network Function Virtualization Environments. In Esrel16. Glasgow, Sept. 2016. ISBN: 978-1-138-02997-2.
- Matta V., Di Mauro M., Longo M., (2016). Botnet Identification in Randomized DDoS Attacks. In EUSIPCO16, Budapest, Aug./Sept. 2016, ISBN: 978-0-9928-6265-7.
- Cirillo M., Di Mauro M., Longo M., Senatore A., (2016). Detection of encrypted multimedia traffic through extraction and parameterization of Recurrence Plots. In Multimedia, Network Security and Communications 2016 (MNSC16), Bangkok, Mar. 2016, ISBN: 978-1-60595-337-3.
- Carullo G., Di Mauro M., Longo M., Tambasco M., (2016). A Performance Evaluation of WebRTC over LTE. In IEEE/IFIP Wireless On-demand Network systems and Services Conference (WONS 2016). Jan, 2016 ISBN: 978-3-901882-80-7 pp. 170-175. 978-3-9018-8279-1.
- Di Mauro M., Longo M., Postiglione F. (2015). Reliability Analysis of the Controller Architecture in Software Defined Networks. In: IN Esrel15. Zurich 7-10 Sept. 2015, Pag.1503-1510 LONDON, TAYLOR and FRANCIS GROUP. ISBN:978-1-138-02879-1.
- Di Mauro M., Longo M. (2015). Revealing Encrypted WebRTC traffic via Machine Learning tools. In: IN Secrypt15. Colmar (Alsace), 20-22 Jul. 2015, pp. 1-5. 978-9-8975-8140-3.
- Di Mauro M., Longo M. (2015). A Decision Theory Based Tool for Detection of Encrypted WebRTC Traffic. In: ICIN15. Parigi, 16-19 Feb. 2015, pp. 89-94, ISBN/ISSN: 978-1-4799-1865-2.
- Di Mauro M., Longo M. (2014). Skype traffic detection: a decision theory based tool". In: 48th International Carnahan Conference on Security Technology. Rome, 13-16 Oct. 2014, pp. 52-57, ISBN: 978-1-4799-3531-4.

Mario Di Mauro - Ingegneria dell'Informazione - XXX ciclo