Optimizing the RoI of Cyber Risk Mitigation

Mohammed Noraden Alsaleh, Ghaith Husari, and Ehab Al-Shaer
Department of Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC, USA
Email: {malsaleh, ghusari, ealshaer}@uncc.edu

***Abstract*—In this paper, we present a security analytics framework that augments host compliance reports with network configuration to assess the risk globally and devise cost-effective mitigation plans. We define metrics to measure the global enterprise risk based on network assets' vulnerabilities, their interdependencies, and network configurations. Our framework takes the decision burden away from administrators by automatically recommending cost-effective mitigation actions that achieve the expected return on investment (RoI). We use XCCDF, a language defined as part of the Security Content Automation Protocol (SCAP), to communicate the compliance benchmarking and scoring reports. In addition, we utilize the basic metrics defined in the standard vulnerability scoring systems, such as CVSS, to accurately assess the global risk. We formalize our proposed mitigation planning solution as a constraints satisfaction problem and we solve it using the Z3 SMT solver.**

I. Introduction

Computer networks are an integral part in many human activities, including social, medical, and financial services. The endpoint services constitute the network interface with both benign and malicious users. Hence, their proper configuration and compliance with security policy is a key property to ensure the overall network security. Traditionally, the compliance of each service is evaluated individually (i.e., in isolation from the rest of the network). We believe that the services' compliance state and the network core configuration need to be tied together in order to perform comprehensive security evaluation.

The risk in this work is imposed by the vulnerabilities and configuration weaknesses of network services. It can be further amplified or diminished based on the assets exposure to threat sources, which is controlled by the network countermeasures. A naive risk mitigation would be to fix all the vulnerabilities in a system, which will completely eliminate the risk. However, this may not be a feasible solution due to technical and financial constraints. We believe that devising cost-effective risk mitigation is a challenge due to multiple reasons. First, vulnerability fixes may not be possible because patches have not been released yet or they require high cost and overhead to implement them. Second, some vulnerable assets cannot be disabled because that may drastically affect the usability of the system and, as a result, the productivity of the employees and/or the satisfaction of the customers. Third, vulnerabilities, that may be scattered all over the network, may be highly interdependent; exploiting a vulnerability in one asset may increase the probability of exploiting another of the same or different asset. Hence, the decision of fixing a vulnerability should account for both the direct and indirect impacts of the vulnerability.

To address these challenges, we present a framework that augments the compliance reports with the network configuration in order to assess the global risk in the network and devise a cost-effective risk mitigation that assures an acceptable return on investment (RoI). We believe the integration of network configuration may introduce new feasible solutions in cases where vulnerability patching is not available, such as blocking or inspecting the traffic originating from or destined to the vulnerable services. However, applying network countermeasures, such as traffic blocking or inspection may restrict the usability of the system. Hence, our mitigation planning finds a trade-off between the financial cost, system usability, and the residual risk.

The problem of risk assessment and countermeasure selection has been tackled by researchers before. One known approach is attack graphs [6], [11], [15]. We believe that this work goes well beyond the attack graph approaches. First, we provide risk metrics that consider vulnerabilities and configuration weaknesses that facilitate the attack propagation, and allow for fine-grain mitigation actions. We do not require a complete knowledge base of attack actions and their association with vulnerabilities. Second, we utilize the standard SCAP specifications and scoring models to communicate compliance reports using the XCCDF language and calculate risk scores. Finally, our mitigation planning technique provides a fine-grain selection of mitigation actions to maximize the RoI considering constraints on budget and usability of the system. To evaluate the connectivity between threat sources and network assets, we utilize an existing network configuration verification tool [1]. Based on the compliance and connectivity reports, we formalize the mitigation planning as a constraints satisfaction problem and we solve it using the Z3 SMT solver [5]. The Z3 SMT solver provides the theoretical basis to solve linear and non-linear arithmetic constraints, which are required to model our risk metrics to drive the mitigation actions.

The rest of the paper is organized as follows. The related work is discussed in Section II. In Section III, we present the system and threat models. In Section IV, we present our risk assessment model, which is utilized in Section V to devise cost-effective risk mitigation. Section VI reports the performance evaluation. Finally, we conclude and present our future plans in Section VII.

978-3-901882-85-2

II. Related Works

In this section, we review the recent research conducted in the area of automated risk scoring and mitigation, focusing on those that employ standard specifications and scoring systems, such as XCCDF, OVAL, and CVSS in quantitative and qualitative risk assessment and mitigation models. Houmb *et al.* [7] derived a frequency and impact metrics based on the CVSS score. They have combined these new metrics to quantitatively estimate the risk level of information systems. Joh and Malaiya [9] employed a stochastic model based on CVSS metrics in a formal quantitative approach for software risk evaluation. Poolsappasit *et al.* [13] proposed a model that uses the sub-scores reported by CVSS as vulnerability exploitation probabilities that are fed to a Bayesian attack graph for calculating the global risk. None of these works integrates the host vulnerabilities with the network configuration for comprehensive analysis. Homer *et al.* [6] propose a model to aggregate vulnerability metrics, in an enterprise network, to measure the likelihood of breaches within a given network configuration through attack graphs. Homer's model and similar risk estimation techniques based on attack graphs [12], [13], [16] are driven by specific attack scenarios and they do not provide global quantitative risk scores.

In [4], Barrere *et al.* present a model for generating vulnerability remediation plans in network systems based on SAT solvers. In [8], the authors use attack graphs that augment feeds from vulnerability scanners in order to prioritize vulnerability patching. These works did not consider reconfiguration actions. Albanese *et al.* in [2] propose an approximation algorithm to automatically generate network hardening recommendations based on attack graph analysis. This solution requires complete information about the attack exploits in terms of preconditions and it is limited in the types of vulnerabilities that are covered.

In this work, we provide a risk assessment model that is utilized automated risk mitigation. We integrate the hosts' compliance reports with the global network configuration to address the vulnerability dependency based on assets' connectivity. Our mitigation actions are not limited to vulnerability patching. We provide the ability to reconfigure the network in order to reduce the risk in cases where vulnerability patching is not available.

III. System and Threat Models

We model the system as a set of interacting and interdependent services. Hence, the end-points in our model are services located in multiple connected hosts. In the following, we present the models for the compliance state of the network along with the standard scoring systems, the network configuration, and the threat model.

A. Compliance State

The compliance state of the network captures the vulnerabilities and weaknesses of all system services collected through XCCDF documents. The XCCDF specification defines a set of results that can be assigned to each rule, those are {*Pass, Fail, Error, Unknown, Notapplicable, notchecKed, notSelected, Informational,* or *fiXed*}. In the following discussion, we use the capitalized letters as abbreviations for the results.

Definition We define the compliance state of the entire network as the three-tuple $S_{comp} = (\mathbb{R}, \mathbb{V}, \mathbb{M})$, where $\mathbb{R}$ is the set of system services, $\mathbb{V}$ is the set of common vulnerabilities and configuration weaknesses, and $\mathbb{M}$ is a matrix that maps each service to its vulnerabilities. $\mathbb{M}[r, v] \in \{P, F, E, U, N, K, S, I, X\}$ for all $r \in \mathbb{R}$ and $v \in \mathbb{V}$.

B. Vulnerability Measurement and Scoring

Standard vulnerability measurement and scoring models measure the exploitability and the impact of common vulnerabilities. NIST Inter-agency Report 7502 [14] categorizes the vulnerabilities into three categories: software flaws, security configuration issues, and software feature misuse vulnerabilities. Three standard scoring specifications have been created to measure the severity of these vulnerabilities: the Common Vulnerability Scoring System (CVSS), the Common Configuration Scoring System (CCSS), and the Common Misuse Scoring System (CMSS). Each of these specifications defines a set of base scores for the availability, integrity, and confidentiality impacts, an *exploitability* sub-score, an *impact* sub-score, and a final *severity* score for each vulnerability.

Definition We model the scores of the common vulnerabilities by the matrix $\mathbb{S}$, with a row for each vulnerability and a column for each of *exploitability* sub-score (E), *impact* sub-score (I), and *integrity-impact* base score (G). $\mathbb{S}[v, sc] \in [0, 10]$ for all $v \in \mathbb{V}$ and $sc \in \{E, I, G\}$.

C. Network Configuration

The network configuration includes the network topology along with the policies of the network forwarding and filtering devices (i.e., routers, firewalls, and intrusion detection systems). A single host in the network can run multiple services, where each service is distinguished by its underlying protocol and a port number. The forwarding and filtering policies of the network core devices determine the connectivity of the system services.

To globally analyze the connectivity of the system services, we utilize a network verification tool called *ConfigChecker* [1], a Binary Decision Diagram (BDD) model checker that models the entire network configuration.

D. Threat Model

Vulnerability exploits are the sources of threats for the network services in our model. Hence, if a service has absolutely no reported vulnerabilities or configuration weaknesses, it is considered secure and it does not impose any risk on the system. Any service in the network is considered a threat source if its integrity is likely to be compromised as a result of exploiting its vulnerabilities. The number and the *integrity-impact* scores of a service's vulnerabilities determine how threatening it is to its neighbors. In addition, the threat can propagate from one service to another if the victim is

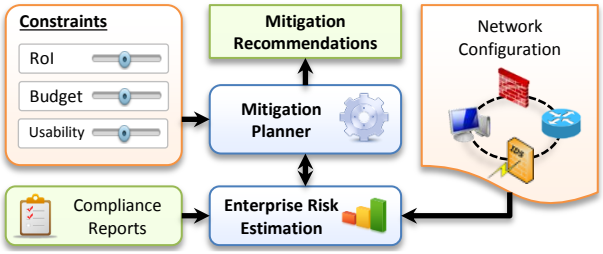

Fig. 1: Risk Estimation and Mitigation Framework.

vulnerable and is reachable from the threat source. Thus, the likelihood of threat propagation depends on the number and the *exploitability* of the victim's vulnerabilities in addition to the network configuration (i.e., connectivity).

To capture the threatening capability of a service, we define the metric *ThreatIndicator*, which intuitively measures the ability of the service to establish attacks against others. Based on our vulnerability scoring model, we use the *integrity-impact base score* in calculating the *ThreatIndicator* as it measures the ability of an exploited vulnerability to modify the system files, install root-kits, and launch attacks. The *ThreatIndicator*, denoted by ThI_r, of a particular service r is formally defined as:

$$ThI_r = \frac{\sum_{u \in V_r} \mathbb{S}[u, E] \times \mathbb{S}[u, G]}{\sum_{v \in \mathbb{V}} \mathbb{S}[v, E] \times \mathbb{S}[v, G]} \quad (1)$$

Where $V_r = \{v : v \in \mathbb{V}, \mathbb{M}[h, v] \in \{F, E\}\}$ is the set of active vulnerabilities of the service r. The *ThreatIndicator* is normalized by dividing on the weighted sum of the *integrity-impact* base scores of all the vulnerabilities identified in the system.

IV. Risk Assessment Model

For accurate risk estimation, our risk model depends on both the network configuration and the compliance state of the network as appears in Figure 1. In this section, we refine our preliminary risk metrics presented in [3] and provide the intuitive and mathematical definitions of our metrics.

A. Network Threat Resistance

The network resistance refers to the ability of the network infrastructure to prevent vulnerability exploits. It is determined by the availability of attack countermeasures in the attack paths from threat sources to vulnerable services. In this work, we consider three types of actions that can be taken by the network devices on the traffic flows: *forward*, *block*, and *inspect*. It is clear that the *forward* action imposes absolutely no threat resistance while the *block* action completely eliminates the threat. However, the effectiveness of the *inspect* action is not fixed for all threats because not all vulnerabilities can be mitigated by inspection.

In this work, we assume that the resistance of each action is given by the matrix $\mathbb{N}$, with a row for each vulnerability in $\mathbb{V}$ and a column for each action in $\mathbb{A} \in \{forward, block, inspect\}$. $\mathbb{N}[v, a] \in [0, 1]$ for all $v \in \mathbb{V}$ and $a \in \mathbb{A}$. The resistance of the *forward* action is always 0 and the resistance of the *block* action is always 1 for all vulnerabilities.

B. Threat Exposure

The *Exposure* of a service depends on three factors: (1) the *quantity* of threat sources that can reach it, (2) the *ThreatIndicator* scores of these threat sources, and (3) the network resistance in-between. To calculate the exposure, we build a reachability tree for each service in the system. A node in the reachability tree corresponds to a potential threat source that can reach the service directly or indirectly.

Since the network resistance varies for different vulnerabilities, we calculate an exposure score for each vulnerability of a service. To define the exposure formally, let TR_r be the reachability tree of the service r and let $(P_{x,r})$ represent a path in the tree from the node x to the node r. We define Exp_r^v, the *exposure* of the service r with respect to the vulnerability v, as follows:

$$Exp_r^v = \sum_{x \in TR_r} \frac{ThI_x \times (1 - Res(P_{x,r}^v))}{Len(P_{x,r})} \quad (2)$$

Where $Len(P_{x,r})$ is the length of the path from x to r. $Res(P_{x,r}^v)$ is the resistance of the action that is the most effective against vulnerability v exploits in the path between x and r.

C. Risk Estimation

We calculate a risk score for each service in the network that is proportional to its exposure, its value, and the severity of its vulnerabilities. The asset value of each service is given as part of the network configuration and it includes the value of the data stored and managed by the service. We define the total risk associated with a service, $Risk_r$, formally as:

$$Risk_r = Val_r \times \frac{\sum_{u \in V_r} (\mathbb{S}[u, E] \times \mathbb{S}[u, I] \times Exp_r^v)}{\sum_{v \in \mathbb{V}} (\mathbb{S}[v, E] \times \mathbb{S}[v, I])} \quad (3)$$

Where Val_r is the asset value of the service r. $V_r = \{v : v \in \mathbb{V}, \mathbb{M}[r, v] \in \{F, E\}\}$ is the set of active vulnerabilities of the service r. The second term of the right side of the equation is the sum of the impact sub-scores of all the vulnerabilities of a service r weighted by their exploitability sub-scores and exposure. Recall that Exp_r^v is the exposure of service r with respect to the vulnerability v and it augments the network resistance. We multiply the weighted impacts of the vulnerabilities by their exposure to reflect the effect of network resistance. We normalize this term by dividing on the weighted sum of the impact sub-scores of all the vulnerabilities identified in the system. Note that the risk is quantified as a portion of the service value. Hence, it is a monetary value that represents the expected loss due to vulnerabilities and configuration weaknesses.

V. Risk Mitigation

In this section, we present our proposed model for the automated risk mitigation based on the risk assessment model presented earlier. The mitigation planner aims at finding a

set of vulnerability *patches* and configuration *updates* that achieves the mitigation objective. We formally define the risk mitigation planning as a constraints satisfaction problem, such that the calculated risk scores drive the risk mitigation decisions.

A. Mitigation Objective

A cost-effective risk mitigation plan is the one that returns a benefit greater than the mitigation cost. Traditionally, the cost-effectiveness of a mitigation plan is evaluated by the Return on Investment (RoI) metric. In this work, we use the following popular RoI definition to measure the RoI of risk mitigation

$$RoI = \frac{Benefit - Cost}{Cost} \tag{4}$$

Where *Benefit* is the reduction in the risk that results from implementing the devised mitigation plan and the *Cost* is the total cost of the mitigation actions. The *Benefit* can be calculated as $(R_{init} - R_{residu})$, where the initial risk, R_{init}, is the risk imposed at the initial state of the network before applying any mitigation actions and R_{residu} is the residual risk that remains after implementing the recommended mitigation.

Based on this definition, the objective of risk mitigation may be expressed as a constraint on the value of *RoI*. If $RoI \leq 0$, then there is absolutely no return from the investment. Hence, the threshold of *RoI* should be a non-negative value.

B. Mitigation Actions

We consider two types of mitigation actions: vulnerability patching and network reconfiguration. For those vulnerabilities that have patches released, the planner will decide which should be *patched* in order to satisfy the mitigation objective. However, it may be impossible to achieve the mitigation objective by only considering vulnerability patching because patches may not be available or the patching cost may be infeasible. In this case, the mitigation planner decides to reconfigure the network by *blocking* or *inspecting* some flows in order to reduce the exposure of network assets, which will reduce the risk and satisfy the mitigation objective.

C. Mitigation Costs

We assume that a cost can be estimated in advance for each of the mitigation actions, including vulnerability patching on the service level and traffic flow blocking and inspection on the network level. For accurate estimation of the total mitigation cost, the costs of each mitigation action should account for the following:

- Implementation cost. This is the operational cost that the organization needs to spend in order to apply the mitigation action. This includes the consultation and labor costs, in addition to the price of new hardware or software equipments that may be required to patch service vulnerabilities or block/inspect their traffic in the network level.
- Usability cost. Other costs may be incurred due to the service interruptions and customers dissatisfaction. For example, installing a patch may require disabling a service temporarily or restarting the host machine. The usability cost is also crucial to select between blocking and inspection. Inspection should have less cost on the usability than blocking, while it may have the same effectiveness in mitigating risk at the same time.

D. Constraints

The devised mitigation plans should satisfy three types of constraints. The first type is the mitigation objective, which is a constraint on the RoI value, which is computed based on Equation 4. The total mitigation cost accumulates the costs of the recommended patching actions in addition to the costs of recommended blocking and inspection actions. The cost here includes both the implementation and usability costs. Recall that the calculation of the RoI depends on the initial risk, $Risk_{init}$, which is independent from our recommended actions. We calculate this value in advance based on the initial compliance and connectivity state of the network.

The second and third types of constraints specify thresholds on the global residual risk and an upper bound on the mitigation budget that is available for spending in risk mitigation. The user does not have to provide all these constraints. The mitigation object constraint is sufficient to devise a mitigation plan, but not necessarily the intended one as the second and third constraints might be needed in some situations to achieve more practical results.

VI. Risk Metrics Validation

Since the risk metrics are the basis for the mitigation planning, we validate that our risk metric returns valid risk estimation compared to the ground-truth. The ground-truth in this experiment is the damage caused by worms calculated through simulation. We generated a number of networks using BRITE topology generator [10] according to Waxman model. The size of the generated networks ranged between 500 and 1000 nodes. We assume that one service is running on each host and we distributed a number of vulnerabilities such that each service has at least one vulnerability. Further, in one set of networks, the majority of vulnerabilities have *Low exploitability* sub-scores while they have *High Exploitability* sub-scores in the other set. For each generated network, we first calculated the risk based on our risk model. Then, we simulated worm attacks assuming uniform and divide and conquer (D&C) scanning models and calculated the damage incurred by the worm. The damage is defined as the ratio of the assets that are compromised by the worm to the total assets in the network. A service is assumed compromised if it has a vulnerability with *high* exploitability sub-score and it can be reached by another compromised service.

The results of this experiment are depicted in Figure 2 (uniform scanning) and Figure 3 (D&C scanning). The results show the risk score calculated based on our metric against the damage measured through worm simulators. Each point in the figures correspond to a network instance. We plot a linear trendline for each experimental setting. The variance from the trend lines is expected due to the randomness of worm scanning. However, all the trend lines are non-decreasing,

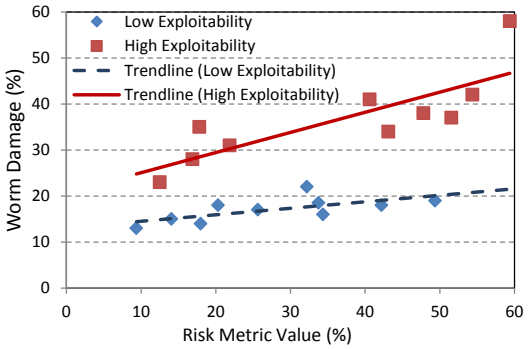

Fig. 2: Metric Validation (Uniform).

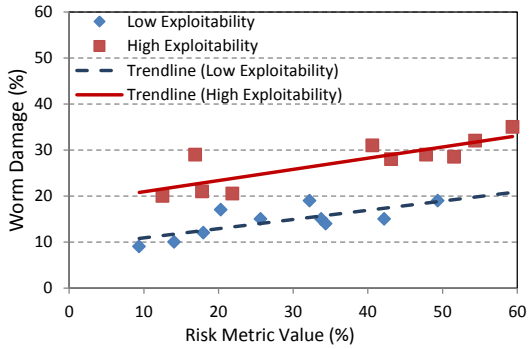

Fig. 3: Metric Validation (D&C Worm).

which means that higher risk scores correspond to higher worm damages. Hence, we believe that our risk scores valid since they are consistent with the damage sustained as a result of worm attacks.

VII. Conclusions and Future Work

In this work, we presented a risk model that measures the network global risk based on configuration and compliance reports. We propose an automated risk mitigation framework that utilizes our risk model in devising cost-effective mitigation by identifying the required vulnerability patches and network reconfigurations in order to achieve the required RoI.

We believe this work is an important step toward fast and automated security hardening utilizing the latest open standards such SCAP (specifically, XCCDF documents). In the future work, we will complete the implementation of the mitigation planning framework using the SMT solver and evaluate its performance on real and synthetic network configurations. In addition, we are planning to extend the mitigation actions to include more host- and network-based actions, such as encryption.

References

[1] E. Al-Shaer, W. Marrero, A. El-Atawy, and K. Elbadawi. Network configuration in a box: Towards end-to-end verification of network reachability and security. In *ICNP*, pages 123–132, 2009.

[2] M. Albanese, S. Jajodia, and S. Noel. Time-efficient and cost-effective network hardening using attack graphs. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pages 1–12, June 2012.

[3] M. N. Alsaleh and E. Al-Shaer. Enterprise risk assessment based on compliance reports and vulnerability scoring systems. In *Proceedings of the 2014 Workshop on Cyber Security Analytics, Intelligence and Automation*, pages 25–28. ACM, 2014.

[4] M. Barrere, R. Badonnel, and O. Festor. A sat-based autonomous strategy for security vulnerability management. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–9, May 2014.

[5] L. De Moura and N. Bjørner. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. R. Rajagopalan, and A. Singhal. Aggregating vulnerability metrics in enterprise networks using attack graphs. *Journal of Computer Security*, 21(4):561–597, 2013.

[7] S. H. Houmb, V. N. Franqueira, and E. A. Engum. Quantifying security risk level from CVSS estimates of frequency and impact. *Journal of Systems and Software*, 83(9):1622 – 1634, 2010.

[8] K. Ingols, R. Lippmann, and K. Piwowarski. Practical attack graph generation for network defense. In *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, pages 121–130, Dec 2006.

[9] H. Joh and Y. K. Malaiya. Defining and assessing quantitative security risk measures using vulnerability lifecycle and cvss metrics. In *The 2011 international conference on security and management (sam)*, 2011.

[10] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pages 346–353. IEEE, 2001.

[11] 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*, pages 336–345. ACM, 2006.

[12] X. Ou and A. Singhal. Security risk analysis of enterprise networks using attack graphs. In *Quantitative Security Risk Assessment of Enterprise Networks*, pages 13–23. Springer, 2011.

[13] N. Poolsappasit, R. Dewri, and I. Ray. Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing*, 9(1):61–74, Jan 2012.

[14] K. Scarfone and P. Mell. The common configuration scoring system (CCSS): Metrics for software security configuration vulnerabilities, December 2010.

[15] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Security and privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 273–284. IEEE, 2002.

[16] X. Yin, Y. Fang, and Y. Liu. Real-time risk assessment of network security based on attack graphs. In *2013 International Conference on Information Science and Computer Applications (ISCA 2013)*. Atlantis Press, 2013.

[17] J. H. Zeng and P. Kazemian. Mini-Stanford Backbone). https://reproducingnetworkresearch.wordpress.com/2012/07/11/atpg/.