Fighting Botnets - A Systematic Approach

Nuno G. Rodrigues, António Nogueira and Paulo Salvador
Instituto de Telecomunicações/University of Aveiro
Campus de Santiago, 3810-193 Aveiro, Portugal
E-mail: nuno@ipb.pt, {nogueira, salvador}@ua.pt

***Abstract*—The increasing impact of Internet in the global economy has transformed botnets into one of the most feared security threats for citizens, organizations and governments. Despite the significant efforts that have been made over the last years to understand this phenomenon and develop detection techniques and countermeasures, this continues to be a field with big challenges to address. The most important detection approaches and countermeasures that have been proposed are usually oriented to address some specific type of botnet threat or fight botnets in particular scenarios or conditions. This paper proposes a generic and systematic model to describe the network dynamics whenever a botnet threat is detected, defining all actors, dimensions, states and actions that need to be taken into account at each moment. We believe that the proposed model can be the basis for developing systematic and integrated frameworks, strategies and tools to predict and fight botnet threats in an efficient way.**

Keywords-*network security; malware; botnet; network resilience.*

I. INTRODUCTION

In the last decades, communication networks, and Internet in particular, incredibly expanded their usage, importance and impact levels in the global economy. Nowadays, significant parts of our daily lives are directly or indirectly related with the Internet, with the use of services like the e-mail, online news or entertainment, teleworking, business transactions, home banking, social networks and much more. This level of dependence raised this network to the level of a global critical infrastructure, where possible failures and disruptions have a tremendous impact in the global economy. If the Internet relevance in current society increases very fast, motivations for launching cyber-attacks and the Internet vulnerability level increase even faster. In many aspects, the new level of importance was not accompanied by the increase of reliability, availability and security [1] or, in other terms, of the network resilience [2].

From the three disciplines that mainly characterize network resilience, security is the most challenging. In fact, the range of security threats that can affect Internet is immense and increasingly complex, reinforced with the beginning of a new era where cyber-war between nations is a reality. One recent example of this situation were the massive Distributed Denial of Service (DDoS) attacks deployed against Georgia governmental Web sites during the summer of 2008, coinciding with the movement of Russian troops into the Georgian province of South Ossetia [3] or the recently discovered Stuxnet botnet [4] that was specifically developed to sabotage the Iranian uranium enrichment infrastructure.

Network security is a very broad topic that includes issues like confidentiality, authenticity, integrity, authorization or non-repudiability. The lack of security of computers and networks is created, in a first instance, by the existence of vulnerabilities that can become a threat. Threats can become attacks, which can result in compromised systems. One of the most common security threats in current networks and computer systems is the use of software with malicious functionalities, known as *malware* [6]. Malware is a generic term that encompasses specific malicious pieces of software like rootkit, virus, worm, spyware, trojan horse, sniffer and many others. A large set of infected computers (bots) that is remotely and coordinated controlled by an attacker (botmaster) is known as a botnet. Although botnets are used for many different malicious purposes, nowadays the most relevant uses are for political and financial benefits [6].

During the last years, several techniques were developed to detect botnets in local networks. These techniques are usually divided among passive and active [6]: passive techniques are only based on monitoring and observation, acting transparently without interfering with the botnet environment, while active techniques use approaches that interact with the environment under observation and monitoring. Whenever a botnet is detected, it is necessary to deploy appropriate countermeasures that should limit the threat and/or eliminate it. Countermeasures can be grouped into three main categories: technical, regulatory and social methods [7].

Although the identification of possible countermeasures that can fight and remove botnet threats in a local network is nowadays reasonably well achieved, their systematic application needs to be significantly improved. Cleaning infected machines using anti-virus software, applying traffic filtering rules or blocking network elements' ports are relatively common measures taken by network administrators in the case of a botnet detection. However, since these threats become more and more complex and sophisticated, the fighting procedures need to be systematized and automated. Besides, having the ability to model all network states (from a security perspective) can help predict future network states/behaviors based on available (input) events. This systematization will facilitate the deployment of automated countermeasures for any detected threat. This paper proposes a generic network model that is able to describe the different network dynamics under the presence of a

 ISBN: 978-1-61208-239-4

botnet threat: all actors, dimensions, states and actions that need to be taken into account at each moment will be defined, allowing the development of appropriate inference procedures that can infer the values of different model parameters based on real data.

The paper is organized as follows. Section II presents the most relevant background on botnet infrastructures, detection approaches and countermeasures; Section III presents the network modeling approach, including all possible network states and all the actions that originate state transitions, besides discussing the necessary steps to infer the model parameters and use it to help network managers and administrators; finally, Section IV presents the main conclusions and topics for future work.

II. Background on Botnets

A botnet is a large collection of computing systems that is infected with the same piece of malware (bot) and is remotely controlled by one or more attackers (botmasters), using a specific command-and-control (C&C) infrastructure [1], with the purpose of performing malicious actions like sending spam email, triggering distributed denial-of-service attacks (DDoS), capturing private information or propagating other types of malware. Infected computers and networks become unstable and, frequently, unable to operate normally.

Nowadays, it is estimated that millions of infected systems exist in the Internet, being part of thousands of botnets [8]. According to Fossi *et al.* [9], the Rustock botnet, for example, controlled more than 1 million bots. If in the last years economic benefit has been the major motivation for botnet deployment, recently we are witnessing its use for political purposes [3], [4] and for several underground cybercrime activities [6]: unsolicited mass mailing (spam), click frauds and pay per install, identity theft, DDoS.

A. Botnet infrastructures

The botnet C&C infrastructure includes bots and a control entity, using an addressing mechanism and one or more protocols to maintain a communication channel and distribute commands between the infected computers and the botmasters [10]. The C&C infrastructure can have a centralized, decentralized or locomotive based architecture.

In a centralized architecture, bots act only as clients, connecting and receiving commands from one or more servers. This architecture is based on a client-server communication model, where HTTP and IRC are the most common communication protocols [11]. Centralized infrastructures can be based on single central C&C servers or in a multi-layered structure of servers and bots. In this second alternative, servers can be divided into different roles: some can be used for command and control and others for delivering contents to bots. Bots can also perform different roles in the botnet structure.

In decentralized architectures, also known as peer-to-peer architectures, there is no differentiation between clients and servers. All nodes participating in the botnet perform the same set of roles, being known as peers. The communication protocol is also based in peer-to-peer models. With this architecture, botmasters control bots by inserting commands and updates in an arbitrary point of the botnet, which makes their localization almost impossible and provides a very high degree of anonymity. There are no central servers to mitigate and disable. However, the propagation of commands through the botnet is slower when compared to centralized approaches. There are some botnets that use hybrid infrastructures, with a centralized infrastructure as the primary option and an alternative peer-to-peer backup channel.

Locomotive botnets use a central C&C infrastructure that is constantly moving over time. This means that the C&C servers are continuously changing, with the support of the DNS service [6].

A highly complex DNS-based technique was used by botnet developers to increase botnet resilience and anonymity: the so called fast-flux service [6]. With this service, it is possible to use several bots as proxy servers to transparently forward malicious communications from clients to a malicious server. The proxy servers hide to the outside the malicious services that are available in the malicious server. The main characteristic of this mechanism is the use of round-robin DNS with very short TTL values associated with the DNS resource records in order to rapidly and continuously change the IP addresses of the bot proxies, being extremely difficult to follow and intercept these communications.

B. Botnet detection and countermeasures

Since botnets act with discretion, their detection is very challenging. One of the solutions that have been used for botnet detection and tracking is based on honeynets [12], a set of honeypots. A honeypot is an intentionally insecure computational system that is placed in the network with the objective of detecting and capturing traffic from botnets in order to understand their characteristics and *modus operandi*. The most important botnet detection techniques that have been proposed are based on passive monitoring and analysis of the network traffic, and can be classified into four main categories [13], [14]:

- Signature-based: these techniques are based on previous knowledge about malware and botnets [15]. One known example is the Snort [16] tool, an open source intrusion detection system (IDS). The main drawback of this type of systems is that they can only detect known botnets and malware.
- Anomaly-based: these techniques are based on the detection of traffic anomalies, like high volumes of traffic, high delay or jitter, unusual ports or unusual system behaviour [8]. However, if the botnet traffic seems to have normal patterns, this type of methods cannot detect it. Botsniffer [17] is an anomaly-based detection tool.
- DNS-based: these techniques apply the same principles of the anomaly-based techniques to the specific case of DNS traffic.

 ISBN: 978-1-61208-239-4

- Mining-based: Since the other techniques are not effective to detect C&C traffic, this approach uses data mining techniques to perform this identification. Masud *et al.* [18] presented a very promising data mining identification methodology.

When a botnet is detected, it is necessary to do all the possible to mitigate the threat, taking measures to shut it down if possible. Because of the dissimulated nature of these systems, this is a challenging task. The most common approach is based on searching for central weak points in the botnet infrastructure that can be disrupted or blocked. In general, two main approaches exist: classical countermeasures and offensive strategies [10]. In the classical countermeasures group, the three most common used techniques are:

- Taking down the C&C server. Whenever possible, this is the most effective and fast way to shut down the botnet. However, it is only applicable to botnets with a central infrastructure and if the location of the C&C server is known. The cooperation of the service provider where the server is connected to is fundamental in this step. Besides, depending on the botnets, bots can be prepared to spread and perform tasks autonomously, without communicating with the C&C server.
- Sinkholing malicious traffic. If shutting down the C&C server is not possible, the traffic between bots and this server can be redirected to a sinkhole. This can be done at the routing level, either in a local or global scale, obviously depending on the cooperation between organizations and ISP's.
- Cleaning infected systems. Although, this is the most sustainable measure to eliminate a botnet threat, it is also the most difficult due to the extremely large spectrum of client systems that normally are infected, covering many different geographical areas, different types of users, etc. The most common approaches are based on the use of up-to-date anti-virus and personal firewalls in the end user systems. However, usually these tasks are not controlled by the network and system administrators, which makes them so difficult to implement.

The effective implementation of classical countermeasures clearly depends on the organizational and political cooperation between different entities, which is usually a slow process when compared to the urgency that is required to fight these threats. Additionally, the most recent botnet threats use increasingly sophisticated obfuscation techniques that make the application of classical countermeasures even more difficult. To solve these limitations, some new proactive offensive approaches have been proposed [10]:

- Mitigation: an offensive approach against the botnet infrastructure, similar to temporary DoS attacks to C&C servers, trapping and blocking connections from infected machines or malicious domains.
- Manipulation: this approach relies on bugs found in bots to access the C&C channel, intercepting commands and forge new fake commands to change their behavior. In the limit, fake commands can order the bots self-destruction.
- Exploitation: this approach explores bugs in the C&C servers or even in the bots to gain control over them and promote their destruction from inside.

Despite being technically feasible and very effective, these types of techniques raise several ethical and legal questions, as the name (offensive) suggests. If fact, the use of these techniques usually implies the unauthorized access to infected machines and infrastructures, which means using the same (and many times illegal) rules as the attackers. An example of this complexity is the recent action of FBI to take the control of the C&C servers of the DNSChanger trojan.

Chainey [19] proposed a new approach for collective cyber threat defense efforts based on the public health models that are used in several countries. In this proposal, the authors defend the use of health certificates for all systems connected to the Internet. These certificates demonstrate the health condition of each device and can be used by service providers to allow or block access to specific resources (like home banking platforms, for example). Despite being an interesting theoretical approach, many practical questions need to be addressed to implement this model, ranging from the specification of certificates and protocols to the construction of a global infrastructure that can manage the system.

Another different and innovative approach is described in [20], where where Li and Liao proposed the idea of using virtual bots to create uncertainty in the attack capacity of each botnet. This study advocates that this uncertainty has a significant impact on the profits of botmasters and attackers, which means that the economic benefits can be destroyed or mitigated and the corresponding interest in using the botnet will automatically decrease.

III. BOTNET FIGHTING - MODELING THE RESPONSE

Formal models that support systematic and methodical approaches are an important tool to improve computer and network security in general [21] and, we believe, to efficiently fight botnets. This section will present a network model that can describe the different network states, according to the degree of botnet infection that is detected, and the actions that lead to state transitions. The finite state machine model that is proposed includes a detailed characterization of the possible states of all network elements (hosts, switches, routers), allowing a rigorous and precise knowledge of the network operation details at any given time instant. The nature of the proposed model allows its use in the prediction of the network states at future time instants.

As a base discipline that affects network resilience, security issues can be addressed using the two-phase ResiliNets strategy D^2R^2+DR described in [2]. The first phase of this strategy (D^2R^2) runs in real-time and corresponds to the **D**efend, **D**etect, **R**emediate and **R**ecover steps, while the second phase (DR) runs in background and includes

 ISBN: 978-1-61208-239-4

the **D**iagnose and **R**efine steps. Considering this strategy, our current work is based on the following assumptions:

1) Network and host defenses can be broken and hosts can be infected by malware, becoming members of botnets;
2) Actual techniques and resources can detect the infection of hosts and the presence of botnet activities in a local network.

This means that the work will be focused in modeling the *Remediate* and *Recover* steps of the ResiliNets strategy, in the presence of botnet threats.

A. The network model

In a first step, the problem will be limited to the perspective of a local network, where it is necessary to model the response behavior of the following actors: switches (from core, distribution and access layers), routers and hosts. A local network that is facing a possible botnet infection can be described by a sub-set of the following states and transitions:

- *Normal state*: in this state, the network is working according to its baseline, without strange events originated by the presence of malware running on hosts. The transition to another state is affected by the following transitions:
  - *Botnet Infection*: if a botnet infection is detected, the network changes from the Normal to the Impaired state;
  - *Massive Botnet Infection*: if an unexpected massive botnet infection is detected, the network changes directly from the Normal to the Generalized Infection state;
- *Impaired state*: some infections on local hosts were detected but their impact in the overall network performance and security is not very significant. The transitions that affect this state are:
  - *Increased Botnet Infection*: if the previously detected botnet infection increases significantly, the network needs to change from the Impaired state to the Generalized Infection state;
  - *Recovery measures*: the deployment of adequate recovery measures was able to eliminate the security threat, allowing the network to recover to the Normal state.
- *Generalized Infection state*: a significant infection was detected on local hosts, with a big impact on the overall performance of the local network. The transitions from this state are affected by the following actions:
  - *Remediation measures*: the deployment of remediation measures that confine the problem inside certain acceptable levels allow the network to return to the Impaired state;
  - *Recovery measures*: the deployment of adequate recovery measures that definitively eliminate the threat allow the network to recover to the Normal state.
- *Quarantine state*: the previous detection of a significant infection on local hosts implied the quarantine of the network, blocking all traffic exchanged with other IP networks in the gateway. The transitions from this state are affected by the following action:
  - *Recovery measures*: the deployment of adequate recovery measures that definitively eliminate the threat allow the network to recover to the Normal state.

Figure 1 graphically represents the finite state machine that includes the four states that were presented and the transition actions between them.

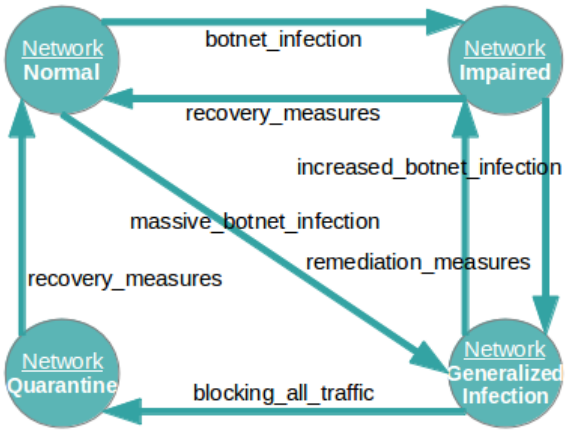

Figure 1. Finite state machine of the local network.

When considering the perspective of an individual host of the local network, the following states and transitions can be identified:

- *Normal state*: the host is not infected with malware. The following transition action will affect this state:
  - *Malware infection*: the detection of malware implies the change of the host to the Infected state.
- *Infected state*: some piece of malware was detected at the host. This state if affected by the following transition actions:
  - *Automatic clean system*: if automatic defenses are able to fight this infection, the system can return to the Normal state;
  - *Filtering malicious traffic*: if the defensive actions cannot automatically clean the system, the malicious traffic must be filtered and the host state will change to Quarantine.
- *Quarantine state*: if the infection cannot be automatically removed, the host must be quarantined. This state can be changed by the following actions:
  - *Manual clean system*: if a manual cleaning of the system with existing tools (like anti-virus) is successful, this implies the host transition to the Cleaned state;
  - *Block all network traffic*: if manual cleaning with existing tools is not possible and additional and more complex tasks are needed, the host transits

 ISBN: 978-1-61208-239-4

to a disconnected mode, with the consequent blocking of all network traffic in the corresponding switch port.

- *Disconnected state*: if the infection cannot be controlled in a short time and is affecting the security and performance of other external elements, then the host must be temporarily disconnected from the network. This state can be changed only by following action:
  - *Offline clean system*: the system is cleaned with the available tools and resources, definitively eliminating the threat. In some cases, a complete system formatting and re-installation might be necessary.
- *Cleaned state*: after the quarantine or disconnected period, the host transits to the cleaned state, where all the previously applied contention measures are removed. The following action will change the system to the Normal state:
  - *Permit all network traffic*: when the threat is definitively eliminated from the host, all the traffic filters that were previously activated can be removed and the host will transit again to normal operation.

The finite state machine that represents all these states and transitions is represented in Figure 2. The dashed lines correspond to actions that occurred in other actors: *filtering_malicious_traffic* is applied at the gateway, while *block_all_network_traffic* is applied at the switch interface. The action *permit_all_network_traffic* is applied in both the gateway and the switch.

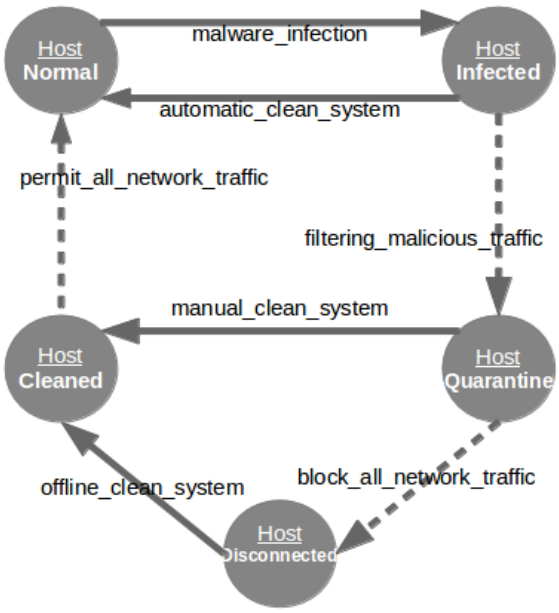

Figure 2. Finite state machine of an individual host.

In the same way, it is relevant to identify and characterize the states of layer two devices (LAN switches). Since these devices physically interconnect the network hosts, they represent the first point available to control the connect/disconnect tasks corresponding to each host:

- *Normal state*: if no actions are taken to disconnect a host from the network, all switch ports are in normal operation (enabled). The following action changes this state:
  - *Block interface*: if the host transits from the Quarantine to the Disconnected state, the corresponding switch interface needs to be blocked (disabled), transiting the switch to the Blocking state.
- *Blocking state*: if a host needs to transit from the Quarantine to the Disconnected state, the corresponding switch port is disabled, blocking the physical connectivity for that host. This state remains active until no more switch interfaces are disabled due to this reason. The following action changes this state:
  - *Release interface*: if no more switch ports are disabled, the switch will come back to the Normal state.

Figure 3 shows the finite state machine corresponding to LAN switches.

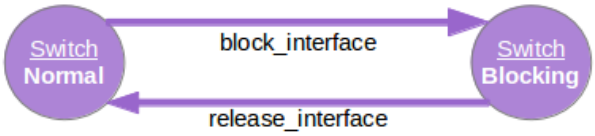

Figure 3. Finite state machine of the LAN switches.

The last relevant actor is the router that interconnects different IP networks of the LAN. The states and actions that characterize this device are:

- *Normal state*: if no malware activities were detected in the local network, the router is operating in the normal state. The following action will change its state:
  - *Filtering malicious traffic*: if malicious activities were detected in the local network and cannot be automatically removed, it is necessary to activate filters that can prevent malicious traffic from going outside.
- *Filtering state*: in this state, the router is filtering malicious traffic and its state can change due to the following actions:
  - *Remove traffic filters*: this action occurs if the threat was definitively removed from the local network. This implies changing the router state to Normal.
  - *Blocking all traffic*: if the threat increased significantly and cannot be contained by using only filters for malicious traffic, it can be necessary to activate more restrictive filters that block all traffic until the threat is eliminated. In this case, the router transits to the Blocking state.
- *Blocking state*: the router is in this state if one or more interfaces need to block all traffic. The Router leaves this state by the influence of the following action:

 ISBN: 978-1-61208-239-4

– *Permit all network traffic*: this action removes the filters that are blocking all traffic from one or more router interfaces and is activated whenever the threats that previously implied the activation of these filters are definitively eliminated.

Figure 4 shows the finite state machine corresponding to the router.

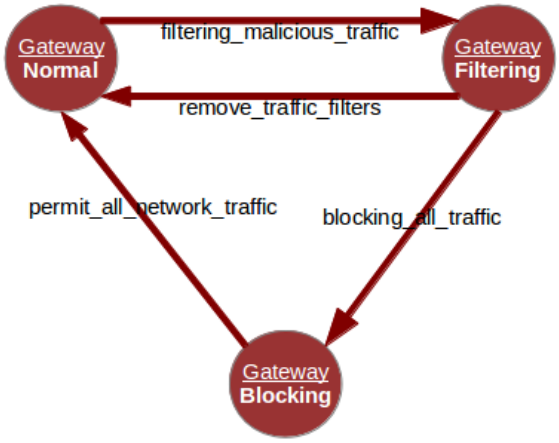

Figure 4. Finite state machine of the local router.

From this discussion, it is clear that the states and transition actions corresponding to the three identified actors are completely interrelated. Figure 5 tries to map the finite state machine of each individual actor with the finite state machine of the network as a whole. The dashed lines represent transitions of an actor from one state to another caused by actions that occurred in another different actor. For example, the Host transits from the Infected to the Quarantine state by the effect of action *filtering_malicious_traffic* that is applied in the Gateway.

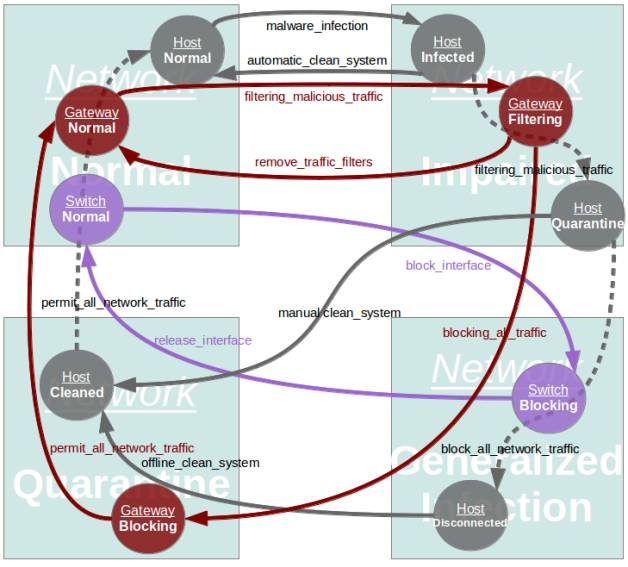

Figure 5. Finite state machine of the overall local network.

The knowledge of the real network state, influenced by the presence of botnet activities, is fundamental to take the right decisions and apply the most effective countermeasures. This knowledge is only possible after inferring all the network model parameters from real and/or reliable network data.

B. From the inference of the model parameters to network management

In a first phase, network data reflecting normal activity and anomalous behaviors induced by the presence of different botnet types should be collected, analyzed and correlated in order to understand which anomalies have occurred and how they can be characterized. The characterization of each anomaly should be as complete as possible, including the amount of data that is generated (alert messages, traffic amount on the different network links, anomalous information on log files, etc.), the timing parameters associated to the anomaly (like, for example, the duration of its characteristic segments) and the transition probabilities between the different states that characterize the anomaly, among other relevant statistics. The data collection step should involve the deployment of laboratorial testbeds where the different security threats can be easily installed in a controlled environment, analyzed and characterized.

The network modeling framework is a multistage space state process able to model the number of error or alert messages and the different states of the network in terms of security threats. Each state is characterized by the type of generation process (deterministic, exponential or other) and its corresponding parameters. The dynamics of the state transitions are heterogeneous and can be ruled by deterministic or exponential processes that define the time of permanence in each state and the destination of the next transition. The modeling framework parameterization will agree with the assumption that state transitions can follow a deterministic or random distribution. State transitions are ruled in parallel by two (or more) parametric matrices that define, respectively, the next transitions after a deterministic amount of time and the probabilistic transitions after a random period of time. The probabilistic/random transitions can follow an exponential distribution (like happens in Markovian models [23] [24]) or any other distribution. The information generation processes associated with each state will also be parameterized by two (or more) vectors defining, respectively, the deterministic values and distribution function parameters for the rates and amount of alert messages generated.

The chain modulated nature of the modeling framework will allow the use of traditional mathematical tools to obtain the model resulting from the superposition of several models or predict the network state at future time instants. The superposition of multiple models (corresponding to different independent networks or different network segments where a certain level of independence can be assumed) can be easily calculated using simple Kronecker sum and product operations [25]. Besides, the chain nature of the resulting model will facilitate the prediction of future network states.

Taking these advantages into account, we believe that the developed network model can be the basis for new

 ISBN: 978-1-61208-239-4

tools that can be intensively used in several network operational and management tasks. The proposed framework can help network managers plan short-term or long-term network reconfigurations and upgrades or design new strategies for network management, traffic routing, service provisioning and other critical network operational issues. The correct planning and location of network failures due to security flaws can greatly increase network operation efficiency and optimize Quality of Service (QoS) parameter values.

IV. Conclusion and Future Work

This paper proposed a network model that is able to describe all network states and the network dynamics in the presence of security threats, specially those originated from botnets, being the first step for a more embracing objective, the development of an integrated framework that is able to identify threats and deploy appropriate countermeasures. All actors, dimensions, states and actions that need to be taken into account at each moment were defined, allowing the future development of appropriate inference procedures that can infer the different model parameters based on real data. Having the ability to model all network states (from a security perspective), events and transitions will be extremely important for network administrators and end users, helping them choose the most appropriate actions/countermeasures for each specific situation. The next step in this work will involve the identification of all relevant network actors and events and the inference of the finite state machine parameters, including the event generation distribution corresponding to each state and the transition probabilities between states.

Acknowledgement

This research was supported by Fundação para a Ciência e a Tecnologia, under research project PTDC/EEA-TEL/101880/2008.

References

[1] S. Goodman and H. Lin, *Toward a Safer and More Secure Cyberspace*, National Academies Press, 2007.

[2] J. P. G. Sterbenz, D. Hutchison, E. K. Cetinkaya, A. Jabbar, J. P. Rohrer, M. Scholler, and P. Smith, *Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines*, Elsevier Computer Networks, vol. 54, Issue 8, pp. 1245-1265, June 2010.

[3] S. W. Korns and J. E. Kastenberg, *Georgia's Cyber Letf Hook*, 2009.

[4] N. Falliere, L. Murchu, and E. Chien, *W32.Stuxnet Dossier*, Version 1.4, Symantec Security Response, February 2011.

[5] C. E. Landwehr, *Computer Security*, International Journal of Information Security, 1, pp. 3-13, 2001.

[6] D. Plohmann, E. Gerhards-Padilla, and F. Leder, *Botnets: Detection, Measurement, Disinfection & Defense*, European Network and Information Security Agency (ENISA), 2011.

[7] *ITU Botnet Mitigation Toolkit*, ICT Applications and Cybersecurity Division, Policies and Strategies Department, ITU Telecommunication Development Sector, 2008.

[8] B. Saha and A. Gairola, *Botnet: An Overview*, CERT-In White Paper CIWP-2005-05, 2005.

[9] M. Fossi, G. Egan, K. Haley, E. Johnson, T. Mack, T. Adams, J. Blackbird, M. Low, D. Mazurek, D. Mckinney, and P. Wood, *Symantec Internet Security Threat Report: Trends for 2010 (Volume 16)*, Symantec Corp, April 2011.

[10] F. Leder, T. Werner, and P. Martini, *Proactive Botnet Countermeasures: an Offensive Approach*, The Virtual Battlefield: Perspectives on Cyber Warfare 3. pp. 211-225, 2009.

[11] M. Fossi, D. Turner, E. Johnson, T. Mack, T. Adams, J. Blackbird, S. Entwisle, B. Graveland, D. Mckinney, J. Mulcahy, and C. Wueest, *Symantec Global Internet Security Threat Report: Trends for 2009 (Volume XV)*, Symantec Corp, April 2010.

[12] Honeynet Project and Research Alliance website. [Online]. Available: http://www.honeynet.org [Accessed: 17 April 2012].

[13] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and A. Arbor, *A Survey of Botnet and Botnet Detection*, Third International Conference on Emerging Security Information, Systems and Technologies, IEEE Computer Society, 2009.

[14] M. Feily, A. Shahrestani, and S. Ramadass, *A Survey of Botnet Technology and Defenses*, CATCH '09 Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security, IEEE Computer Society, 2009.

[15] Y. Xie , F. Yu , K. Achan , R. Panigrahy , G. Hulten, and I. Osipkov, *Spamming Botnets: Signatures and Characteristics*, ACM SIGCOMM Computer Communication Review, vol. 38 no. 4, 2008.

[16] Snort website. [Online]. Available: http://www.snort.org [Accessed: 2 May 2012].

[17] G. Gu, J. Zhang, and W. Lee, *BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic*, Proceedings of 15st Annual Network and Distributed System Security Symposium, 2008.

[18] M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. Hamlen, *Flow-based identification of botnet traffic by mining multiple log files*, Proceedings of International Conference on Distributed Frameworks & Applications, Penang, Malaysia, 2008.

[19] S. Charney, *Collective Defense: Applying Public Health Models to the Internet*, Security & Privacy, IEEE , vol. 10, Issue 2, pp. 54-59, 2012.

[20] Z. Li and Q. Liao. *Botnet economics: uncertainty matters* In M. Johnson, ed., Managing Information Risk and the Economics of Security: 245-267. Springer, 2008.

[21] C. Landwehr. *Formal Models for Computer Security*, ACM Computing Surveys (CSUR), vol. 13 no. 3, pp. 247-278, 1981.

[22] P. Smith, D. Hutchison, J. Sterbenz, M. Schöller, A. Fessi, M. Karaliopoulos, C. Lac, and B. Plattner, *Network Resilience: A Systematic Approach*, IEEE Communications Magazine, July 2011.

 ISBN: 978-1-61208-239-4

[23] A. Nogueira, P. Salvador, R. Valadas, and A. Pacheco. *Fitting self-similar traffic by a superposition of MMPPs modeling the distribution at multiple time scales*, IEICE Transactions on Communications, E84-B(8), 2134-2141.

[24] A. Nogueira, P. Salvador, R. Valadas, and A. Pacheco. *Hierarchical approach based on MMPPs for modeling self-similar traffic over multiple time scales*, Proceedings of the First International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks, 2003.

[25] R. A. Horn, and C. R. Johnson. *Topics in Matrix Analysis*, Cambridge University Press, p. 208, 1994.

 ISBN: 978-1-61208-239-4