Attack Countermeasure Tree (ACT) meets with the Split-protocol

Bharat S Rawal

Department of Information Sciences and Technology, Penn State Abington, Abington, PA 19001, USA.

ABSTRACT

In this paper, we present a novel attack tree paradigm called attack countermeasure tree (ACT) comprising an additional attack resistant element known as the Split-protocol. ACT which circumvent the fabrication and way out of a state-space representation and takes keen on account attack, as well as countermesures (in the form of detection and mitigation events). Split-protocol as an attack resistant element enhances the availability of the system during or after a security attack on the system. We compare ACT with or without Split-protocol implantation. The split-protocol concept stemmed from splitting the HTTP/TCP protocol in webserver application. An HTTP/TCP protocol is standard on a webserver can be split into two segments, and each part can be separately run on a different Web server, thus constituting dual servers. These servers communicate across a network by using inter-server messages or delegate messages.In ACT, recognition and alleviation are allowed not just at the leaf node but also at the intermediate nodes,andsimultaneouslythe state-space explosion problem is avoided in its analysis. We study the consequences of incorporating countermeasures in the ACT and Split-protocol using various case studies.

KEYWORDS

attack trees, non-state-space model, mincuts, split-protocol, and reliability.

1.INTRODUCTION

The concept of attack trees is introduced from fault trees in software safety. Fault trees are used to describe how errors disseminate in software systems, and analysis of this could be used to exam software [10] and [11]. Although fault trees are most commonly used to model how problems occur in critical systems, given the built in focus on error propagation, attack trees have a slightly dissimilar perspective. The starter of an attacker, or group of attackers, makes it believable to model extortion to an institute as well as the aspect of targeted attacks. Deliberations of likelihoods based on existing money, tools or incentive for the attacker provides a more carefully grounded duplicate of the risk level. Bruce Schneier offered the concept of attack trees as a way to model threats against computer systems[12] The basic tree model describes the Attack Tree with two different types of nodes, AND-nodes, and OR-nodes. At OR-nodes, a smallest of one sub-goal should be fulfilled to achieve the goal of the node. At AND-nodes, every sub-goal should be achieved to realize the objective of the node. The Boolean calculation could be done on the values of sub goal nodes based on a Boolean expression (AND-/OR) in the node, giving the following account in the node [9].

To evaluate the safety of the system security, modeling is used.Regular step towards security assessment is to plan and build a scalable model [12], [13] that helps to compute the security [14] in terms of important characteristics such as the damage produced by an outbreak or the gain achieved by implementing a certain set of countermeasures [15]. The simplest model issued in this context is attack tree (AT) [1]. AT utilize the genetic algorithms to find optimal countermeasure sets for the system from their AT models [6]. Though, the basic construct of AT does not take into account defense mechanisms. Roy et al. developed a novel attack tree model called attack countermeasure tree (ACT). The ACT is built on following deliberation, A) defense mechanisms not restricted to not just at the leaf B) Mincuts are used to generate and analysis of the attack and the attack countermeasure scenarios. C) Security analysis using various measures is performed in an integrated manner [1].

The remainder of this paper is organized as follows. A brief related work is presented in section II. In Section III, Split-protocol architecture presented. In Section IV, describe Split-protocol configurations V. Describes attack countermeasure tree (ACT). VI. Present quantitative and qualitative analysis. Some simulation results and the impact Split-protocol using ACT on security analysis are discussed. Finally, we conclude the paper in Section VII.

2.RELATED WORK

A security risk is being modeled using a graphical, mathematical, decision tree structure called an attack tree. There is cause to believe that attack trees are widely patronized by the intelligence community. Fault trees were invented in the early 1960s for use in the Minuteman Missile System [16]. Weiss described the threat logic trees [17]. Amoroso [18] detailed a modeling concept he called threat trees. Then, Schneider [19] (noted security expert) promoted the idea though he called it attack trees (AT). Moore et.al [20] prolonged Schneider's AT by familiarizing attack scenarios and attack profiles. Mauw et.al [21] developed an alternative formalism for AT where the goal was associated with the set of all mincuts. When applied to complex case studies, AT often became large and unwieldy. Therefore, Daley [22] proposed a layered approach to partition attack tree nodes with respect to their functionality.

To the best of our knowledge, a technique for splitting an HTTP-based TCP connection in this manner has not been proposed previously. Splitting is similar to mask failures in TCP [23] and to use the M-TCP (Migratory TCP) protocol to migrate TCP connections from one server to another [24]. However, connection migration in M-TCP does not involve splitting an HTTP request and TCP connection or operating in split mode. Moreover, the client needs to initiate the migration process in M-TCP, whereas TCP connection splitting is transparent to the client. TCP connection splitting is also different from migrating Web applications in mobile computing [25], which moves applications from one node to another on the Web by using virtualization techniques. Likewise, connection splitting is different from process migration in [26], which requires that an agent be dispatched from one system to run on another at a remote location.

3.SPLIT-PROTOCOL ARCHITECTURE

The basic split protocol architecture used for the experiments described in [2] is reproduced here for illustration as shown in Figure 1. The http request is splited at the GET command between a

CS and a DS. The CS handles the connections, and the DS handles the data transfer. Also to connections, the CS also handles the data ACKs and the connection closing. The CS has complete knowledge of the requested file, its name, size, and other attributes, but it may or may not have the file itself. However, the DS has the file and serves the data to the client.

After the GET command is received by CS, it sends an ACK to the client and also sends a delegate message DM1 to DS. The message DM1 contains the state infprmation of the request that is stored in CS in the configuration of an entry in the TCP table (known as a TCB entry). When DM1 reaches the DS, it creates its TCB entry and starts processing this request as if it was initiated itselfin the DS. When a DS sends data to the client, it uses the CS's IP. After CS hadreceivedan FIN-ACK from the client to signify connection closing, it sends another inter-server packet referred to as DM2 to DS. The DM2 received by DS will close the state of the request in DS. These DM1 and DM2 inter-server packets serve as the start and end of the request at DS. More details of the design and implementation can be found in [3].

The CS and DS architecture illustrated in Figure 1 provides a variety of delegation configurations of given requests. A request received by CS can be either processed wholly at CS or delegated to DS. That is, some requests can be handled at CS and some can be transferred to DS resulting in a variation of delegation ratio. As CS and DS are identical functional units, they can also perform any given role of CS or a DS. During these experiments, we found that when the delegation percentage is 25% in both directions (CS and DS) [4], we achieved the maximum throughput. The measurements indicated that the optimal split server performance was 1.8035 for two servers (maximum can be 2.0). Thus, the two-server system suffers only 20% capacity (10% for each server). These initial results provided us motivation to construct split protocol based servers as described below. These novel splitting techniques and associated Web server architecture introduced in this section also showed some potential in distributed computing and improving server reliability.

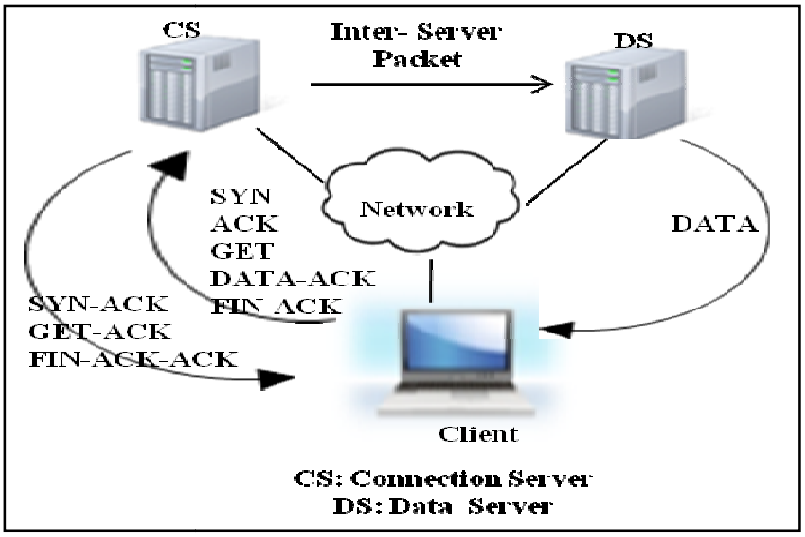

Figure 1. Split-protocol Architecture [3]

5.SPLIT CONFIGURATIONS

Configuration 2 in Fig. 2 shows a single CS with two or more DSs in the system with partial or full delegation. In partial delegation mode, clients designated as non-split request clients (NSRCs) send requests to the CS, and these requests are processed completely by the CS as usual. The connections between the NSRCs and the CSs are shown as dotted lines. With full delegation, clients designated as split-request clients (SRCs) make requests to the CS, and these requests are delegated to DSs. For full delegation, there are no NSRCs in the system. When requests are delegated to DSs, we assume that they are equally distributed among DS1, DS2 and DS3 in round-robin fashion. It is also possible to employ other distribution strategies [4].

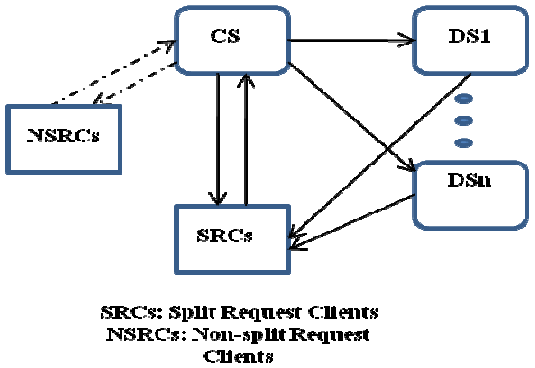

Figure 2. Split-protocol Architecture Configuration 1[4]

Figure 3 shows a general configuration for connecting one DS, one or more CSs and one or more clients (Configuration 2.). It shows two CSs and one DS with both SRCs and NSRCs. For this configuration, we used small file sizes to avoid overloading the single DS[4].

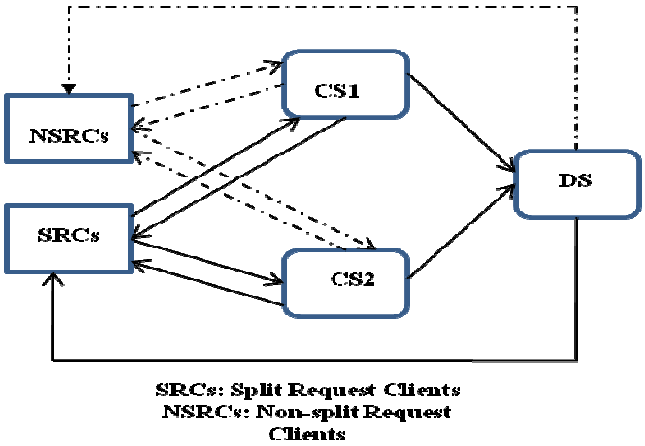

Figure 3. Split-protocol Architecture Configuration 2 [4]

Figure 4 describes a general configuration for connecting multiple CSs, and various DSs and one or more clients (Configuration 3.). Various split configurations are useful according to the need of system functionality. Example if we need a faster data transfer for alarge file, then Multi Client /Multi Server (MC/MS) configuration best choice. In MC/MS architecture, one client establishes a connection with one connection server, and concurrent data transfer dispatched from multiple data servers located on different subnets from each other. The data can be sent to the multiple clients anywhere on the network, which then reassemble or otherwise process the data. MC/MS distributes the data of large file across multiple servers without any redundancy. The separation of data transfer from a connection establishment is entirely transparent to the client.

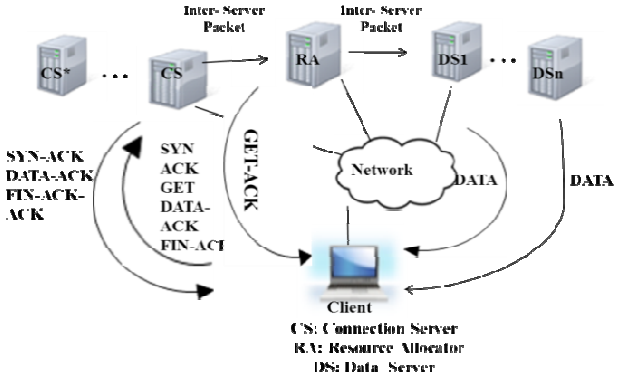

Figure 4. Split-protocol Architecture Configuration 3[5]

6.ATTACK COUNTERMEASURE TREE (ACT)

6.1Parameters

A_k is an attack event
D_k is a detection event
M_k isa mitigation event
CM_k is a countermeasure

ACT = {V, ψ , E} (V: set of all vertices in ACT, :

set of all gates in ACT, E: aset of all boundaries in ACT)

where V= {∀k, vk: vk ∈ {Aj}‖ vk ∈ {Di}‖ vk ∈

{Ml}} where A1, A2, D1, D2, M1, M2, are the events of the ACT, = {∀k, k: k ∈ {AND, OR, k-of-n gate}}, E= {∀k, ek: ek ∈ (vi, ψ_j) ‖ ek ∈ (ψ_i, ψ_j)} and X = ($x_{A1}x_{A2}$ ...$x_{D1}x_{D2}$ ...$x_{M1}x_{M2}$ ...) is a state vector for the ACT where x_{Ak}, x_{Dk}, x_{Mk} are the boolean variables associated with events A_k, D_k,M_k respectively.

Φ(X) structure function of an ACT
p_{Ak}odds of occurrence of attack event A_k

p_{Dk}odds of success of detection event D_k
p_{Mk}odds of success of mitigation event M_k
Pgoal odds of attack success at the ACT goal
p_{UD}odds of undetected attack at the ACT goal
p_{DUM}odds of detected but unmitigated attack at the ACT goal

In this subsection, the basic formalism of ACT is reproduced. In ACT, there are three different classes of events: attack events (e.g., install a keystroke logger), discovery events (e.g., sense keystroke logger) and mitigation activities (e.g., get rid of keystroke logger).

ACT for a regular server system (non-Split system)

Figure 5(a) shows simple ACT with a single-attack event. The corresponding expression for the probability of success a successful attack at goal node is shown Eq. (1).

$$P_{Goal} = P_A \quad (1)$$

In figure 5(b), one attack event and one detection mechanism are applied. The corresponding expression for the probability of an undetected attack at goal node is shown Eq.(1)..

$$P_{Goal} = P_A (1 - P_D) \quad (2)$$

For n, detection mechanisms are being used to detect one attack event equation becomes.
The corresponding P_{Goal} is:

$$P_{Goal} = P_A (1 - P_{D1}) (1 - P_{D2}) (1 - P_{D3}) \ldots (1 - PD_n) \quad (3)$$

In figure 5(c), one attack event, one detection mechanism, and mitigation events are applied. The corresponding expression for the probability of an undetected attack at goal node is shown Eq.(4).

$$P_{Goal} = P_A (1 - P_D P_M) \quad (4)$$

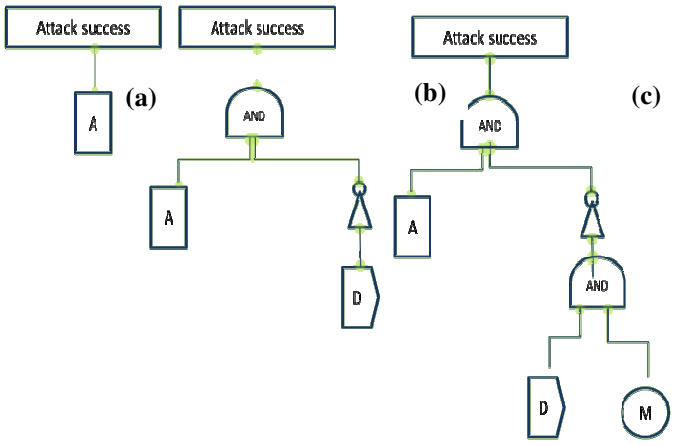

Figure 5. ACT without Split-protocol [1]

Figure 5(a), representACT with one attack event 5(b),ACT with one attack and one detection event and 5(c) represents ACT with one attack,one detection event and onemitigation event for ACT tree withoutSplit-protocol and the corresponding expression for the probability of success a successful attack at goal node is shown are equation 5, 6 and 7.

$$P_{Goal} = ½ P_A \quad (5)$$

$$P_{Goal} = ½ P_A (1- P_D) \quad (6)$$

$$P_{Goal} = ½P_A (1- P_D P_M) \quad (7)$$

Figure 6, represent the equivalent ACT for figure 5(a) and 5(b) and with Split-protocolimplementationFigure 5(a), representACT with one attack event on CS and DS 5(b),ACT with two attacks , two detections an two mitigation events. Figure 7, represents ACT with two attacks, two detectionand two mitigation eventsfor ACT tree with Split-protocol.

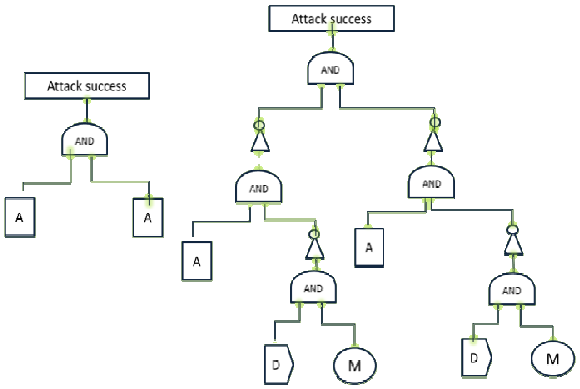

Figure 6. Equivalent ACT for figure 5(a) &5(b) with Split-protocol

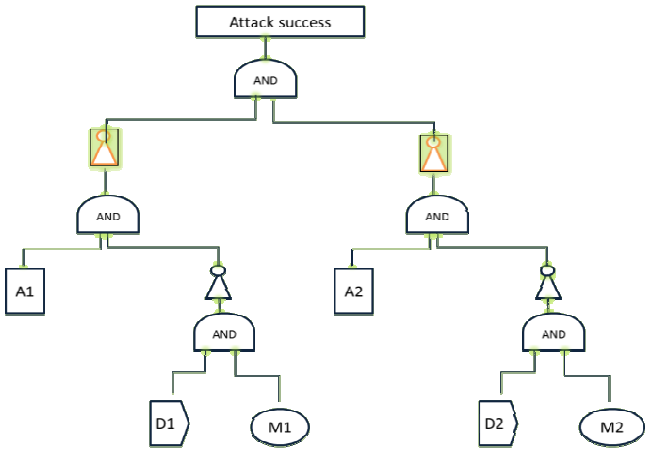

Figure 7. Equivalent ACT for figure 5(c) with Split-protocol

Figure 8. shows one attack event, n detection events, and n mitigation events. The corresponding expression 8 displays the probability of a successful attack, either attack was not detected or attack was detected but not successfully mitigated.

$$P_{Goal} = PA(1 - (1 - \prod_{i=0}^{m}(1 - PDi))) \times (1 - \prod_{i=0}^{n}(1 - PMi))) \quad (8)$$

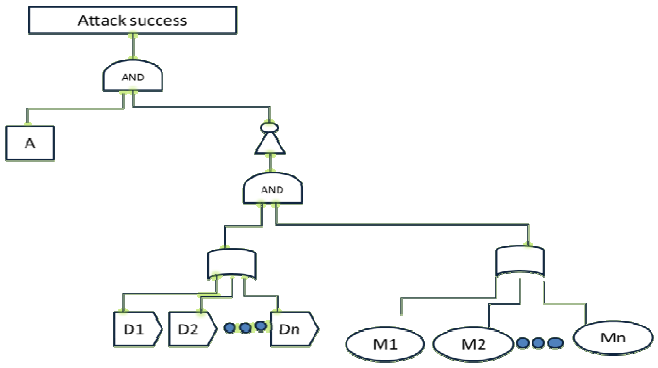

Figure 8. ACT without Split-protocol for multiple detection and mitigation events [1].

Figure 8. shows aSplit system with one attack event, n detection events, and n mitigation events. The corresponding expression 9 shows the probability of a successful attack; either attack was not detected, or attack was detected but not successfully mitigated.

$$P_{Goal= (1/2)\{} PA(1 - (1 - \prod_{i=0}^{m}(1 - PDi))) \times (1 - \prod_{i=0}^{n}(1 - PMi)))\} \quad 9$$

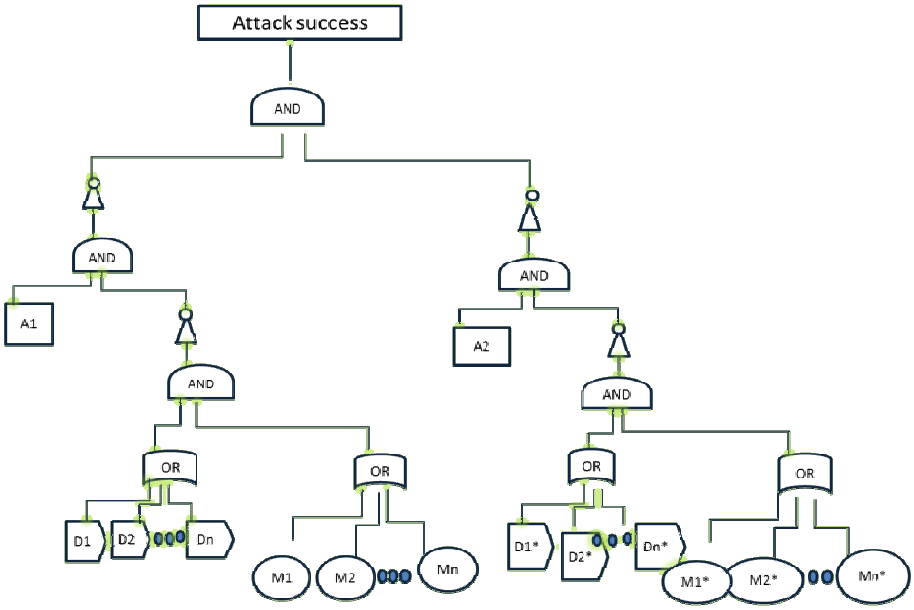

Figure 9. Equivalent ACT for figure 8 with Split-protocol

Figure 10. shows one attack event, n pairs of detection event and mitigation event. The corresponding expression is displayed by the equetion 10 for the probability of successful attack.

$$P_{Goal} PA \prod_{i=0}^{n} (1 - PDi \times PMi) 10$$

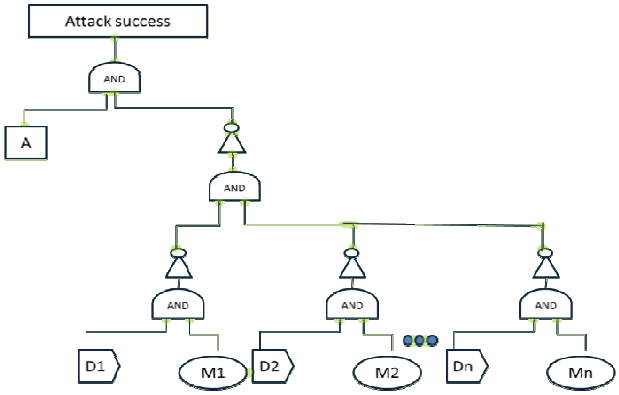

Figure 10. ACT without Split-protocol formultiple pairs of detection and mitigation events [1].

Figure 11. shows ACT structure with asplit system with one attack event, n pairs of detection event and mitigation event. The corresponding expression 11 displays the probability of successful attack.

$$P_{Goal} = ½ PA \prod_{i=0}^{n} (1 - PDi \times PMi) 11$$

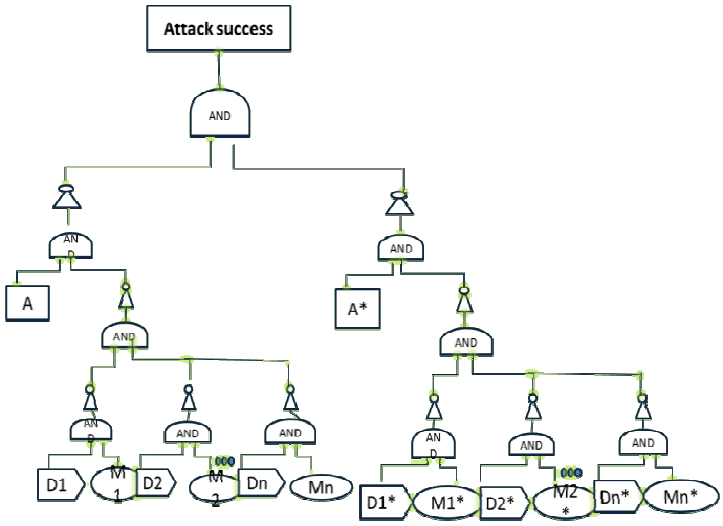

Figure 11. ACT with Split-protocol for multiple pairs of detection and mitigation events.

7.QUALITATIVE AND QUANTITATIVE ANALYSIS WITH ACT

7.1 Probabilistic Analysis

Table I. methods for probability of attack success

Gate Type	Prob. of attack success
AND gate	$\prod_{i=1}^{n} p(i)$
OR gate	$1 - \prod_{i=1}^{n} (1 - p(i))$
k/n gate*	$\sum_{J=k}^{n} \binom{n}{j} p^j * (1-p)^{n-j}$

*for identical inputs [1]

Figure 12, illustartes adirect attack tree for resetting the BGP session. In both AT and ACT, the top eventis connected with the set of all mincuts.Mincuts ofAT represent attack scenarios [1, 26] whereas those of an ACT, represent attack-countermeasure scenarios

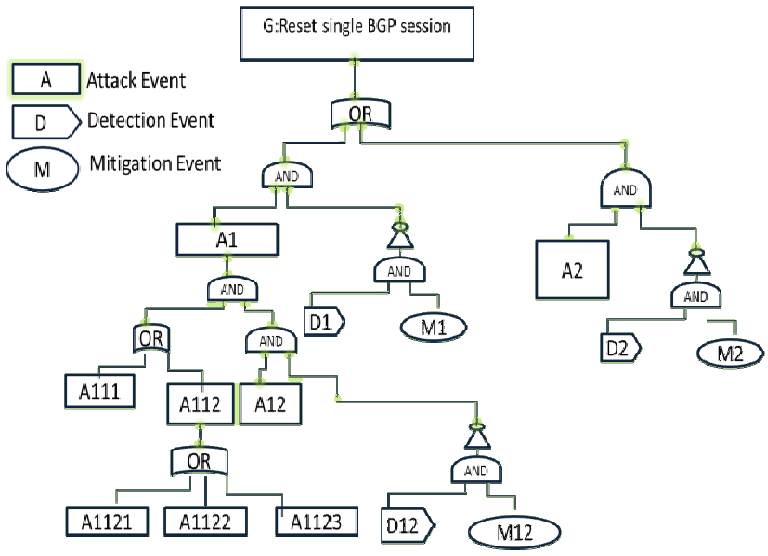

Figure 12. A straightforward attack tree for resetting the BGP session [1]

Top= $\{\{[(A_{1121}+A_{1122}+A_{1123}) + (A_{111})]\}. [(D_{12}).M_{12}). (A_{12})].(D_1.M_1)\} + \{(D_2.M_2).A_2)\}\}$

Some of the Boolean algebra laws are.

i. Commutative Law

(a) A + B = B + A
(b) A B = B A

ii. Associative Law

(a) (A + B) + C = A + (B + C)
(b) (A B) C = A (B C)

iii. Distributive Law

(a) A (B + C) = A B + A C
(b) A + (B C) = (A + B) (A + C)

iv. Identity Law

(a) A + A = A
(b) A A = A

v.Redundancy Law

(a) A + A B = A
(b) A (A + B) = A

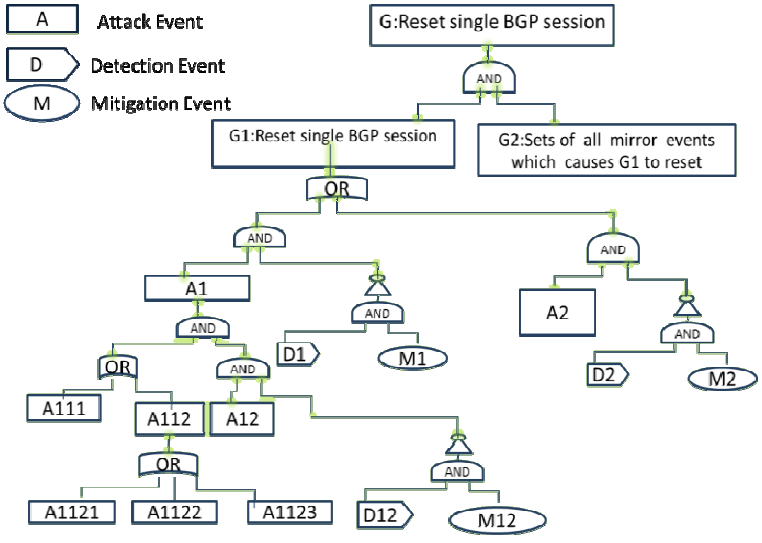

Figure 13. An attacktree with Split-protocol for resetting the BGP session

TOP = $\{\{[(A_{1121}+A_{1122}+A_{1123})+(A_{111})]\}.\{[(D_{12}).M_{12}).(A_{12})].(D_1.M_1)\}+\{(D_2.M_2).A_2)\}\}$
$*\{\{[(A_{1121}+A_{1122}+A_{1123})+(A_{111})]\}.[(D_{12}).M_{12}).(A_{12})].(D_1.M_1)\} +\{(D_2.M_2).A_2)\}\}$

vi. Applying the Identity Law

A.A = A

$= \{\{[(A_{1121}+A_{1122}+A_{1123}) + (A_{111})]\}. [(D_{12}).M_{12}). (A_{12})].(D_1.M_1)\} + \{(D_2.M_2).A_2)\}\}$

Split Protocol: Failure Rate and Survival Function

The distribution of each protocol component is the binomial distribution. With the number protocol of component n approach infinity, distribution of each component is poison distribution [4] with arrival rate, λ that is equal to the receiving rate of each component. Thus, service time or failure rate (FR) of each element is an exponential distribution [3].

1. All the n components service time X is exponentially distributed:

$$F(T) = P\{X \leq T\} = 1 - e^{-\lambda T};\ f(T) = \lambda e^{-\lambda T}$$

2. Each i^{th}component $1 \leq i \leq n$, *Failure Rate (FR) is constant.(*,$\lambda_i(t) = \lambda_i$)
3. All *n* components are identical. Then FR of each element is equal to λ ($\lambda_i = \lambda$; $1 \leq i \leq n$)
4. All n components are independent. Then

5. $P\{X_1, X2...X_n > T\} = P\{X_1 > T\}\ P\{X_2 > T\} \ldots P\{X_n > T\}$

6. The reliability of each component, $R_i(T)$ is
   a. $R_i(T) = P\{X_i > T\} = e^{-\lambda T}$
   b. $\lambda = -\ln(R_i(T))/T$
   c. T denoted system mission time.

7. System failure rate is 1 – R (T) where R (T) indicated the reliability of the whole system.

There are supposed "n" protocol components and probability of non-failure (of each component(x1, x2, x3, x4...) are exponentially distributed: For simplicity we will assume, every i^th component $1 \leq i \leq n$ probability of failure is equal to all component, i.e. Failure Rate (FR) for each component is same and ($\theta i\ (\tau) = \theta i$). For given operational time and all system, components are identical and their failure time is independent. Therefore, the reliability of, any ithcomponent (1 <i< n) reliability "Π i(τ)": "$\Pi_i(\tau)$" = $P(Xi > \tau) = e^{\theta i \tau} \Rightarrow \theta_i = -\ln(\Pi_i(\tau))/\tau$.

First, we have assumed identical components, which are identical DS in a cluster System, and they have same FR. Also, they are independent components those whose failure does not affect the performance of any other system component [2].

Reliability of Parallel identical components:

$\Pi_S = 1-(1 - \Pi_1) \times (1 - \Pi_2) \times \ldots (1 - \Pi_n)$; if the component reliabilities differ
$\Pi_S = 1-(1 - \Pi) \times (1 - \Pi) \times \ldots (1 - \Pi)$; if the component with similar reliability
$1-[1 - \Pi]^n$

For example system of two parallel components (CS, DS)

$\Pi \tau = 1-\{(1-\Pi\, 1(\tau))\, (1-\Pi\, 2(\tau))\}$
$= 1- \{(1- e^{-\theta 1\tau})\, (1 - e^{-\theta 2\tau})\}$
$= e^{-\theta 1\tau}+e^{-\theta 2\tau} - e^{-(\theta 1+\theta 2)\tau}$

And MTTF = $\mu = \int_0^{\infty} \pi\,(T) d\tau = \int_0^{\infty} (e^{-\theta 1\tau} + e^{-\theta 2\tau} - e^{-(\theta 1+\theta 2)\tau}) d\tau$
$= \frac{1}{\theta 1}+\frac{1}{\theta 2} - \frac{1}{\theta 1+\theta 2}$

And FR = θs = Density Function / Survival Function

$= - \frac{d}{dt}\pi\,(\tau) / \Pi\,(\tau)$
$= (\theta 1 e^{-\theta 1\tau} + \theta 2 e^{-\theta 2\tau} - (\theta 1 + \theta 2) e^{-(\theta 1+\theta 2)\tau} / (e^{-\theta 1\tau} + e^{-\theta 2\tau} - e^{-(\theta 1+\theta 2)\tau})$(22)

This system hazard rate $\theta s(\tau)$ can be calculated as a function of any mission time τ [3].

7.2 Mincut Analysis

According to Roy, et al, the mincuts (attack alleviationscenarios) of the ACT in Figure10 are {(A_{111}, CM_1, A_{12}, CM_{12}), (A_{1121}, CM_1, A_{12}, CM_{12}), (A_{1122}, CM_1, A_{12}, CM_{12}), (A_{1123}, CM_1, A_{12}, CM_{12}), (A_2, CM_2)} (where CM_1= (D_1M_1), CM_{12}= ($D_{12}M_{12}$), M_2= (D_2M_2))[1]. Each of the 5 mincuts corresponds to a permutation of actionseach of happening will result in attack hit at the target. For example, the mincut (A_{1122}, CM_1, A_{12}, CM_{12}) indicates that iftogether the attack events A_{1122} and A_{12} were to take place and if both the defense activity CM_1 and CM_{12} fail, attack will be successful. From the mincut (A_{1122}, CM_1, A_{12}, CM12) we also observe that the pair of attack events (A_{1122}, A_{12}) is covered by either of the countermeasures CM_1 or CM_{12} [1].

7.3 Qualitative Analysis:

Minimal cut set (mincut): a minimum combination of primary events that induce the top event Introducing Split-protocol increase length of mincut, which signals low vulnerability. The split does not introduce additional new cut sets. This implies that the inclusion of split system does not introduce additional vulnerability in the overall system. Split- protocol reduces thechance of a single point of failure. Spilt –protocol introduces n parallel components, to fail system all n component must be faulty. The splitprotocol offers inbuilt architecture reliability and fault tolerance against DoS/DDoS attack [8].

8.CONCLUSION

In this paper,we have presented the attack countermeasure trees (ACT) with implementing, a non-state-space representation that permits us to perform qualitative and probabilistic analysis of the security of the system. ACT takes into account attacks as well as countermeasures (in the form of detection mechanisms and mitigation techniques). The detection and mitigation can be placed not just at the leaf node but also at any intermediate node. When we implement the Split-protocol in thesystem, it reduces the probability of system failure by 50%. If thesystemis made of n split unit, system reliability will improve by n times. The innovative splitting system and associated Web server architecture introduced in this paper have potential applications in distributed computing and improving server reliability.

REFRENCES

[1] Roy, Arpan, Dong Seong Kim, and Kishor S. Trivedi. "Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees. “Security and Communication Networks 5.8 (2012): 929-943.

[2] Roy, Arpan, Dong Seong Kim, and Kishor S. Trivedi. "Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees." Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on. IEEE, 2012.

[3] B.Rawal, R. Karne, and A. L. Wijesinha. “Splitting HTTP Requests on Two Servers.” The Third International Conference on Communication Systems and Networks: COMPSNETS 2011, January 2011, Bangalor, India.

[4] B. Rawal, R. Karne, and A. L. Wijesinha. “Mini Web Server Clusters based on HTTP Request Splitting” HPCC 2011 : The 13th IEEE International Conference on High Performance Computing and Communications ,Sep 2, 2011- Sep 4, 2011, Banff, Canada.

[5] Rawal, Bharat, et al. "Split-Encoding: The Next Frontier Tool for Big Data."Advanced Computing, Networking and Informatics-Volume 1. Springer International Publishing, 2014. 501-510.

[6] START Understanding Series and Parallel Systems Reliability, Selected Topics in Assurance Related Technologies, volume 11, Number 5 (34)

[7] Sheldon M. Ross, A First Course In Probability, Eighth Edition

[8] Resistant Augmented Split Architecture,” IEEE 10th HONET- CNS , EMU, Famagusta, Cyprus 2013.

[9] Espedalen, Jeanne H. "Attack trees describing security in distributed internet-enabled metrology." (2007).

[10] Leveson, N. G. 1995. Safeware: System Safety and Computers. Addison-Wesley,Reading MA.

[11] Viega, J. & McGraw, G. 2002. Building Secure Software: How to Avoid Security Problems the Right Way. Addison-Wesley.

[12] Ortalo R, Deswarte Y, Kaˆaniche M. Experimenting with quantitative evaluation tools for monitoring operational security. IEEE Trans. on Software Engineering 1999; 25(5):633–650.

[13] Schneier B. Secrets and Lies: Digital Security in a Networked World. John Wiley and Sons Inc., New York, NY, USA, 2000.

[14] Trivedi KS, Kim DS, Roy A,Medhi D. Dependability and security models. Proc. DRCN, IEEE, 2009; 11– 20.

[15] Schneier B. Secrets and Lies: Digital Security in a Networked World. John Wiley and Sons Inc., New York, NY, USA, 2000.

[16] Schneier B. Secrets and Lies: Digital Security in a Networked World. John Wiley and Sons Inc., New York, NY, USA, 2000.

[17] J.D. Weiss, A System Security Engineering Process, Proceedings of the 14th National Computer Security Conference, 1991

[18] Edward G. Amoroso, Fundamentals of Computer Security Technology, pp 15-29, Prentice-Hall, 1994, ISBN01310892935
[19] B. Schneier, Attack Trees, Dr. Dobb's Journal, v. 24, n. 12, December
[20] Moore AP, Ellison RJ, Linger RC. Attack Modelingfor Information Security and Survivability.CMU/SEI-2001-TN-001 2001.
[21] Mauw S, Oostdijk M. Foundations of Attack Trees.LNCS2006; 3935:186–198
[22] Daley K, Larson R, Dawkins J. A StructuralFramework for Modeling Multi-stage Network Attacks. Proc. ICPPW, 2002; 1530–1536.
[23] D. Zagorodnov, K. Marzullo, L. Alvisi and T.C. Bressourd, "Practical and low overhead masking of failures of TCP-based servers," ACM Transactions on Computer Systems, Volume 27, Issue 2, Article 4, May 2009.
[24] K. Sultan, D. Srinivasan, D. Iyer and L. lftod. "Migratory TCP: Highly Available Internet Services using Connection Migration," Proceedings of the 22nd International Conference on Distributed Computing Systems, July 2002.
[25] G. Canfora, G. Di Santo, G. Venturi, E. Zimeo and M.V.Zito, "Migrating web application sessions in mobile computing," Proceedings of the 14th International Conference on the World Wide Web, 2005, pp. 1166-1167.
[26] T. Venton, M. Miller, R. Kalla, and A. Blanchard, "A Linux-based tool for hardware bring up, Linux development, and manufacturing," IBM Systems J., Vol. 44 (2), IBM, NY, 2005, pp. 319-330.Gan Z, Tang J, Wu P, Varadharajan V. A NovelSecurity Risk Evaluation for Information Systems.Proc. FCST, 2007; 67–73.

Author

Dr. Bharat Rawal, has conducted research in the area of computer networks, including wireless networks, Split- protocol designs and analyzes, and network performance evaluations, HPC and Network security. He was the author and co-author in several papers in networking and security area. Currently, he has focused on solving a big integers and data compression in Split-protocol infrastructure. He is now server as Assistant Professor in IST department at Penn State Abington and Visiting Assistant Professor in the department of Electrical and Computer Engineering at Duke University.