

Chapter 2

Related Work

In this chapter, we provide a brief review of related work, including attack graphs and applications, existing network hardening techniques, and other relevant topics, such as alert correlation and security metrics.

2.1 Attack Graph

A number of tools are available for scanning network vulnerabilities, such as Nessus [12], Nmap [35], Snort [53], Cisco security scanner, System Scanner by ISS [21], CyberCop [7] and Computer Oracle and Password System (COPS) [14]. Those solutions are being applied in real world networks and they can help to identify weaknesses and flaws in such networks. However, most existing solutions usually identify vulnerabilities or attacks in isolation, which only provides a partial picture about securing a network since many attackers may employ sophisticated multi-step attacks to breach a seemingly well guarded network.

Attack graphs are constructed by analyzing the inter-dependency between vulnerabilities and security conditions that have been identified in the target network [2, 8, 14, 22, 40, 44, 51, 52, 56, 60, 73]. Such analysis can be either forward starting from the initial state [44, 60] or backward from the goal state [51, 56]. Model checking was first used to analyze whether the given goal state is reachable from the initial state [49, 51] but later used to enumerate all possible sequences of attacks between the two states [24, 56].

More specifically, Phillips and Swiler [44, 60] propose the concept of attack graph and they also present a graph-based approach for generating attack graphs. In their model, the inputs of an attack graph include configuration files, attacker profiles, and a database of attack templates, which must be manually created. The nodes of the attack graph are attack templates instantiated with particular users and machines, whereas edges are labeled with probabilities of success or cost of attacks.

The graphs can be analyzed to find the shortest paths between given start and end nodes. The idea of grouping similar nodes is mentioned although the correctness critically depends on identical configuration among such nodes.

Another model [61] expresses attack graphs with the require and provide approach using the precondition and postcondition of each exploit. For each successful attack, the attacker can obtain the ability to perform more attack steps so each successful exploitation increases his/her capabilities in launching new attacks. An attack specification language JIGSAW is used in describing attack steps. The language requires low level details of capabilities and requirements of attacks which may be hard to obtain. This require and provide approach brings flexibility in discovering potentially new attack scenarios. Using this language and the given specifications, an IDS and attack analysis system can be created.

In [51], model checking is applied to the analysis of multi-step network attacks. Known vulnerabilities on network hosts, connectivity between hosts, initial capabilities of the attacker are described as states and exploits as transitions between states. This model is given to a model checker as its input and the reachability in terms of given goal states is given as a query. The model checker then produces a counterexample if a sequence of exploits can lead to goal states. Such a sequence of exploits indicates a potential attack that must be avoided to secure the network. The term *topological vulnerability analysis* is coined in [52] which provides more details on how connectivity should be modeled at different layers.

In [24, 56], model checking is used for a different purpose, that is to enumerate all attack paths. A modified model checker is used to take as input the finite-state machine created from network information. The model checker provides all counterexamples to a query about the safety of the goal states. Those are essentially the possible attack paths. Other types of analysis are also discussed by the authors, including how to find a *cut set* in the attack graph, such that goal conditions can no longer be reached. The problem of finding the minimum possible attack that leads to the given goal conditions is shown to be intractable. One apparent limitation of this approach is that all attack paths are explicitly enumerated in its result, which leads to a combinatorial explosion.

A *monotonicity assumption* is adopted in [2] to address the scalability of model checking-based approaches. It states exploits will never cause the attacker to relinquish any previously obtained privileges. Attack paths can then be implicitly modeled as paths in a directed graph including exactly one copy of each exploit and its pre- and post-conditions; edges interconnect exploits to their conditions. The assumption thus reduces the complexity of attack graph from exponential to polynomial in the number of hosts. However, it also makes some attacks impossible if they disable services or invalidate vulnerabilities. Attack graphs are generated using a two-pass search that first links exploits by starting from the attacker's initial state and then removes those irrelevant states by searching backward from the goal state.

In [41] the authors proposed a logic-based approach to attack graph generation to improve the efficiency. Using this approach, the generated attack graph always has a polynomial size in the size of analyzed network. A Network security analyzer

MulVAL is used to build the attack graph generation tool. Here, each node of the attack graph represents a logical statement and the edges define the relationship between network configuration and what the privileges the attacker potentially could gain. Here, the main focus is on the root causes of the attack. Using this logical attack graph representation, one can obtain all the possible attack scenarios using a simple depth-first search. Using this representation, it can also be ensured that an attack graph will always have a polynomial size in the size of the network.

Attack graphs have been used for correlating intrusion alerts into attack scenarios [36, 67, 68]. Such alert correlation methods are parallel to our work, because they aim to employ the knowledge encoded in attack graphs for detecting and taking actions against actual intrusions, whereas our work aims to harden the network before any intrusion may happen. The relationship between those methods and our work is analogous to that between IDSs and vulnerability scanners, although IDSs and vulnerability scanners work on the alert and vulnerability level, whereas the alert correlation methods and our methods work at a higher level, i.e., the attack scenarios. In forensics, attack graph is also used to find probable attacks and to assess damages of the system [59]. For taking legal action against the attacker, the analyst needs to show attack steps as evidence. Using attack graph, the whole attack paths can be matched to data extracted from IDS logs.

2.2 Network Hardening

In [56] a *minimum critical set* is computed which is basically the minimum set of exploits in the attack graph by disabling which we can harden the network. The minimum critical attack set is essentially the concept of a cut set in graph theory. In [2], the authors propose an algorithm called findMinimul to find the attack that takes the least number of steps from the initial state to the goal condition to launch an attack. Another such approach proposed in [24] introduces the minimum critical attack set in a more scalable way. However, all these solutions cannot be directly used by system administrators as the set of exploits to be disabled are unavoidable consequences of other exploits, which must be disabled in the first place.

We introduce the concept of network hardening with respect to initially satisfied conditions in [37] and later refine it in [69]. In contrast to the above minimum critical set-based approaches, we argue that disabling initial conditions is a better choice with respect to the need of security administrators. To disable exploits in the critical set, we have to also disable the causes of such exploits, which ultimately leads to a set of security conditions that are initially satisfied and do not depend on others. Such initial conditions are only the pre-condition of some exploits but they are not the post condition of any exploit. This indicates that these conditions can be disabled independently. Therefore, an effective hardening measure is to find the set of initial conditions disabling which can disable the goal conditions. However, as we show in [1], this approach has an unavoidable exponential worst-case complexity, because the result itself may be exponential in size. So, for larger networks, enforcing this

approach will be costly or even impossible depending on the number of hosts and their connectivity. Therefore, we address this issue with a heuristic approach that yields reasonably good results in significantly less time. In this book, we develop a more comprehensive and coherent study on network hardening based on those previous work.

2.3 Other Related Topics

Parallel to network hardening, alert correlation aims to understand multi-step attacks from the attack's point of view. It reconstructs multi-step attack scenarios from isolated alerts. This may employ prior knowledge about attack strategies [6, 10, 11, 13] or the causal relationships between attacks [5, 32, 33]. Those methods may either aggregate alerts with similar attributes [4, 9, 58, 63] or statistical patterns [25, 46]. Hybrid approaches also exist that combine different techniques for better results [33, 47, 71]. Alerts missed by IDSs can be tolerated by clustering alerts with similar attributes [34], and incomplete knowledge can be pieced together through statistical analyses [46, 47]. Alert correlation techniques are also used for other purposes such as to relate alerts to the same thread of attacks [20]. The privacy issue of alert correlation is investigated in [72]. Alert correlation is a potential method for dealing with insider attacks in [48, 50]. Existing efforts on integrating information from different sources exist, such as the model in *M2D2* [31] and the Bayesian network-based approach [74]. Some commercial products claim to support realtime analyses of alerts such as Tivoli Risk Manager [19].

An interesting future direction in network hardening is to integrate the network hardening techniques with various efforts on security metrics and security quantification. There exist numerous standardization efforts on security metrics, such as the Common Vulnerability Scoring System (CVSS) [30] and, more recently, the Common Weakness Scoring System (CWSS) [62]. The former focuses on ranking known vulnerabilities, whereas the latter on software weaknesses. Both CVSS and CWSS measure the relative severity of individual vulnerabilities in isolation and do not address their overall impact. On the other hand, these efforts form a practical foundation for research on security metrics, as they provide security analysts and vendors standard ways for assigning numerical scores to known vulnerabilities which are already available in public vulnerability databases, such as the National Vulnerability Database (NVD) [38]. On the research front, network security metrics has attracted much attention, as surveyed in [23, 54, 64]. In particular, the Network Compromise Percentage Metric (NCP) is proposed while evaluating the so-called *defense in depth* strategy using attack graphs [27], which basically indicates the percentage of network assets that may be compromised by attackers. In another work [29], the authors rank states in an attack graph based on probabilities of attackers reaching these states during a random simulation; the PageRank algorithm is adapted for such a ranking; a key assumption made in this work is that attackers would progress along different paths in an attack graph in a random fashion.

A similar work replaces attack trees with more advanced attack graphs and replace attack paths with attack scenarios [43]. A Mean Time-to-Compromise metric is proposed based on the predator state-space model (SSM) used in the biological sciences in [26]; defined as the average time required for compromising networks, the metric provides richer semantics than other abstract metrics; the main limitation of this work lies in an oversimplified model of network intrusions and differences between vulnerabilities. A recent work proposes a risk management framework using Bayesian networks to quantify the chances of attacks and to develop a security mitigation and management plan [45]. Another recent study of several CVSS-based vulnerability metrics shows the correlation between those metrics and the time to compromise of a system [17]. In our recent work, we have proposed a general framework for designing network security metrics [70], Bayesian network-based metrics [15], a probabilistic approach [65], a k -zero day safety metric [66], a base metric-level framework [3], and a mean time-to-compromise approach [39]. Parallel to our work on probabilistic security metrics, the authors in [18] address several important issues in calculating such metrics including the dependencies between different attack sequences in an attack graph and cyclic structures in such graphs.

Another interesting future direction in network hardening is to address the threat of *zero day attacks*. Most existing work focus on known vulnerabilities in a network. A few exceptions include an empirical study on the total number of zero day vulnerabilities available on a single day based on existing facts about vulnerabilities [28], a report on the popularity of zero day vulnerabilities among attackers [16], an empirical study on software vulnerabilities' life cycles [55], and more recently an effort on estimating the effort required for developing new exploits [57]. Finally, in the context of software security, the attack surface metric measures how likely a software is vulnerable to known or unknown attacks based on the degree of exposure [42].

References

1. Massimiliano Albanese, Sushil Jajodia, and Steven Noel. Time-efficient and cost-effective network hardening using attack graphs. In *Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, Boston, MA, USA, June 2012.
2. P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of ACM CCS'02*, 2002.
3. Pengsu Cheng, Lingyu Wang, Sushil Jajodia, and Anoop Singhal. Aggregating cvss base scores for semantics-rich network security metrics. In *Proceedings of the 31st IEEE International Symposium on Reliable Distributed Systems (SRDS 2012)*, pages 31–40. IEEE Computer Society, 2012.
4. F. Cuppens. Managing alerts in a multi-intrusion detection environment. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01)*, 2001.
5. F. Cuppens and A. Mieke. Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 187–200, 2002.

6. F. Cuppens and R. Ortalo. LAMBDA: A language to model a database for detection of attacks. In *Proceedings of the 3rd International Symposium on Recent Advances in Intrusion Detection (RAID'01)*, pages 197–216, 2001.
7. Network associates, cybercop scanner. Available at http://www.nss.co.uk/grouptests/va/edition2/nai_cybercop_scanner/nai_cybercop_scanner.htm.
8. M. Dacier. Towards quantitative evaluation of computer security. Ph.D. Thesis, Institut National Polytechnique de Toulouse, 1994.
9. O. Dain and R.K. Cunningham. Building scenarios from a heterogeneous alert system. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, 2001.
10. O. Dain and R.K. Cunningham. Fusing a heterogeneous alert stream into scenarios. In *Proceedings of the ACM Workshop on Data Mining for Security Applications*, pages 1–13, 2001.
11. H. Debar and A. Wespi. Aggregation and correlation of intrusion-detection alerts. In *Proceedings of the 3rd International Symposium on Recent Advances in Intrusion Detection (RAID'01)*, pages 85–103, 2001.
12. R. Deraison. Nessus scanner, 1999. Available at <http://www.nessus.org>.
13. S.T. Eckmann, G. Vigna, and R.A. Kemmerer. STATL: An attack language for state-based intrusion detection. *Journal of Computer Security*, 10(1/2):71–104, 2002.
14. D. Farmer and E.H. Spafford. The COPS security checker system. In *USENIX Summer*, pages 165–170, 1990.
15. M. Frigault, L. Wang, A. Singhal, and S. Jajodia. Measuring network security using dynamic bayesian network. In *Proceedings of 4th ACM QoP*, 2008.
16. A. Greenberg. Shopping for zero-days: A price list for hackers' secret software exploits. *Forbes*, 23 March 2012.
17. Hannes Holm, Mathias Ekstedt, and Dennis Andersson. Empirical analysis of system-level vulnerability metrics through actual attacks. *IEEE Trans. Dependable Secur. Comput.*, 9(6):825–837, November 2012.
18. J. Homer, X. Ou, and D. Schmidt. A sound and practical approach to quantifying security risk in enterprise networks. Technical Report, 2009.
19. IBM. IBM tivoli risk manager. Available at <http://www.ibm.com/software/tivoli/products/risk-mgr/>.
20. SRI International. Event monitoring enabling responses to anomalous live disturbances (EMERALD). Available at <http://www.sdl.sri.com/projects/emerald/>.
21. System Scanner Internet Security Systems. Internet security systems, system scanner. Available at <http://www.iss.net>.
22. S. Jajodia, S. Noel, and B. O'Berry. Topological analysis of network attack vulnerability. In V. Kumar, J. Srivastava, and A. Lazarevic, editors, *Managing Cyber Threats: Issues, Approaches and Challenges*. Kluwer Academic Publisher, 2003.
23. A. Jaquith. *Security Metrics: Replacing Fear Uncertainty and Doubt*. Addison Wesley, 2007.
24. S. Jha, O. Sheyner, and J.M. Wing. Two formal analysis of attack graph. In *Proceedings of the 15th Computer Security Foundation Workshop (CSFW'02)*, 2002.
25. Klaus Julisch and Marc Dacier. Mining intrusion detection alarms for actionable knowledge. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 366–375, 2002.
26. D.J. Leversage and E.J. Byres. Estimating a system's mean time-to-compromise. *IEEE Security and Privacy*, 6(1):52–60, 2008.
27. R. Lippmann, K. Ingols, C. Scott, K. Piwowski, K. Kratkiewicz, M. Artz, and R. Cunningham. Validating and restoring defense in depth using attack graphs. In *Proceedings of the 2006 IEEE conference on Military communications*, MILCOM'06, pages 981–990, Piscataway, NJ, USA, 2006. IEEE Press.
28. M.A. McQueen, T.A. McQueen, W.F. Boyer, and M.R. Chaffin. Empirical estimates and observations of Oday vulnerabilities. *Hawaii International Conference on System Sciences*, 0:1–12, 2009.

29. V. Mehta, C. Bartzis, H. Zhu, E.M. Clarke, and J.M. Wing. Ranking attack graphs. In *Recent Advances in Intrusion Detection 2006*, 2006.
30. P. Mell, K. Scarfone, and S. Romanosky. Common vulnerability scoring system. *IEEE Security & Privacy*, 4(6):85–89, 2006.
31. B. Morin, L. Mé, H. Debar, and M. Ducassé. M2D2: A formal data model for IDS alert correlation. In *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID'02)*, pages 115–137, 2002.
32. P. Ning, Y. Cui, and D.S. Reeves. Constructing attack scenarios through correlation of intrusion alerts. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, pages 245–254, 2002.
33. P. Ning and D. Xu. Learning attack strategies from intrusion alerts. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, 2003.
34. P. Ning, D. Xu, C.G. Healey, and R.S. Amant. Building attack scenarios through integration of complementary alert correlation methods. In *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS'04)*, pages 97–111, 2004.
35. Nmap-network mapper. Available at <http://nmap.org/index.html>.
36. S. Noel and S. Jajodia. Correlating intrusion events and building attack scenarios through attack graph distance. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04)*, 2004.
37. S. Noel, S. Jajodia, B. O'Berry, and M. Jacobs. Efficient minimum-cost network hardening via exploit dependency graphs. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03)*, 2003.
38. National vulnerability database. available at: <http://www.nvd.org>, May 9, 2008.
39. W. Nzoukou, L. Wang, S. Jajodia, and A. Singhal. A unified framework for measuring a network's mean time-to-compromise. In *Proceedings of the 32nd IEEE International Symposium on Reliable Distributed Systems (SRDS 2013)*, pages 215–224, 2013.
40. R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Trans. Software Eng.*, 25(5):633–650, 1999.
41. X. Ou, W.F. Boyer, and M.A. McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS'06, pages 336–345, New York, NY, USA, 2006. ACM.
42. J. Wing P. Manadhata. An attack surface metric. Technical Report CMU-CS-05-155, 2005.
43. J. Pamula, S. Jajodia, P. Ammann, and V. Swarup. A weakest-adversary security metric for network configuration security analysis. In *Proceedings of the ACM QoP*, pages 31–38, 2006.
44. C. Phillips and L. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the New Security Paradigms Workshop (NSPW'98)*, 1998.
45. Nayot Poolsappasit, Rinku Dewri, and Indrajit Ray. Dynamic security risk management using bayesian attack graphs. *IEEE Trans. Dependable Secur. Comput.*, 9(1):61–74, January 2012.
46. X. Qin and W. Lee. Statistical causality analysis of INFOSEC alert data. In *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, pages 591–627, 2003.
47. X. Qin and W. Lee. Discovering novel attack strategies from INFOSEC alerts. In *Proceedings of the 9th European Symposium on Research in Computer Security (ESORICS 2004)*, pages 439–456, 2004.
48. A. R. Chinchani and Iyer, H. Ngo, and S. Upadhyay. Towards a theory of insider threat assessment. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN'05)*, 2005.
49. C.R. Ramakrishnan and R. Sekar. Model-based analysis of configuration vulnerabilities. *Journal of Computer Security*, 10(1/2):189–209, 2002.
50. I. Ray and N. Poolsappasit. Using attack trees to identify malicious attacks from authorized insiders. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS'05)*, 2005.

51. R. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 156–165, 2000.
52. R. Ritchey, B. O’Berry, and S. Noel. Representing TCP/IP connectivity for topological analysis of network security. In *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC’02)*, page 25, 2002.
53. M. Roesch. Snort - lightweight intrusion detection for networks. In *Proceedings of the 1999 USENIX LISA Conference*, pages 229–238, 1999.
54. R.M. Savola. Towards a taxonomy for information security metrics. In *Proceedings of the 3rd ACM QoP*, pages 28–30. ACM, 2007.
55. M. Shahzad, M.Z. Shafiq, and A.X. Liu. A large scale exploratory analysis of software vulnerability life cycles. In *Proceedings of the 34th International Conference on Software Engineering (ICSE)*, 2012.
56. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the IEEE S&P’02*, 2002.
57. Teodor Sommestad, Hannes Holm, and Mathias Ekstedt. Effort estimates for vulnerability discovery projects. In *Proceedings of the 2012 45th Hawaii International Conference on System Sciences, HICSS ’12*, pages 5564–5573, Washington, DC, USA, 2012. IEEE Computer Society.
58. S. Staniford, J.A. Hoagland, and J.M. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10(1/2):105–136, 2002.
59. P. Stephenson. Using formal methods for forensic analysis of intrusion events- - a preliminary examination. white paper. available at <http://www.imfgroup.com/DocumentLibrary.html>.
60. L. Swiler, C. Phillips, D. Ellis, and S. Chakerian. Computer attack graph generation tool. In *Proceedings of the DARPA Information Survivability Conference & Exposition II (DISCEX’01)*, 2001.
61. S. Templeton and K. Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 New Security Paradigms Workshop (NSPW’00)*, pages 31–38, 2000.
62. The MITRE Corporation. Common weakness scoring system. <http://cwe.mitre.org/cwss/>, 2010.
63. A. Valdes and K. Skinner. Probabilistic alert correlation. In *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 54–68, 2001.
64. V. Verendel. Quantified security is a weak hypothesis: a critical survey of results and assumptions. In *Proceedings of the 2009 NSPW*, pages 37–50. ACM, 2009.
65. L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In *Proceedings of the 22nd IFIP DBSec*, 2008.
66. L. Wang, S. Jajodia, A. Singhal, and S. Noel. k-zero day safety: Measuring the security risk of networks against unknown attacks. In *Proceedings of the 15th ESORICS*, pages 573–587, 2010.
67. L. Wang, A. Liu, and S. Jajodia. An efficient and unified approach to correlating, hypothesizing, and predicting intrusion alerts. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS 2005)*, pages 247–266, 2005.
68. L. Wang, A. Liu, and S. Jajodia. Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts. *Computer Communications*, 29(15):2917–2933, 2006.
69. L. Wang, S. Noel, and S. Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, 11 2006.
70. L. Wang, A. Singhal, and S. Jajodia. Measuring network security using attack graphs. In *Proceedings of the 3rd ACM QoP*, New York, NY, USA, 2007. ACM Press.
71. D. Xu and P. Ning. Alert correlation through triggering events and common resources. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC’04)*, pages 360–369, 2004.
72. D. Xu and P. Ning. Privacy-preserving alert correlation: A concept hierarchy based approach. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC’05)*, 2005.

73. D. Zerkle and K. Levitt. Netkuang - a multi-host configuration vulnerability checker. In *Proceedings of the 6th USENIX Unix Security Symposium (USENIX'96)*, 1996.
74. Y. Zhai, P. Ning, P. Iyer, and D. Reeves. Reasoning about complementary intrusion evidence. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04)*, pages 39–48, 2004.

Network Hardening

An Automated Approach to Improving Network Security

Wang, L.; Albanese, M.; Jajodia, S.

2014, VIII, 60 p. 23 illus., Softcover

ISBN: 978-3-319-04611-2