Naggen: a Network Attack Graph GENeration Tool

– IEEE CNS 17 Poster –

Martín Barrère and Emil C. Lupu
Department of Computing, Imperial College London, UK
{m.barrere, e.c.lupu}@imperial.ac.uk

***Abstract*—Attack graphs constitute a powerful security tool aimed at modelling the many ways in which an attacker may compromise different assets in a network. Despite their usefulness in several security-related activities (e.g. hardening, monitoring, forensics), the complexity of these graphs can massively grow as the network becomes denser and larger, thus defying their practical usability. In this presentation, we first describe some of the problems that currently challenge the practical use of attack graphs. We then explain our approach based on core attack graphs, a novel perspective to address attack graph complexity. Finally, we present Naggen, a tool for generating, visualising and exploring core attack graphs. We use Naggen to show the advantages of our approach on different security applications.**

I. Background

Over the last years, the rise in cyber security threats combined with the sustained growth of networks involving multiple classes of devices has become an extremely challenging problem to network and security administrators. In 2016 alone, 6449 vulnerabilities have been officially published by NIST and approximately 58% of them have been classified as high severity or critical vulnerabilities [1]. That is roughly 10 new non-negligible vulnerabilities per day. As the Internet of Things (IoT) continues to evolve and the involvement of cyber-physical systems comes into scene, understanding security risks at a large scale becomes even more convoluted [2]. It has been also shown however, that a considerable number of cyber attacks repeatedly leverage non-exotic and sometimes quite old vulnerabilities [3]. According to [4], 99% of exploited vulnerabilities by 2020 will be publicly known by the security community for at least one year. Therefore, there is a critical need for efficient tools able to handle available security information and help security practitioners to broadly analyse network weaknesses and timely understand how security issues might be combined to amplify the attack surface.

In that context, the objective of an attack graph [5], [6] is to depict the different ways in which an attacker may compromise assets in a network. Despite the theoretical beauty of the concept and its benefits on many areas of network security (e.g. network hardening, security monitoring, risk analysis, forensic investigations), the practical use of attack graphs has remained a challenging problem since their initial conception on the early 2000s. Attack graph complexity is one of the main issues that prevent attack graphs from being widely used in practice. As networks become larger and denser, such complexity inherently defies scalability aspects not only at a computational level but also from a human understanding perspective. Even though different narrowing techniques have been proposed in the past [7], [8], their application on dense scenarios can still yield complex and often cyclic attack graphs. Such complexity inevitably prevents attack graphs from conveying the appropriate levels of security insight, and therefore, challenging their usability on practical network settings.

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC), UK, under grant number EP/L022729/1.

II. Network security analysis with Naggen and Core Attack Graphs

Our research work aims at addressing attack graph complexity from a novel perspective. The proposed approach relies in identifying the main attack avenues towards specific network targets by performing a structural summarisation process over the input network. The process essentially summarises alternative routes between any two directly connected nodes and only keeps those routes than cannot be summarised into any other link in the graph. As a result, the obtained graphs present simpler structures which in turn can be further explored and analysed in a hierarchical manner. We call these graphs core attack graphs, or simply, core graphs. The compactness provided by core graphs permits to reduce attack analysis complexity, handle network cycles, ease visualisation aspects and support efficient subsequent analysis. Our theoretical and practical results strongly suggest that our approach can widely support

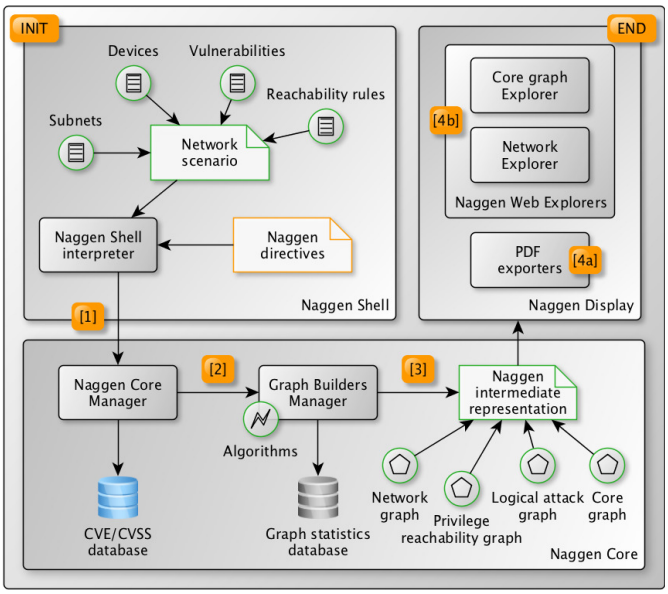

Fig. 1: Naggen high-level architecture

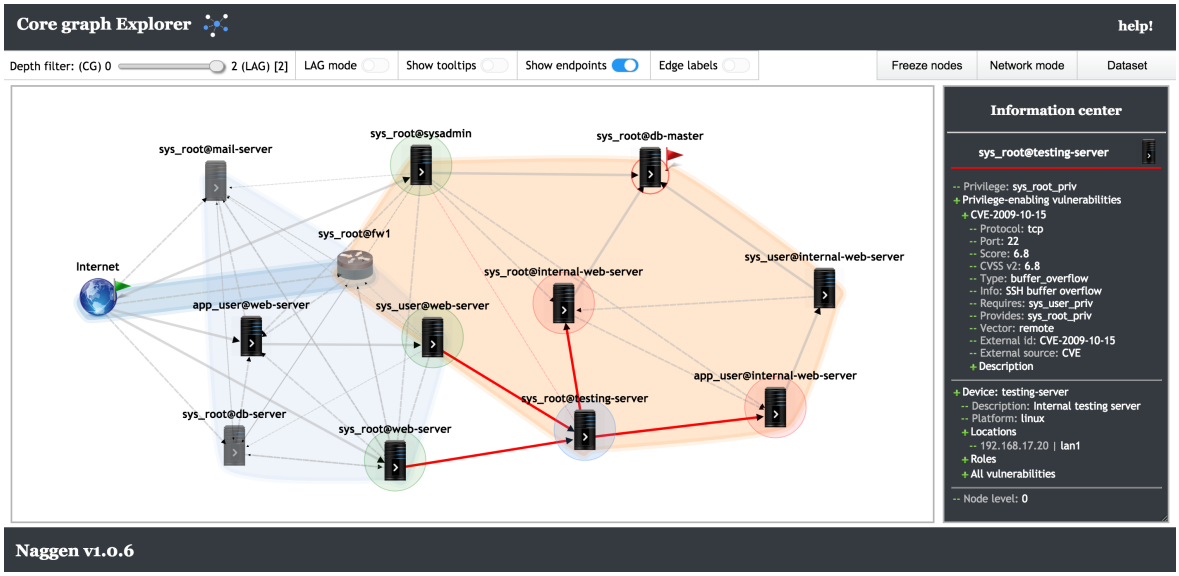

Fig. 2: Naggen snapshot: Core Graph explorer [9]

the actual use of attack graphs in practice and therefore, having a significant impact in network security terms.

In that context, Naggen (Network Attack Graph GENerator) is a security tool aimed at the generation and visualisation of core graphs. The overall system architecture is illustrated in Fig. 1. Naggen is composed of three main building blocks: (1) Naggen Shell, a command-line interface that allows to configure and control the generation process, (2) Naggen Core, responsible for the analysis and graph generation processes, and (3) Naggen Display, which contains different visualisation mechanisms to display the generated attack graphs.

The overall operation in Naggen is as follows. The process starts by specifying and loading a scenario that contains information about the network under analysis, including data about the subnets, the devices in those subnets, the vulnerabilities present in the devices, and reachability rules, i.e., which device can reach each other and on which ports. At step 1, this information is combined with specific directives (e.g. source and target specification, graph generation type) to invoke the Naggen Core manager. The Naggen Core manager organises the overall generation process according to the specified directives and enriches the input specification model with external security information (e.g. CVSS scores). At step 2, different graph builders and analysers are invoked to generate the requested graphs. This component contains the main generation algorithms and produces, at step 3, an intermediate representation file (JSON format) which describes the generated graphs (e.g. network graph, privilege reachability graph, logical attack graph, core graph). These graph representations are then used for visualisation purposes, either in PDF format (4a) or using the Network and Core Graph explorers (4b). The explorers are interactive Web-based interfaces (HTML, JavaScript and D3.js [10]), as shown in Fig. 2, intended to display the generated attack graphs and provide the tools to explore them.

III. Presentation and Demo

This presentation looks into the main issues that currently affect the practical use of attack graphs and brings forward a novel approach to deal with attack graph complexity using core graphs. Besides supporting this discussion with our framework illustrated in the poster, we will perform a live demo over a real case study using Naggen [9] on a standard notebook. We will first explore the network scenario and show how the standard logical attack graph looks like for the proposed example. Then, we will show the advantages of using core graphs on dense scenarios and will present three potential security applications using Naggen: (1) network monitoring and hardening, (2) security perimeters, and (3) forensic investigations. Finally, we will use these visual outcomes to stimulate and engage into enriching discussions about state-of-the-art techniques, research issues yet to be solved, and future steps including probabilistic risk analysis, security metrics and forensic methodologies.

References

[1] "National Vulnerability Database, NIST." https://nvd.nist.gov/. Cited: June 2017.

[2] "Questions and Answers: Responding to the 2017 Security Landscape." https://www2.fireeye.com/WEB-RPT-2017-Cyber-Security-Predictions.html. Cited: June 2017.

[3] "How the Rise in Non-Targeted Attacks Has Widened the Remediation Gap." https://kennasecurity.com/. Cited: June 2017.

[4] "Gartner's Top 10 Security Predictions 2016." http://www.gartner.com/smarterwithgartner/top-10-security-predictions-2016/. Cited: June 2017.

[5] K. Kaynar, "A taxonomy for attack graph generation and usage in network security," *Journal of Inf. Security and Applications*, 2016.

[6] R. Lippmann and K. Ingols, *An Annotated Review of Past Papers on Attack Graphs*. Technical report, MIT, Lincoln Laboratory, 2005.

[7] S. Jajodia and S. Noel, "Topological Vulnerability Analysis," in *Cyber Situational Awareness: Issues and Research* (S. Jajodia, P. Liu, V. Swarup, and C. Wang, eds.), pp. 139–154, Springer US, 2010.

[8] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, Graph-based Network Vulnerability Analysis," *In Proc. of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, p. 217, 2002.

[9] "Naggen - Network Attack Graph GENerator." http://demo.naggen.org. Cited: June 2017.

[10] "D3.js - Data Driven Documents." https://d3js.org/. Cited: June 2017.