Modeling Cybersecurity Risks

Proof of concept of a holistic approach for integrated risk quantification

Diane Henshel, Alex Alexeev, Mariana Cains
School of Public and Environmental Affairs
Indiana University
Bloomington, IN, USA
dhenshel@indiana.edu

Jeff Rowe
Department of Computer Science
University of California
Davis, CA, USA

Hasan Cam[1], Blain Hoffman[2]
[1]Network Security, [2]Cyber and Networked Systems
US Army Research Laboratory
[1]Adelphi & [2]Aberdeen Proving Ground, MD, USA

Iulian Neamtiu
Department of Computer Science
New Jersey Institute of Technology
University Heights Newark, NJ, USA

Abstract— **Decision-making in cyber-security is mostly ad-hoc and highly reliant on static policies, as well as human intervention. This does not fit current networks/systems, as they are highly dynamic systems where security assessments have to be performed, and decisions have to be made, automatically and in real-time. To address this problem, we propose a risk-based approach to cybersecurity decision-making. In our model, the system undergoes a continuous security risk assessment based on risk; decisions for each action are taken based on constructing a sequence of alternative actions and weighing the cost-benefit trade-offs for each alternative. We demonstrate the utility of our system on a concrete example involving protecting an SQL server from SQL injection attacks. We also discuss the challenges associated with implementing our model.**

Keywords—risk assessment; risk calculation; cybersecurity; dynamic risk; SQL injection

I. SCOPE

In this paper we will define our approach to holistic risk-based cybersecurity decision-making, introduce the integrative risk parameter framework and the need to apply this framework to a cyber network and task / mission / operation, and then present that theoretical application of identified parameterized metrics to the database SQL query operation, with a modeled quantitative interpretation of the results. Finally we will close with a discussion of future work, expansion of the parameterization of the full operation model, with calculation of the risk for each modeled network state, and a discussion of plans to apply this work to a testbed /virtualized network.

II. INTRODUCITON

The current state of the art for cyber security risk assessment, as currently formalized in the two NIST risk assessment framework and risk management guidances [1,2] is focused on assessing risks from an IT manager's perspective, with a goal of helping the IT managers harden the network, identify possible (known) threats, identify (known) vulnerabilities in the network based on hardware, software, and (as much as possible) use of the network, and finally help the IT manager identify and implement the best available (affordable) protections against the known threats for the known vulnerabilities. This approach is based on previously developed approaches to evaluating risk within a networked system, including OCTAVE (Operationally Critical Threat and Vulnerability Evaluation) [3], CVSS (Common Vulnerability Scoring System) [4], CRAMM (CCTA Risk Analysis and Management Method) [5], and CORAS [6] among others. Virtually all of the methodologies focus on vulnerability estimation of impact based on cost of replacement, or estimation of impact on Confidentiality of information, Integrity of information, files, and software, and Accessibility of the network services. And several consider management of system recovery (as emphasized in the NIST framework). Other classical engineering approaches include using fault trees to quantify the causes of an outcome, and event trees which quantify the probabilities of different outcomes, adverse and non-adverse. A number of studies use graphical analysis to predict individual outcomes [7,8].

Virtually all of the risk assessment and risk management systems being used for networks are reactive, determining risk in a system based on known vulnerabilities and risks, and calculated statically, determined quantitatively, semi-quantitatively, or qualitatively once, but not continuously. Yet in any cyber system, the risks change dynamically. First of all, humans change the risk in a system dynamically. Attackers observe and respond and change tactics based on what they see in the network, the system itself is constantly changing states, which can affect risk. Users change how they interact with the system. And defenders can respond to the risks they interpret in the system, and the system states, for example, by altering hardware settings, software settings, activating defenses, or changing the operating environment. Secondly, the system states change dynamically in response to the last activity in the system. Risks and vulnerabilities (which contribute to the risk)

The researchers are sponsored by the U.S. Army Research Laboratory (ARL) Cybersecurity Collaborative Research Alliance (Csec CRA). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ARL, Department of Defense, Indiana University, New Jersey Institute of Technology or any official policies of any of these entities.

are dependent on the hardware and the software in the system, as well as system policies which control hardware, software and human interactions in the system. To fully model cyber security risk, the relevant vulnerabilities in the software and hardware, as well as the human interactions and the interactions within the system would ideally be modeled, in order to best predict impact (that is, the manifestation of system risk).

In current cyber security practice, most decisions and actions are ad-hoc: driven by experience and a rigid (uni-dimensional) policy, rather than by quantitative analysis. Moreover, while security decisions involve a cost-benefit analysis, these decisions are made offline (and are hard to change) rather than online, as the system runs. Consider, for example, a typical enterprise setting where users are granted access to the enterprise system using a username & password authentication scheme. A person P with username U might have access to certain enterprise resources depending on the P's position in the organization. The problem is, as the account is being used, risk not being reassessed dynamically. If the password associated with U is compromised, a malicious person M (which now has assumed P's identity) can inflict some serious harm, because the risk is associated with the user account U, not with who (P or M) is actually using that account. The root of this problem is inadequate, static risk characterization whereas cyber networks function as dynamic systems with time dependent and time-sequenced actions whose risk must be reassessed continuously (in the aforementioned case, an online masquerade detection scheme will detect that the risk associated with the account U should be Risk_M (high) rather than Risk_P (low)).

A. Maintaining the Integrity of the Specifications

To address these issues, we propose a novel, cost-benefit approach for risk quantification and decision-making. First, our approach uses a holistic, quantitative model for risk assessment, where risks associated with users, actions, systems, etc. are multi-dimensional and continuously adjusted based on changes in system state. Second, security decisions, e.g., whether to allow a certain action, are taken via a quantitative, cost-benefit analysis, rather than being driven by a rigid a priori-defined policy that rarely changes. For example, in the aforementioned masquerading attack, the risk Risk_U associated with U's actions should immediately change from Risk_P to Risk_M as soon as M gains access to the system. Unfortunately, in current settings, Risk_U = Risk_P until a sysadmin discovers the attack. This is time-consuming, error-prone, and effort-intensive, and dangerous (M can inflict significant harm until the attack is discovered). Our approach involves taking other factors into account, e.g., the sequence of actions performed by U, when assessing the risk associated with that account. As another example, consider deciding whether to allow a remote client to execute an SQL query on a server. On current systems, this decision is made based on client's role or access privilege. In our model, this decision is made based on client risk (e.g., client's geographical location), command risk (is this a new request or a part of a sequence of requests, probability that the command is an SQL injection attack), etc.

More precisely, our risk framework is an iterative decision-making process that models system (hardware, software, network) and human state; for each task, based on known and predicted threats, it constructs alternative actions and for each of these actions, evaluates the potential risk posed to system and human assets; and continuously updates the model, e.g., according to OODA Loop/NIST Cybersecurity Framework Core Structure/Environmental Risk Management Framework. In addition risk managers can compare the subsequent risks of each agility option and choose the best option given the goal and mission of the operation.

III. Hypothesis

We can model risk in a complex system by carrying out a dynamic risk calculation (recalculated with each change of state) using risk metrics including and beyond probabilities that characterize the system. This is the initial paper for the applied model, and is effectively a proof of concept.

IV. Methods

A. Choosing Risk Variables

The goal for any risk model is to identify the minimum number of necessary and sufficient variables that capture the risks of a system throughout any operation in order to ensure the required level of cyber security for the given system within the current environment. In order to identify such risk variables, we must first parameterize the components that comprise said system—in this case an SQL server. This top down approach identifies risk variables for each assessment goal which then provides for the identification of usable measureable metrics to quantify risk. Expert elicitation is then used to select the relevant assessment and measurement variables from the universe of risk variables (as determined by the risk parameterization effort).

Experts in the fields of computer science, computer security, intrusion detection, and risk assessment assisted the selection of the following risk variables for an SQL injection:

- Connection (C0 and C1): Is the connection to the server internal or external to the US?
- Provenance (P): What is the origin of the connection?
- Port (X): What port is being used for connection and sending query to SQL server?
- User permissions (U): What are the user permissions of the user submitting query?
- Environmental factors (E): Are there external events that may increase chance of an SQL injection?
- Potential for database access (A)—What is the potential that the query is successful and the user has access to the SQL server?

B. Issues with Chosen Parameters

The risk parameters should not be thought of as high-quality observable features of an SQL-injection attack, as would be necessary for online intrusion detection systems. The

parameters are meant to serve as indicators for higher level risk estimates that traditional security monitoring techniques fail to take into account. There are a variety of reasons that our chosen parameters are not high-fidelity measures of attack occurrences:

- Connections C0 and/or C1 must be assigned a country of origin. This could be done based upon DNS registry information, assigned IP address blocks, or AS information from BGP routing. All these methods might be subject to manipulation by very sophisticated attackers aiming to deceive the identification procedure
- Provenance P, while fairly strong if cryptographic authentication is employed between well known, trusted communicators, might be weak if a connection comes from a source with undetermined trust.
- Port numbers X, of the SQL client might not be trustworthy if the attacker controls the client computer's operating system.

However, unlike online cyber-security monitoring systems, we believe these parameters can provide invaluable data regarding ongoing attack activity when updating the dynamic risk profile of the entire system.

C. Modeling SQL Injection Attack

Using the model of an SQL injection attack, we can select from the universe of risk parameters relevant elements in order to generate a means to project and observe risk factors and understand the potential impact(s) involved. Some of the measurable variables are directly related to the technical aspects of the virtual interaction, including the ports used, source and destination, and account information. When a connection is made to the database, obvious negatives can be noted quickly, such as if a request comes from a known malicious source or if the wrong ports are used, and afford the opportunity to assign risk to the interaction. Additional evaluation should be conducted even if these initial checks pass, such as examining the permissions allowed for any involved accounts. For legitimate and illegitimate accounts alike, understanding what permissions and access they possess highlights what effects they can enact upon the database, with risk increasing as access increases. As a result, there will always be some risk value, ranging from lower value for a legitimate source and user account to high risk from malicious sources and administrative-level permissions. Other metrics touch upon the relationship of the database and accounts involved to the rest of the enterprise. For example, accounts known to be compromised or to have evidence of compromise or legitimate requests that are sent rapidly and numerously that may represent a potential DoS attempt can be flagged in the framework and, thus, contribute a higher risk than normal within the model. Historical data should also be used, with respect to both identifying characteristics of potential risks as well as evaluating the legitimacy of connections. The framework also allows for, as appropriate, the use of factors such as physical accessibility of the database, the operating conditions where the server is located, and how up-to-date the involved systems are (patches, known vulnerabilities, etc.). Each of the factors pulled from the risk framework enable a risk score to be calculated at each relevant stage in a scenario model, and in turn the score can be used to evaluate potential actions and responses to a situation.

V. Statistical Application

We applied the framework of Bayesian network analysis (BNA) to the assessment of the risk of cyber-attack in the example of server SQL injection introduced above. Figure 1 present the structure of a simple stylized model which using direct acyclic graphs (DAG) allowing at most one path connecting each pair of nodes.

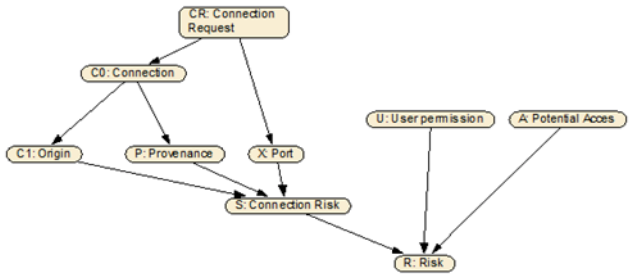

Figure 1. Direct acyclic graph (DAG) for risk assessment in the example of SQL injection attack.

The nodes represent states of the variables identifying the risk of the execution of SQL-query request. In this simple setup we identified three groups of variables attributing connection per se (Node S, characterizing risk of the connection), implemented access policies to the server (node U, specifying risk associated with the user permissions), and physical accessibility of the network (node A, describing risk of potential database access). The parent nodes U and A act as independent and direct causes for the total risk - terminal node R, attributing aggregate risk of the execution of the SQL-query. The variables in the "connection" cluster quantify risk levels associated with geographic location of the origin (code C1) and provenance (Node P) of the initiated query-request. Between nodes CR and C1 and P, there is node C0 which moderates risk of the connection: if connection is internal, the risk level of the connection is minimal, and the aggregate risk R is defined mainly by the nodes U and A. In the case when the node C0 identifies connection as an external one, the risk levels at C1 and P affect R. Node X characterize the risk associated with port (secure/insecure) through which the query was requested. A child node S (risk of the connection) is a connection risk-triggering. The idea of the triggering node S is that the risk levels associated with use of insecure port in the example, is not necessarily posing a high risk of the connection, whereas high-risk states of nodes U and A, may trigger the total risk of the connection to the high level. The nodes {C1,P,X,U,A} characterizing risk in our example have three states Low, Medium and High, i.e. {L,M,H}. The joint probability of a given state $\underline{\boldsymbol{S}}$ ={C1,P,X,U,A} is defined using Bayes rule and can be reduced to the following:

$$P(CR,C0,S,R,\underline{\boldsymbol{S}}) = P(CR)\,P(C0|CR)\,P(C1|C0)\,P(P|C0)\,P(X|CR)$$
$$*P(U)\,P(A)\,P(S|C1,P,X)\,P(R|S,U,A)$$

We compared the risk level corresponding to the consecutive states (state 1: connection accepted and state 2:

query), and take a decision whether or not the SQL-query will be executed or denied. Note, that in a trivial case when IDS flag is set positive, and the connection is denied, then no change in aggregate risk level R is being observed, and R=0. The below table demonstrates change in probability levels of low/med/high risk of execution of the query request: probability of High Risk rises from 0.06 to 0.57 due to change in the vector of variables in different states:

Risk Variables		**State S_1**	**State S_2**
Where is connection from?	C1:	**low risk, (L)**	**external to US, medium risk (M)**
Provenance	P:	**low risk, (L)**	**wired, China, high risk, (H)**
Ports	X:	**low risk, (L)**	**http port 80, medium risk (M)**
User permissions	U:	**medium risk, (M)**	**elevated privileges, medium risk, (M)**
Environmental factors:	E:	**static**	**static**
Potential for database access:	A:	**low risk, (L)**	**potential access, medium to high risk, (M)**
Probabilities	***R:***	$P(R=L\|S_1)$=.30 $P(R=M\|S_1)$=.64 $P(R=H\|S_1)$=.06	$P(R=L\|S_2)$=.17 $P(R=M\|S_2)$=.26 $P(R=H\|S_2)$=.57

Some concluding remarks: (i) conditional probabilities can be derived from real/simulated data for different types of attack; and then, (ii) this BN allows for learning: using real data with identified attacks, one can estimate probability of the attack for given state (consider attack as a latent variable); (iii) Another simplified setup, which can be modeled as a chain, is to consider variables C, P, X, U, A as independent causal factors for R. Assuming that other variables hold the same, the change in state, for example, of Origin: C will result in change in Risk: R, which, in turn, is conditioned on the preceding states of both C and R, and future state of C. The assumption made, seems to be plausible, since, for example, SQL injection attack normally uses one port 8080, and variables U and A can be considered as fairly static.

VI. Relevance to Homeland Security

Understanding cybersecurity issues, identifying needs, and establishing metrics to evaluate cyber operations has become a focal mission within the Department of Homeland Security (DHS) [9]. Cyber warfare is a very real thing, and modern society sits upon and relies on technology such that it is vital to defend and protect cyber systems [10]. This paper introduces a practical and accurate risk assessment metric by developing a representative model of a cybersecurity environment, which fits the DHS initiative of developing effective metrics stated at (https://www.dhs.gov/science-and-technology/csd-elsmu). This risk assessment enables defenders to identify and prioritize cybersecurity risks for the DHS Continuous Diagnostics and Mitigation (CDM) process (https://www.dhs.gov/cdm).

The proposed approach basically provides the ability of quantifying cybersecurity vulnerabilities of assets, potential impact of attacks, and risk in a cost-efficient and practical manner by taking into account the interactions and dependencies of cyber assets and events. The accuracy and coverage of cybersecurity risk assessment are enhanced by breaking out the various components into manageable components. In so doing, our approach meshes with the NIST Special Publication 800-30, wherein the understanding and management of risk permeates throughout technologies' uses within an organization [11]. Risk is a multifaceted front, requiring knowledge and perception across a broad space. Modeling helps determine how to mitigate the exploitation of vulnerabilities and the propagation of attack impact and malware. For example, developing an awareness of attacker strategies and adaptations can enable the creation and evolution of stronger defense measures to counter them [12]. Defining and modeling characteristics of risk in an inclusive scope as presented here leads to a more efficient utilization and allocation of cybersecurity resources, as needed in the CDM process of DHS

VII. Future Applications

We now present several applications of this model beyond SQL injection. On the server side, a similar model can be used for masquerade detection: construct a continuously-evolving Markov model [13] that holds, for each user U, the most common activities, e.g., programming, editing, file management, including command frequency, time-of-day and day-of-week patterns. Therefore, at any point, we can quantify the probabilities of future actions. When an action deviates from the predicted actions, it might indicate that an attacker M has learned U's credentials, and our model would increase the risk assessment (of a potential masquerade attack) accordingly.

For example, our recent work [14] has quantified patterns of user expertise, command frequency, time-of-day and day-of-week UNIX command usage that show when and what an user is supposed to do. If the account of a novice user, who mostly edits documents between 10a.m. and 4.p.m, is suddenly being used to issue advanced network probing commands at 2a.m., this should indicate a deviation from the user's Markov model and the risk of the attack being compromised should be raised accordingly.

In another scenario, this time on the smartphone side, consider a Homeland Security official using a smartphone in different scenarios:

(1) in a governmental agency's headquarters, connected to the enterprise network WiFi, and using an application ("app") vetted by the agency's security team. Here, the risk of attacks via the network or physical proximity (Bluetooth) is low [15].

(2) at a coffee shop, connected to the shop's WiFi, and using an app that has been downloaded in response to a browser displaying an advertisement.

Here, the risk of the app being malicious, and the risk of an attack being underway (from incoming WiFi or Bluetooth channels) is much higher. Our model will incorporate these variables and adjust the current risk profile accordingly.

VIII. Conclusion

Our research efforts are focused on detailing the technical metrics and risk metrics of the SQL injection attack to the full operation model (beyond connection accepted and query). At such time, the full operation model can be deployed using a testbed. Analysis of the data collected by the testbed will assist in the validation of the risk parameters used in the Bayesian network analysis. Such a process will allow for us to determine where risk parameters are missing and or where they may be aggregated to the minimum number of necessary and sufficient variables that capture the risks of the system required for decision-making.

Acknowledgment

All authors thank the of the Cybersecurity Collaborative Research Alliance researchers that participated in the risk parameterization process. Thank you J. Abbott, B. Bennett, N. Buchler, L.J. Camp, G. Deckard, L. Flora, D. Kelly, A. Kott, H. Marshall, L. Marvel, P. McDaniel, B. Rivera, A. Siegler, Q. Sun, A. Swami, and F. Wu.

References

[1] NIST, “Guide for conducting risk assessments,” NIST Special Publication 800-30 rev.1, 2012.

[2] NIST, “Framework for Improving Critical Infrastructure Cyber Security, ”ver 1.0, Feb 12, 2014.

[3] First.org, “CVSS Common Vulnerability Scoring System ver.3 Specification Document,” 2015. Accessed March 25, 2016, https://www.first.org/cvss/specification-document

[4] J. Aagedal, D. Braber, T. Dimitrakos, B. Gran, D. Raptis, and K. Stolen, “Model-based risk assessment to improve enterprise security,” Proceedings of the Sixth International Enterprise Distributed Object Computing Conference, 2002, p. 51–62 EDOC’02.

[5] Z. Yazar, “A qualitative risk analysis and management tool –CRAMM,” SANS Institute; 2002.

[6] C. Alberts, A. Dorofee, J. Stevens, C. Woody, “Introduction to the OCTAVE approach,” Software Engineering Institute, 2003.

[7] F. Baiardi, C. Telmon, D. Sgandurra, “Hierarchical, model-based risk management of critical infrastructures,” Reliabil Eng Syst Saf 2009; 94(9):1403–15.

[8] H. Cam and P. Mouallem, "Mission assurance policy and risk management in cybersecurity," Environment Systems and Decisions, 33.4, 2013: 500-507.

[9] Department of Homeland Security (DHS), “Cybersecurity”, 2016. Accessed 25 Mar 2016. https://www.dhs.gov/topic/cybersecurity

[10] J. Brickley, J. Cox, J. Nelson, and G. Conti, G, “The case for cyber”. Small Wars Journal, 2013. http://smallwarsjournal.com/jrnl/art/the-case-for-cyber

[11] M.L. Winterrose, K.M. Carter, N Wagner, and W.W. Streilein, “Adaptive attacker strategy development against moving target cyber defenses”, In Proceedings of MODSIM World 2014, 2014.

[12] G. Stoneburner, A. Goguen, and A. Feringa, “Risk management guide foriInformation technology systems: recommendations of the National Institute of Standards and Technology,” NIST, 2012. Accessed at http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf

[13] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, “Computer intrusion: detecting masquerades,” Statistical science, 2001, pp. 58–74.

[14] M. Gharehyazie, B. Zhou, I. Neamtiu, "Expertise and behavior of Unix command line users: an exploratory study,” Proceedings of the Ninth International Conference on Advances in Computer-Human Interactions, April 2016.

[15] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos. "Malicious android applications in the enterprise: What do they do and how do we fix it?" ICDE Workshop on Secure Data Management on Smartphones and Mobiles, April 2012.