ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1583

Using a Deep Understanding of Network Activities for Network Vulnerability Assessment

Mona Lange[1] and **Felix Kuhr**[2] and **Ralf Möller**[3]

Abstract. In data-communication networks, network reliability is of great concern to both network operators and customers. Therefore, network operators want to determine what services could be affected by software vulnerabilities being exploited that are present within their data-communication network. To determine what services could be affected by a software vulnerability being exploited, it is fundamentally important to know the ongoing tasks in a network. A particular task may depend on multiple network services, spanning many network devices. Unfortunately, dependency details are often not documented and are difficult to discover by relying on human expert knowledge. In monitored networks huge amounts of data are available and by applying data mining techniques, we are able to extract information of ongoing network activities. From a data mining perspective, we are interested to test the potential of applying data mining techniques to real-life applications.

1 Introduction

Over the last few years, approximately 2500 software vulnerabilities were discovered every year [13]. The United States intelligence community has identified malicious actors exploiting cyberspace as a top national security threat [3]. Furthermore, IBM's 2015 cyber security intelligence index reveals that approximately half of all cyber attacks originate from within a company's own network [4]. Hence, network devices that are not connected to the internet also have to be considered as potential entry points for cyber attacks. Due to the large number of software vulnerabilities and the security threat they impose, understanding their impact on a monitored network has become an important objective. So network administrators are faced with the challenge of assessing the security impact of vulnerabilities on the network in order to choose appropriate mitigation actions.

For deriving how susceptible a network is to attackers exploiting software vulnerabilities, it is essential to understand what ongoing network activities could potentially be affected by a cyber attack. A network is built with a higher purpose or mission in mind and this mission leads to interactions of network devices and applications causing network dependencies. A monitored infrastructure's mission can be derived through human labor, however missions are subject to frequent change and often knowledge of how an activity links to network devices and applications is not available. This is why an automatic network service dependency methodology called Mission Oriented Network Analysis (MONA) [9] was introduced to derive these missions as network activity patterns. MONA was compared to three state of the art network service dependency discovery methodologies: NSDMiner [11], Sherlock [1] and Orion [2]. NSDMiner addresses the same problem of network service dependency for network stability and automatic manageability. Sherlock is another approach, which learns an inference graph of network service dependency based on co-occurrence within network traffic. A well-known approach is called Orion, which was developed to use spike detection analysis in the delay distribution of flow pairs to infer dependencies. MONA was compared via F-measures to all these state of the art methodologies and was shown to have a better performance. Current network vulnerability approaches [10] focus on identifying critical nodes in a network without focusing on the impact of software vulnerabilities. Even though, software vulnerabilities can be remotely exploitable and sometimes even exploits are readily available online, network vulnerability assessment currently does not take currently present known vulnerabilities into account.

Developing a deeper understanding of network activities allows network vulnerability assessment to analyze what network services would be potentially be affected by a software vulnerability that was detected in a monitored network. Knowing what network activities would be affected by a software vulnerability being exploited, supports network operators in developing a deeper understanding on how their network is affected by software vulnerabilities.

2 Network Vulnerability Assessment

Network vulnerability assessment consists of two parts: detecting present software vulnerabilities in a monitored network and analyzing a network's sensitivity to particular software vulnerabilities. In a monitored network, vulnerability scanners detect present software vulnerabilities. According to the ISO 27005 standard, a vulnerability is a "weakness of an asset or group of assets that can be exploited by one or more threats" [7]. Whereas an asset is defined by ISO13355 ISO/IEC TR13355-1 [6] as being "anything that can have value to the organization, its business operations and their continuity, including information resources that support the organization's mission". The non-profit organization MITRE since 1999 defines common Vulnerabilities and Exposures (CVE) identifiers for software vulnerabilities [8]. The purpose of vulnerability scanning is to identify all software vulnerabilities, which can be linked to a monitored data-communication network.

2.1 Vulnerability scanning

Vulnerability scanning a data-communication network is the process of assessing whether software vulnerabilities can be linked to monitored network devices. Software vulnerabilities can be linked to operating systems, software or firmware [5, 12]. Network vulnerability

[1] Universität zu Lübeck, Germany, email: lange@ifis.uni-luebeck.de
[2] Hamburg University of Technology, Germany, email: felix.kuhr@tuhh.de
[3] Universität zu Lübeck, Germany, email: moeller@uni-luebeck.de

analysis helps network operators to verify whether a software vulnerability linked to an application within the monitored network might endanger ongoing network activities. In the following we rely on the network model introduced in [9]. Vulnerability scanning provides us with a mapping function $SVULN$, which links CVE identifiers $cveId_j$ to network services in a monitored data-communication network. Such that we are able to associate a network service $s_i \in S$

$$SVULN : s_i \rightarrow cveId_j \tag{1}$$

with a $cveId_j$. Given that an operating system of network device d_j is affected by a vulnerability $cveId_i$

$$SVULN : HOSTS(d_j) \rightarrow cveId_i, \tag{2}$$

all hosted network services are linked to the vulnerability. Hence, we assume that an affected operating system will lead to application hosted by that network device being compromised. A software vulnerability with confidentiality impact signifies the threat of information disclosure, in comparison a vulnerability with an integrity impact signifies the threat of data modification and a vulnerability with availability impact could lead to performance degradation. As network activities often span multiple network services for a higher mission, not only network services directly linked to a vulnerability could be affected by an attacker exploiting this vulnerability. All network services relying on requests or responses from a network service linked to a vulnerability with a confidentiality, integrity or availability impact could also be affected.

Consider a vulnerability with a confidentiality impact. Given that a network service is linked to this vulnerability, all information provided by other network services could be leaked as well. Hence, these network services would also be affected by this vulnerability. A network service, which is linked to a vulnerability with an integrity threat, implies that requests sent from this network service could potentially be modified. Similarly, a network service relying on information from another network service, which is linked to a vulnerability with an availability impact, would also be affected by performance degradation of this vulnerability. The set of affected network services AS, which are directly affected by detected software vulnerabilities is defined as

$$AS = SCC\left((\forall s_i \in S : SVULN(s_i), map(asSet, SDEP))\right), \tag{3}$$

where SCC denotes the strongly connected components of the hypergraph given as parameter ($asSet$ maps a tuple into a set of components).

3 Motivating Example

The disaster recovery site of an energy distribution network, provided an Italian water and energy distribution company, was available for non-invasive experimentation. Based on this network, we are able to collect and analyze real-life network traffic and also scan the network for present software vulnerabilities. Figure 1 shows all network service dependencies detected by MONA. These network service dependencies were considered complete and correctly identified by network operators.

Figure 2 shows the result of network dependency based vulnerability assessment for software vulnerabilities CVE-2007-5423 and CVE-2010-2075 that were detected via network scanning on network device mferp2. Both software vulnerabilities can be exploited by automated code. CVE-2007-5423 is a vulnerability that allows remote attackers to execute arbitrary code in TikiWiki 1.9.8

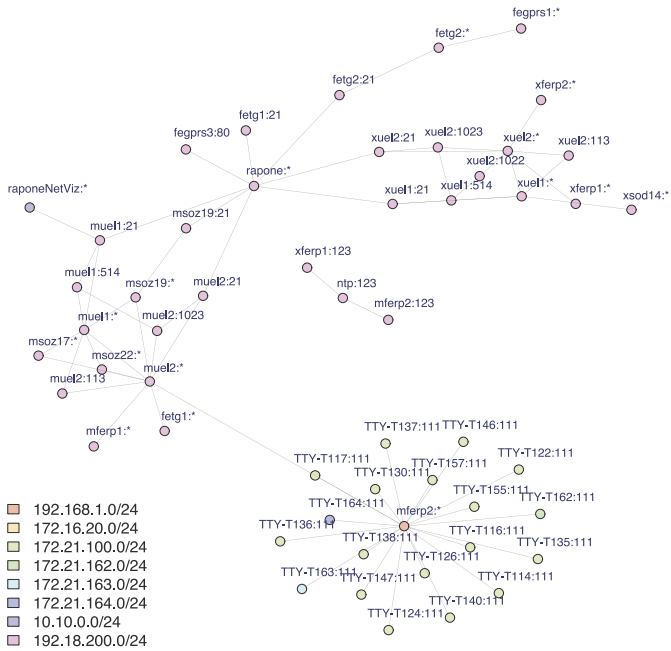

Figure 1: Network service dependency analysis in an energy distribution network.

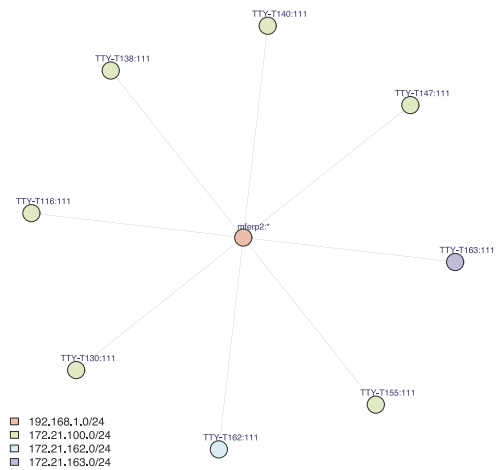

Figure 2: Network dependency based vulnerability assessment for vulnerabilities CVE-2007-5423 and CVE-2010-2075, who were detected on mferp2.

and CVE-2010-2075 is an unauthorized-access vulnerability due to a backdoor in UnrealIRCd 3.2.8.1. TTY-T[116-163] are remote terminal units of substations, which are dependent on requests from the front end server mferp2. Hence, network based vulnerability assessment concludes that TTY-T[116-163] are affected by CVE-2007-5423 and CVE-2010-2075, which were detected on mferp2.

4 Conclusion

We have introduced a novel network based vulnerability analysis approach. Network service dependency analysis allows the automatic detection of ongoing network activities. Based on automatically detected network service dependencies, we are able to link exploitable software vulnerabilities to ongoing network activities. The proposed framework is fully automated and is able to integrate vulnerability specification from the bug-reporting community and helps network operators develop a deeper understanding on how networks are affected by software vulnerabilities.

Acknowledgments

This work was partly supported by the Seventh Framework Programme (FP7) of the European Commission as part of the PANOPTESEC integrated research project (GA 610416).

REFERENCES

[1] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A Maltz, and Ming Zhang, 'Towards highly reliable enterprise network services via inference of multi-level dependencies', in *ACM SIGCOMM Computer Communication Review*, volume 37, pp. 13–24. ACM, (2007).

[2] Xu Chen, Ming Zhang, Zhuoqing Morley Mao, and Paramvir Bahl, 'Automating network application dependency discovery: Experiences, limitations, and new solutions.', in *OSDI*, volume 8, pp. 117–130, (2008).

[3] James R. Clapper. Statement for the record, worldwide threat assessment of the us intelligence community. http://www.dni.gov/index.php/newsroom/testimonies/209-congressional-testimonies-2015/1174-statement-for-the-record-worldwide-threat-assessment-of-the-u-s-ic-before-the-sasc, 2014.

[4] IBM Corporation. 2015 cyber security intelligence index, july 2015.

[5] Bhadreshsinh G Gohil, Rishi K Pathak, and Axaykumar A Patel, 'Federated network security administration framework', (2013).

[6] ISO ISO and IEC Std, *ISO/IEC 13335-1: 2004*, 2004.

[7] ISO ISO and IEC Std, 'Iso 27005: 2011', *Information technology–Security techniques–Information security risk management. ISO*, (2011).

[8] MITRE. Common vulnerabilities and exposures. https://cve.mitre.org/, 2000.

[9] Ralf Möller Mona Lange, 'Time Series Data Mining for Network Service Dependency Analysis', in *The 9th International Conference on Computational Intelligence in Security for Information Systems*. Springer International Publishing, (2016).

[10] Alan T Murray, 'An overview of network vulnerability modeling approaches', *GeoJournal*, **78**(2), 209–221, (2013).

[11] Arun Natarajan, Peng Ning, Yao Liu, Sushil Jajodia, and Steve E Hutchinson, *NSDMiner: Automated discovery of network service dependencies*, IEEE, 2012.

[12] The Government of the Hong Kong Special Administrative Region. An overview of vulnerability scanners. http://www.infosec.gov.hk/english/technical/files/vulnerability.pdf, 2008.

[13] TechTarget. Growing threats make security vulnerability management essential. http://searchsecurity.techtarget.com/video/Growing-threats-make-security-vulnerability-management-essential, 2015.