Accepting the Inevitable: Factoring the User into Home Computer Security

Malgorzata Urbanska[†] Mark Roberts[†] Indrajit Ray[†] Adele Howe[†]
Zinta Byrne[‡]
[†]Department of Computer Science & [‡]Department of Psychology
Colorado State University Fort Collins, CO 80523

ABSTRACT

Home computer users present unique challenges to computer security. A user's actions frequently affect security without the user understanding how. Moreover, whereas some home users are quite adept at protecting their machines from security threats, a vast majority are not. Current generation security tools, unfortunately, do not tailor security to the home user's needs and actions. In this work, we propose Personalized Attack Graphs (PAG) as a formal technique to model the security risks for the home computer informed by a profile of the user attributes such as preferences, threat perceptions and activities. A PAG also models the interplay between user activities and preferences, attacker strategies, and system activities within the system risk model. We develop a formal model of a user profile to personalize a single, monolithic PAG to different users, and show how to use the user profile to predict user actions.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Invasive software, Unauthorized access*

General Terms

Human Factors, Security

Keywords

security risk modeling, attack graphs, system security, attacks and defenses, security personalization

1. SECURITY ANALYSIS FOR THE HOME USER

Effective prevention, protection and mitigation of security threats on home computers requires an understanding of the user. Different user activities can impact different vulnerabilities within the home computer system in a variety of ways. How a user uses the computer and the Internet, what types of online activities that the user typically participates in, how a user perceives security risks, even how much risk a user is willing to accept to obtain online benefits – these are some factors that need to be evaluated to tailor security measures for the home computer system. (We use the term "home computer system" to refer to both the computer and the user.) These together constitute the profile for the home computer system.

We make three contributions in this work. First, we model the security threats in a home computer as they adapt to the user of the system. We refine and extend the notion of *attack graphs* to the *Personalized Attack Graph* (PAG) for this purpose. A PAG explicitly encodes, in terms of pre- and post- conditions, the dependencies between vulnerabilities existing in a *standalone system*, the system configuration (including those influenced by user preferences), and the user activities, system actions and attacker actions. Traditional attack graph/attack tree models rely solely on network connectivity between different hosts as the enabler of related vulnerability exploitation. In other words, if and only if a vulnerability exists in a host that is connected to another host, traditional models identify a related vulnerability in the second host as exploitable. A PAG, on the other hand, works on standalone systems to identify attack scenarios; network connectivity is a non-issue.

Second, we develop a formal model of the home computer user and apply this model to "prune" and personalize a single monolithic PAG to different users. (For clarity, we will use the term "*instantiated* PAG" as opposed to a "*monolithic* PAG" when a PAG is personalized). Similar systems (that is, machines having the same architecture, running the same OS with the same configuration and same set of applications) have identical monolithic PAGs to begin with. However, depending on how the user uses a system and what the user's security related characteristics are, a PAG becomes more personalized. The instantiated PAG better reflects the true security risk of the particular system.

Finally, we show how a user profile can be modeled in Bayesian Networks to predict activities that a user undertakes on the home computer. These probability values are used in the PAG to estimate probability of occurrence of different threats to the system. We run experiments on synthetic data to demonstrate, as a proof-of-concept, how the PAG model can be used to personalize security.

CODASPY'13, February 18–20, 2013, San Antonio, Texas, USA.
Copyright 2013 ACM 978-1-4503-1890-7/13/02 ...$15.00.

2. RELATED WORK

Attack trees [8, 24] have been proposed as a systematic way to model a networked system's risk to malicious attacks. They help to organize and analyze intrusion and/or misuse scenarios in a network (also called "attack scenarios") by enumerating known vulnerabilities or weak spots in the system, and capturing the cause-consequence relationships between system configuration and these vulnerabilities in the form of an And-Or tree. Attack graphs [1, 13, 25, 29] are similar data structures that have been widely used to determine if a certain goal state can be reached by an attacker who is trying to compromise a system, starting from an initial state. An exploit dependency graph [20] is an extension of the attack graph that explicitly captures the different conjunctive and disjunctive relationships between the nodes. The notion of multiple-prerequisite graphs with three different types of nodes – state nodes, prerequisite nodes, and vulnerability instance nodes – have also been adopted for security modeling [15]. Bayesian Networks, which are probabilistic graphical models (see [14] for more information), have often been used to model the states of the attack graph and encode the probabilities of the network vulnerabilities [10, 16, 22].

While each of these representations elegantly captures all possible ways by which an attacker can compromise a specific system resource, none of them are specifically geared towards the home computer system. Network connectivity between hosts is a major model element in these models. This is, however, irrelevant in a single, standalone home computer. In addition, none of these models factor in the effect of the user's actions on security. Our notion of Personalized Attack Graphs is specifically geared towards a single, standalone system, and explicitly models the interplay between user attributes, attacker strategies, and system activities within the system risk model.

3. THE PERSONALIZED ATTACK GRAPH

A PAG is a built around a set of *exploit trees.* The terminal nodes of the PAG collect together known exploits and represent different possible security compromised states for the home computer system. To help illustrate a PAG, Figure 1 shows an example of an instantiated PAG with two possible exploit trees resulting in a compromised system. We call this graph *instantiated* because it is actually a subgraph of a larger, more generic graph with nodes included only if they capture the state of a specific home computer system. We constructed the example PAG from the vulnerabilities that were identified on an actual machine running Microsoft Windows XP Professional SP3 with common configurations. Before collecting our data, the system was secured and updated. Subsequently, the machine was disconnected from the Internet, and automatic updates were disabled. After three months, the machine was plugged into the Internet and scanned using NeXpose from Rapid7 LLC [23]. NeXpose found 216 vulnerabilities during this scan. Of these, 133 were critical, 74 severe, and 9 moderate.

For this example, the PAG includes not only possible states of the system but also states resulting from user actions that can lead to information leakage. Layers in the graph indicate preconditions, but across the graph the layers are otherwise insignificant. Nodes in the graph include system states and vulnerabilities (shown as white boxes in the figure), execution state of attack actions (gray boxes), and execution states of user actions (dashed boxes). An arc in the graph is used to represent a state transition that contributes to a system compromise. The simplest transition is between two nodes (as in the box labeled E2). Conjunctive (AND) nodes (as in the box labeled E1) require all preconditions to be met for a state transition; they are indicated by multiple arcs that are incident on the node. Disjunctive (OR) branches (as in the box labeled E3) have a small circle for the choices and require only a single branch to be true. Each node has an associated probability; probabilities are presented as integers out of 100 within the node. Nodes that represent the execution of user actions are associated with a *user profile* (see Section 4 for details).

3.1 A Formal Model of the PAG

Our model of PAG is based on the formal model of attack trees presented in [8]. We propose to "personalize" attack graphs by explicitly capturing user, attacker and system actions, and tailoring the representation to specific home computer systems through pruning and analyzing the graphs.We build up a set of definitions that are increasingly inclusive to present the formal model of a PAG. We will refer to Figure 1 to exemplify each definition.

Definition 1. *A* System Attribute Template *(SAT) is a generic property of a system that can contribute towards a system compromise. It can include, but is not limited to the following:*

- *system vulnerabilities as reported in different vulnerability databases,*
- *system configuration, e.g., data availability, use of security tools, open ports, un-patched software,*
- *access privileges, e.g., root account, guest account.*

In the bottom left of Figure 1, the "VBScript MsgBox()" is an instance of a system configuration SAT, while its parent "VBScript MsgBox() CVE-2010-0483" is an instance of a system vulnerability SAT, and "User is Reading Email" is an instance of an access privilege SAT (user has privilege to read email).

Only some instances of SATs are relevant for a specific system. A successful security compromise depends on which relevant instances of SATs are present or absent (that is true or false). Instantiating an attribute template with truth values on the specific instances allows us to implicitly capture the susceptibility of the system. We define a *System Attribute* with such a concept in mind.

Definition 2. *A* System Attribute*, s_i, is a Bernoulli random variable representing the state of an instance of a System Attribute Template. It is associated with a state – True / 1 or False / 0 – and a probability value, $Pr(s_i)$, indicating the probability of the state being True / 1.*

For example, for the system in Figure 1, s_1 = "VBScript MsgBox() CVE-2010-0483" is a system attribute when associated with a truth value, signifying whether the specific vulnerability exists or not. $Pr(s_1)$ is the probability of the attribute being in state *True*.

Definition 3. *A* User Attribute Template (UAT) *is a generic property of a user that helps describe the influence of the user to home computer security. It is specified in terms of parameters that include but are not limited to:*

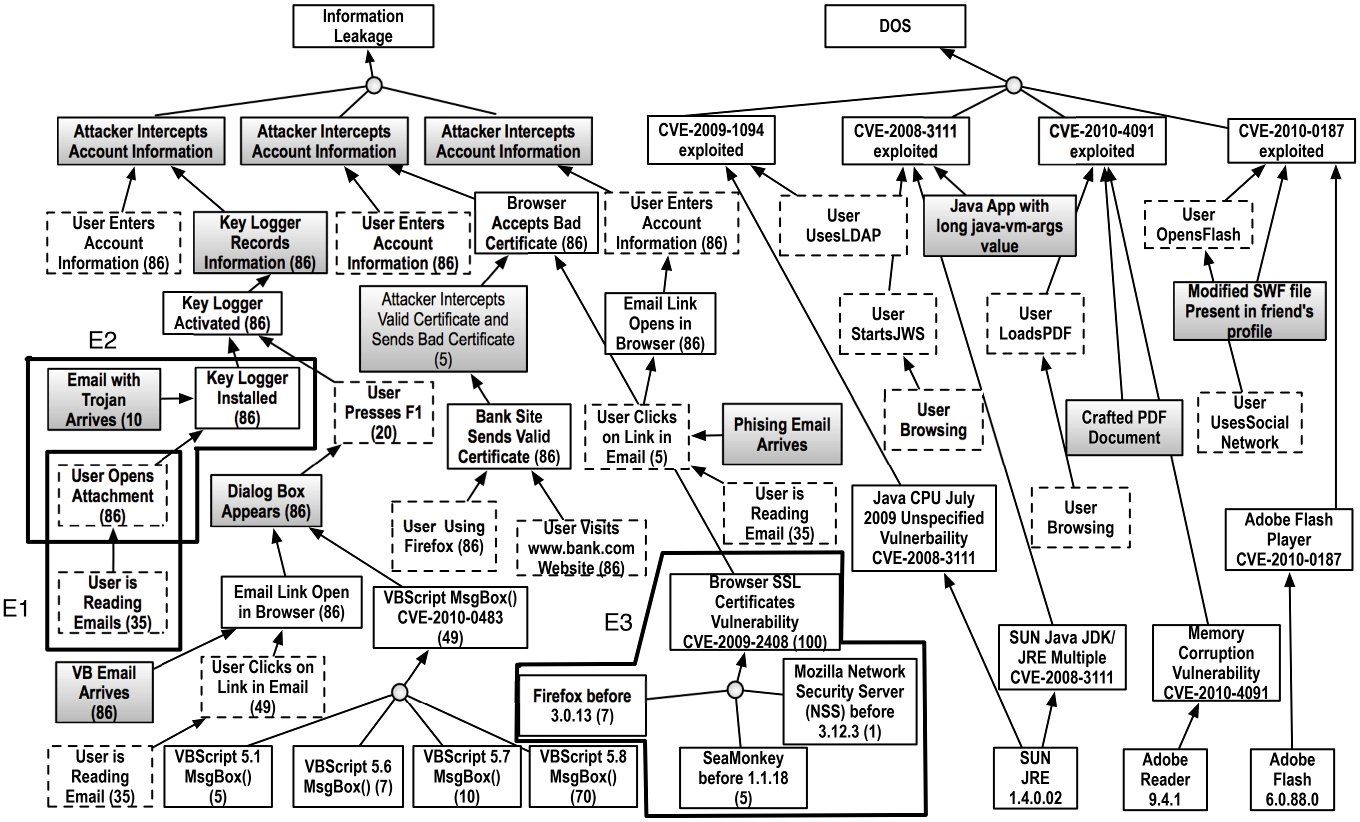
Information Leakage
DOS
Attacker Intercepts Account Information
Attacker Intercepts Account Information
Attacker Intercepts Account Information
CVE-2009-1094 exploited
CVE-2008-3111 exploited
CVE-2010-4091 exploited
CVE-2010-0187 exploited
User Enters Account Information (86)
Key Logger Records Information (86)
User Enters Account Information (86)
Browser Accepts Bad Certificate (86)
User Enters Account Information (86)
User UsesLDAP
Java App with long java-vm-args value
User OpensFlash
Key Logger Activated (86)
Attacker Intercepts Valid Certificate and Sends Bad Certificate (5)
Email Link Opens in Browser (86)
User StartsJWS
User LoadsPDF
Modified SWF file Present in friend's profile
E2
Email with Trojan Arrives (10
Key Logger Installed (86)
User Presses F1 (20)
Bank Site Sends Valid Certificate (86)
User Clicks on Link in Email (5)
Phising Email Arrives
User Browsing
Crafted PDF Document
User UsesSocial Network
User Opens Attachment (86)
Dialog Box Appears (86)
User Using Firefox (86)
User Visits www.bank.com Website (86)
User is Reading Email (35)
Java CPU July 2009 Unspecified Vulnerbaility CVE-2008-3111
User Browsing
Adobe Flash Player CVE-2010-0187
E1
User is Reading Emails (35)
Email Link Open in Browser (86)
VBScript MsgBox() CVE-2010-0483 (49)
E3
Browser SSL Certificates Vulnerability CVE-2009-2408 (100)
SUN Java JDK/ JRE Multiple CVE-2008-3111
Memory Corruption Vulnerability CVE-2010-4091
VB Email Arrives (86)
User Clicks on Link in Email (49)
Firefox before 3.0.13 (7)
Mozilla Network Security Server (NSS) before 3.12.3 (1)
User is Reading Email (35)
VBScript 5.1 MsgBox() (5)
VBScript 5.6 MsgBox() (7)
VBScript 5.7 MsgBox() (10)
VBScript 5.8 MsgBox() (70)
SeaMonkey before 1.1.18 (5)
SUN JRE 1.4.0.02
Adobe Reader 9.4.1
Adobe Flash 6.0.88.0

Figure 1: An example of a Personalized Attack Graph representing two exploits.

- *user system configuration choices, e.g., use of specific browser,*
- *user habits or activities, e.g., checking email at specific intervals, clicking indiscriminately on links,*
- *a user's sensitive information (assets) that need to be protected.*

The User Attribute Template helps capture a user's impact on security much the same way as SAT helps capture the system characteristics. Thus, the UAT contains only those parameters that are relevant for securing the home system. For example, Figure 1 shows that the bank account can be compromised by the user using the browser Firefox before version 3.0.13. Using such a browser, consequently, is an instance of the user preference UAT.

Definition 4. *A* User Attribute, u_i, *is a Bernoulli random variable representing the state of an instance of a User Attribute Template. It is associated with a state – True / 1 or False / 0 – and a probability value,* $Pr(u_i)$, *indicating the probability of the state being True.*

The notion of User Profile that we define later on (see Section 4) is different from UAT although some of the parameters of UAT can be parameters of User Profile. The difference is that the user profile concerns characteristic features of human beings that directly affect user attributes that are of type habits and activities. To illustrate, the education level of a user can be considered a parameter of user profile but is not an user attribute as defined in Definition 4.

Definition 5. *An* Attack Attribute Template (AAT) *is a generic representation of the conditions set up by an attacker (in terms of actions that the attacker can/has taken) that lead to exploitation of a vulnerability and enable a successful attack. It includes, but is not limited to*

- *performing scanning of a system*
- *installing malicious software*
- *delivering specially crafted messages*

Referring to box E1 in Figure 1, "Key Logger Installed" is an instance of installing malicious software AAT.

Definition 6. *An* Attack Attribute, a_i, *is a Bernoulli random variable representing the state of an instance of an Attack Attribute Template. It is associated with a state – True / 1 or False / 0 – and a probability value,* $Pr(a_i)$, *indicating the probability of the state being True.*

To analyze a system for potential compromise, we make a closed-world assumption for attacks. For a successful attack to take place, the corresponding attributes should have the value of true. If corresponding values are false (or are rendered false), an attack will not be successful. Consider, for example, the attacker attribute "Phishing Email Arrives" (extreme right side of PAG in Figure 1). If we expect that the attacker cannot ever successfully deliver a phishing email, this attribute will be false. Thus, we can be assured that the exploit described by this scenario will never occur. Modeling these attributes as Bernoulli random variables allows us to compute the probability of a system being compromised.

Definition 7. Atomic Exploit: *Let S be a set of system attributes, U be a set of user attributes, and A be a set of attacker attributes. Let $X = S \cup U \cup A$. Let $s_j \in S, u_k \in U, a_l \in A$ and $x_i = (s_j, u_k, a_l) \in X$. Let $\mathcal{F}$, a conditional dependency between a pair of attributes in X, be defined as $\mathcal{F} : X \times X \rightarrow [0, 1]$. Let $x_{pre}, x_{post} \in X$ be two attributes. Then $AtmExp : x_{pre} \rightarrow x_{post}$ is called an atomic exploit iff*

1. $x_{pre} \neq x_{post}$, *and*
2. *if* $x_{post} = True$ *with probability* $\mathcal{F}(x_{pre}, x_{post}) > 0$, *then* $x_{pre} = True$

The attribute x_{pre} is the pre-condition of the exploit denoted as $pre(AtmExp)$ and x_{post} the post condition denoted as $post(AtmExp)$.

An atomic exploit allows an attribute x_{post} to be transformed from x_{pre} with a some probability $\mathcal{F}(x_{pre}, x_{post})$. It is the simplest state transition that potentially leads to some security breach in the system. It can be visualized as a graph with two nodes x_{pre} and x_{post} with an arc from x_{pre} to x_{post} (rectangle E2 in Figure 1).

Definition 8. Branch-Decomposed Exploit *In order to build more complex exploits, let $BranchExp = \{x_{pre_1}, \ldots, x_{pre_k}, x_{post}\} \subseteq X$ be a set of attributes such that if $x_{post} = True$ with some non-zero probability,*

1. $\forall i, x_{pre_i} = True$, *or*
2. $\exists i, x_{pre_i} = True$

then BranchExp is called a Branch-Decomposed Exploit. *Case (1) is called an* and-decomposition *and has the precondition: $pre(BranchExp) = \{x_{pre_1}, \ldots, x_{pre_k}\}$. Case (2) is called an* or-decomposition *and has the precondition: $pre(BranchExp) = x_{pre_i}, \forall i = 1, \ldots k$. The postcondition of both cases is: $post(BranchExp) = x_{post}$.*

A branch-decomposed exploit that is an and-decomposition is visually represented as a set of nodes $x_{pre_1}, \ldots, x_{pre_k}, x_{post}$ with arcs from x_{pre_i} to x_{post}. For an or-decomposition, the arcs from the x_{pre_i}'s are incident to a small circle from which an arc is incident on x_{post}. In Figure 1, an example of an or-decomposed branch exploit is the set of attributes enclosed by E3, while an and-decomposed exploit is the set of attributes enclosed by E1. We will call a set E of attributes an exploit, if either E is an atomic exploit or a branch-decomposed exploit.

Definition 9. Exploit Tree – *Let X be a set of attributes and E be either an atomic exploit or a branch-decomposed exploit. An* Exploit Tree *is a tuple $ET = \langle \epsilon_{root}, \mathcal{E}, \mathcal{P} \rangle$, where:*

1. $\mathcal{E} = \{E_1, E_2, \ldots .E_n\}$ *is a set of exploits defined over the set of attributes X.*
    - $x \in X \leftrightarrow \exists E_i \mid x \in E_i$
    - *If* $x \in E_i, x \neq \epsilon_{root} | x = post(E_i)$ *then* $\exists E_j, j \neq i \mid x \in pre(E_j) \wedge \nexists E_k, k \neq j \neq i | x \in pre(E_k)$
2. $\epsilon_{root} \in X$ *is a goal attribute that the attacker wants to be true such that* $\nexists E_i \in \mathcal{E} \mid epsilon_{root} \in pre(E_i)$
3. $\mathcal{P}$ *is a set of estimated probability distributions. The elements of $\mathcal{P}$ are all the $Pr(x)$'s associated with attributes x's in ET.*

By the above definition, any proper subtree of an exploit tree is also an exploit tree. An exploit tree is characterized more by the goal attribute, ϵ_{root}, that the attacker wants to be true (as perceived by a security analyst), rather than the other attributes and the associated state transitions.

A home computer system may have only one exploit tree. However, more often than not, several goal attributes will be "of interest" to the attacker, requiring several exploit trees. Moreover, these exploit trees can be related to one another in the sense that rendering a goal attribute to be true in one tree leads to an attribute in another tree being true. To model this scenario we introduce the notion of Personalized Attack Graph.

Definition 10. *A* Personalized Attack Graph *is a set of related exploit trees. It is represented by a tuple $PAG = \langle \mathcal{G}_1, \mathcal{G}_2, \mathcal{V}_1, \mathcal{V}_2 \rangle$, where:*

1. $\mathcal{G}_1 = \mathcal{E}_p, \ldots, \mathcal{E}_q$ *and* $\mathcal{G}_2 = \mathcal{E}_m, \ldots, \mathcal{E}_n$ *are* disjoint *sets of exploit trees such that* $\mathcal{E}_i \in \mathcal{G}_1 \leftrightarrow \mathcal{E}_i \notin \mathcal{G}_2$.
2. *Let $\mathcal{V}_1$ be the set of goal attributes of exploit trees in $\mathcal{G}_1$ and $\mathcal{V}_2$ the set of goal attributes in $\mathcal{G}_2$ such that $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$, the set of all attributes in $\mathcal{G}_1$ and $\mathcal{G}_2$ and $\mathcal{V}_1 \cap \mathcal{V}_2 = \phi$. A goal attribute $v_i \in \mathcal{V}_1$ iff $\nexists x_k \in \mathcal{E}_d \in \mathcal{G}_2 \mid pre(x_k) = v_i$. A goal attribute $v_j \in \mathcal{V}_2$ iff $\exists x_l \in \mathcal{E}_b \in \mathcal{G}_1 \mid pre(x_l) = v_j$.*

Essentially, a PAG is a graph constructed out of the exploit trees, E_i's, present in a home system. The set of exploit trees is partitioned into two sets $\mathcal{G}_1$ and $\mathcal{G}_2$. The set $\mathcal{G}_1$ consists of all those exploit trees that have those goal attributes which are goals in themselves and do not lead to different attributes in other exploit trees being set to true; these goal attributes are not pre-conditions of any attribute of any exploit tree. These goal attributes are the *terminal nodes* of the PAG. The set $\mathcal{G}_2$, on the other hand, consists of all those exploit trees that have goal attributes that, if set to true, can lead further to attributes in other exploit trees to be set to true as well; these goal attributes are pre-conditions of some other attributes. To prevent cycles, we explicitly forbid the goal attributes in $\mathcal{V}_2$ to be pre-conditions of attributes of exploit trees in $\mathcal{G}_2$. A cycle in a PAG (if it was allowed to exist) would contain a sequence of goal attributes of the form $v_1, v_2, \ldots, v_n, v_1$ such that $v_1 \in pre(x_a) \in pre(x_b), \ldots, \in pre(v_2), \in pre(x_k) \ldots \in pre(v_3) \ldots \in pre(v_n) \ldots \in pre(v_1)$. By following this sequence the attacker sets to true what has already been set to true, and is essentially of no value to further risk analysis; this follows from the monotonicity property [1].

4. MODELING THE HOME USER

The most important characteristic of the PAG that differentiates it from other threat modeling paradigms is its ability to capture the contribution of the human user to system security. What activities a user performs on their computer, how they perceive the risks and what benefits they think accrue from their activities impacts the security threat to the home computer. Thus, the PAG requires several types of information about the user and leverages that information to identify the vulnerabilities that are most severe or likely for a specific home computer system.

User information is represented as User Attributes in the PAG. User attributes include user actions, preferences, and assets. At this time, user actions, user preferences and assets are manually incorporated into a general version of the PAG. For example, Figure 1 includes "User is Reading Emails" and "User Opens Attachment" as user actions, "Firefox before 3.0.13" and "SUN JRE 1.4.0.02" as user preferences and "User Enters Account Information" as assets. The user actions and assets are identified from the vulnerability descriptions.

User preferences are represented as the probabilities associated with the User Attributes. These probabilities are critical to determining what poses the strongest threats to a home computer system. To compute these probabilities, we develop a model, represented as a Bayesian Network, that relates characteristics of home computer users to preferences and behavioral tendencies.

Our Bayesian network model has been significantly influenced by two prior models of human behavior in computer security: Ng, Kankanhalli and Xu's [19] and Claar's [6]. Ng et al. adapts the Health Belief Model (HBM) [27, 26], to predict computer security behavior. Their model includes six primary factors and one moderating factor that can predict a person's decisions about security. We use these primary factors and the moderating factor in our model. The moderating factor is the output node of the Bayesian Network and called the *target node*. The six primary factors we use are:

- *Perceived severity* captures a user's belief in the seriousness of a possible security violation from a specific activity.
- *Perceived benefits* captures a user's perception of effectiveness or benefit of adopting an action or a specific preference.
- *Perceived barriers* describes a user's perception of cost or disadvantages associated with specific actions or preferences.
- *Risk tolerance* describes an individual's ability to handle or undertake different degrees of potentially harmful activities. (It is intended to account for the result of studies that have shown that users are willing to accept risk if the potential benefit is viewed as more important [21, 12]. Ng et al. [19] calls this factor as *Perceived susceptibilty* while Claar [6] terms this as *Perceived vulnerability.* We believe that the term Risk Tolerance is more appropriate, keeping the nature of the factor in mind.)
- *Self-efficacy* captures a user's belief that he or she is capable of taking specific action. (It has been observed to be an important factor in several home user studies [2, 3, 18].)
- *Cues to action* captures the user's motivation to cause a change in behavior. (Some studies [11, 30, 2] have shown importance of cues to action that encourage users to undertake certain activities.)

Several studies [31, 4, 11] have also shown that a user's *prior knowledge* and *prior experience* with computers and security may affect how the user perceives and acts on security threats. We include this as two other factors that can predict a user's decisions about security.

As in Claar's work [6], our model includes demographic factors (*gender*, *age*, *socio-economic*, and *education*) as predictors of user's decisions about security. Inclusion of these demographic factors is further supported by other studies: Friedman et al.[9] (user community), Szewczyk et al.[30] (socio-economic factors) [9, 4, 18, 11] (age and gender).

4.1 Predicting user actions

The user attribute probabilities for a specific user is calculated from the values for the 12 factors italicized above. The user will be led through a series of questions and answers that results in specific values (in the range [0,1]) for a given user. The demographic factors, prior knowledge and prior experience are straightforward characteristics of the user and are easy to evaluate. The six factors – risk tolerance, perceived severity, perceived benefits, perceived barriers, self efficacy, and cues to action – on the other hand, are more difficult to assess; we expect to determine how best to do so from the human subject studies that we are currently pursuing that relate the users' perceptions to their actions in security settings.

To predict user actions we use the values of these factors as inputs to Bayesian Networks. For every user attribute (see Definition 4) that is relevant for a given user, we build a Bayesian Network. This collection of BNs is called a *Bayesian User Profile* (BUP). Each of the networks (or subnets) in the BUP provides the posterior probability value for a specific user attribute in an instantiated PAG when the subnet is populated with evidence values corresponding to the specific user. To estimate the prior probabilities we rely on the previously mentioned user studies (such as [30]).

The dependent variable of the Bayesian Network gives the probability for a given user attribute. We call this output the *target node* of the subnet.

To make concrete how users can lead to different BUP target values, we present three hypothetical user profiles and set the value of the independent variables for those users in the corresponding BUPs. UserA is a retired person who was recently given a Windows XP machine that runs Internet explorer (IE). UserA is familiar with the inventory computer system from a recent job but is a new user of the Internet and email. UserB is a 20-year-old college student with a portable laptop running Windows7. UserB has used computers since kindergarten, is very confident when using them, and insists on browsing the Internet with Firefox. UserB automatically accepts any dialog that the browser displays. UserC is a 22-year-old college student who happens to be a computer science major. UserC is aware of security concerns and is diligent about installing updates and being observant of what her computer downloads. As an example, Table 1a shows the calculated values from the BUP for the three users and a set of the user attributes from the right side of the PAG in Figure 1.

Let us consider the example configuration for UserA more specifically. According to the description given above, we can assume that UserA is unlikely to use any social network or read PDF files. However, there is still a nonzero probability that user can take these actions. For this reason, we assigned probability equal to 0.015 to these attributes. For this example, let us further assume that UserA is a highly educated female at age 60 with very good socioeconomic standing, but she has very low experience in Internet. The example question can be: what is the likelihood of her opening a flash file (attribute "OpensFlash")? She has a harm avoidance personality (Risk Tolerance on low level) and she perceives lower severity for taking this action because she does not know much about security concerns in Internet.

Table 1: Results of experiments to determine which system attributes are most relevant for DOS exploit

(a) User attribute probabilities of engaging in specific activities.

User Attribute	UserA	UserB	UserC
UsesSocialNetwork	0.015	0.974	0.961
OpensFlash	0.914	0.972	0.959
UsesLDAP	0.015	0.949	0.015
Browsing	0.922	0.974	0.971
StartsJWS	0.929	0.966	0.015
LoadsPDF	0.015	0.974	0.963

(b) Probability estimates of vulnerabilities from CVSS scores.

	CVSS		
Vulnerability	AC	Base	p(e)
CVE-2009-1094	0.71	10	0.99968
CVE-2010-4091	0.61	9.3	0.85888
CVE-2010-0187	0.61	4.3	0.85888
CVE-2008-3111	0.71	10	0.99968

(c) How individual vulnerability probabilities change w.r.t. user profiles.

	User Profile		
Exploit Name	UserA	UserB	UserC
DOS Exploit	0.121	0.542	0.169
CVE-2009-1094	0.007	**0.474**	0.007
CVE-2010-0187	0.001	0.075	0.073
CVE-2008-3111	**0.12**	0.132	0.002
CVE-2010-4091	0.002	0.106	**0.105**

This flash file also contains interesting information about upcoming political event; therefore the perceived benefit is at a medium level. But because she does not know much about the Internet, she is not sure how to find and run this file (Perceived Barriers). In addition, she does not feel comfortable and is afraid to take this action (Self Efficacy is at a low level). Nevertheless, because her good friend recommended that she open the file, the Cues to Action are at a medium level.

For each of the user attributes, similar scenarios are considered and appropriate subnet configuration is assigned. The effect of these characteristics is translated to probability values for the user attributes by the corresponding Bayesian networks for the other values presented in Table 1a.

Note that, since there can be multiple occurrences of the same user attribute in an *instantiated* PAG, there can be a one-to-many relationship between a target node of a subnet and user attributes in the instantiated PAG. Additionally, not all subnets in an instantiated BUP will be relevant to an instantiated PAG. For example, if we are interested in determining the probability of the user logging into a social network portal in the PAG in Figure 1, then we will be interested in determining the probability of the user attribute "UsesSocialNetwork"; we will not be interested in the user attribute "UsesEmail."

5. USER-CENTRIC SECURITY ANALYSIS

Our goal in this work is to develop a methodology for personalizing security by matching home computer security to each user. To achieve this, we instantiate a PAG and use the BUP to update probabilities within the PAG. Owing to the dynamic nature of the threat model, it appears appropriate to use a Bayesian Network to calculate the changing probability values in the PAG. We thus implement a PAG as a Bayesian Network for analysis purposes. To avoid confusion, we would like to re-iterate that Bayesian Networks are used in two different contexts in this work – one to model the likelihood that the user engages in specific activities (a value we read from the target nodes of each subnet of the BUP) and the other to facilitate automated analysis of the PAG (that we discuss next).

In the experiments below, we focus on the Denial of Service (DOS) exploit tree of the PAG presented in Figure 1 (see the right most exploit tree, starting under node "DOS"). We choose this subgraph because it contains sufficient numbers of system, user, and attacker attributes to support our analysis: four CVE vulnerabilities, six user actions, and three system attributes that rely on the system configuration. We assess the impact that the user profile has on system compromise probability estimates in the PAG.

We implemented both the BUP and the DOS subtree of the PAG with the Bayesian Network package called SMILE [7]. Each system, user, or attack attribute (see Definitions 2, 4 and 6) is a node in the Bayesian Network with arcs connecting to its preconditions and/or postconditions.

Each node has single prior probability, p, associated with it (see Section 3.1). Leaf nodes have a probability of existence. All other nodes have a probability of being exploited. The probability of vulnerability nodes (beginning with CVE in Figure 1) are calculated from equations in [22], which apply values from the Common Vulnerability Scoring System (CVSS) [17]. Table 1b presents the set of vulnerabilities we use in this paper with associated vectors from the Base Metric Group of CVSS. The columns include the Access Vector (AV), the Access Complexity (AC), the Authentication Score (AU), and the Base Score from CVSS and the corresponding calculated probabilities. For all vulnerabilities, the AV score was 1 and the AU score was 0.704. The attack attribute nodes of the PAG have probabilities estimated from expert knowledge. Similarly, expert knowledge is used to compute probability values for system attributes that are not vulnerabilities.

User attributes in the PAG that are connected with target nodes from a BUP retrieve their probabilities from those targets. Each relevant subnet of a BUP provides a probability value for a target node that attaches to a user attribute node in the PAG (see Definition 4). Each of the user attribute nodes in a PAG is connected to some target node in the user profile (see Section 4.1), as it is an execution state that is directed by user activity.

We begin by examining which system attributes (see Definition 2) are most relevant to the DOS exploit. To perform the experiment, we apply the three user profiles and observe the probabilities of the four CVE vulnerability related systems attributes in the DOS exploit (the Or-decompositions of the DOS exploit in Figure 1). In this experiment, we do not adjust any evidence related to system or attack attributes, nor do we set specific evidence. So our results are based only on the estimated probabilities that came from the original PAG for system and attack nodes.

As can be seen in Table 1c, the probabilities of the vulnerabilities do change based on the user profile. In this case, CVE-2008-3111 (a Java vulnerability) is more likely for UserA, CVE-2009-1094 (a LDAP vulnerability) is more likely for UserB, and CVE-2010-4091 (an Acrobat Reader vulnerability) is more likely for UserC.

We next examine how changes in the presence of evidence impact the final probability of the DOS exploit. We manually set the evidence of specific nodes to *NotExists* or *Exists* as appropriate, which effectively sets the values to 0.05 and

0.95 respectively. We then update the BN and read the value of the exploit node.

Table 2 shows the probability of the DOS exploit before any changes are made (the baseline) and after a set of user action changes are applied. There is little change to the DOS exploit occurring if the user is using (or likely to use) Social Media, Starts a Java Web Start Application, or just Browsing the Internet. However, it is also clear that the probability of exploit increases greatly for UserA and UserC if a LDAP connection is started. Conversely, if UserB were to get rid of the LDAP connection, then the probability of exploit drops dramatically. These results suggest that the DOS exploit is sensitive to the user's actions, and the probability can jump dramatically if the user takes the worst-case action. The results also suggest that understanding the user's current actions or likely actions can contribute to selecting which action(s) are important to observe.

6. SUMMARY AND FUTURE WORK

Studies have repeatedly shown that routine computer activities such as checking emails, web browsing, and filling out on-line forms, deliver the vast majority of security threats to the home computer [5, 28]. This happens most of the time because many users do not fully understand how their activities impact security. To design effective security tools for the home user, we need to determine which user actions might impact security. Towards this end, we investigate the problem of personalizing security risk analysis and matching home computer security to each user's needs. We extend the classical attack graph model for security risk analysis to include the different actions that a user can take and the implications of those actions on home computer security. We call this model the Personalized Attack Graph model. A PAG captures the interplay between user actions, attacker strategies, and system activities. We demonstrate how a PAG can be instantiated as a Bayesian Network to rank potential security risks. This, in turn, allows us to propose suitable interventions for activities that can be potential risks. Toward generalization to different users, we formalize a model of a user and apply this model to personalize a single, monolithic graph to different users.

Our long term goal is to develop a semi-autonomous, intelligent and personalized approach to computer security that leverages psychological studies of what users want/need, what security and privacy risks are imminent based on the status of the system and the user's actions, and what interventions will be most effective. The PAG is the core model for capturing the relationships between different user actions, system states and vulnerabilities. The PAG will be the core representation for an on-line security agent that will monitor user actions deemed to be critical to security and suggest actions that will keep the system safe.

However, the PAG does not enumerate the details needed to either analyze what can happen or identify intervention plans to protect the user. As the next step, we plan to use AI planning techniques to determine what activities need to be monitored and what actions can be taken to prevent security breaches while taking into account the user's desired level of security risk and utility for performing a specific action.

One of the components of the PAG model is estimated probability values for different attributes. While we have a sound basis for estimating probability values for system attributes and we have discussed how probability values for user attributes can be determined via the creation of user profiles, we are still in the process of determining a good way to estimate attack attribute probabilities. Currently, we are basing this on expert opinion. Future work includes subject studies to assess perceptions of risk and investigating more informed ways of estimating these probability values.

7. ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 0905232. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] Ammann, P., Wijesekera, D., and Kaushik, S. Scalable, graph-based network vulnerability analysis. In *Proc. of the 9th ACM Conference on Computer and Communications Security* (Washington, DC, 2002).

[2] Aytes, K., and Connolly, T. Computer security and risky computing practices: A rational choice perspective. *Journal of Organizational and End User Computing 16*, 3 (July-Sept 2004), 22–40.

[3] Bryant, P., Furnell, S., and Phippen, A. Improving protection and security awareness amongst home users. In *Advances in Networks, Computing and Communications 4*, P. S. Dowland and S. M. Furnell, Eds. University of Plymouth, April 2008.

[4] Byrne, Z., Weidert, J., Liff, J., Horvath, M., Smith, C., Howe, A., and Ray, I. Perceptions of internet threats: Behavioral intent to click again. In *Proc. of the Society for Industrial and Organizational Psychology Conference* (San Diego, CA, April 2012).

[5] CERT Coordination Center, Software Engineering Institute. Home computer security. Available at `http://www.cert.org/homeusers/HomeComputerSecurity/`, 2008.

[6] Claar, C. L. *The Adoption of Computer Security: An Analysis of Home Personal Computer User Behavior Using the Health Belief Model.* PhD thesis, Utah State University, 2011.

[7] Decision Systems Labratory, University of Pittsburgh. GeNIe and SMILE Homepage. Available from `http://genie.sis.pitt.edu/`.

[8] Dewri, R., Poolsappasit, N., Ray, I., and Whitley, D. Optimal security hardening using multi-objective optimization on attack tree models of networks. In *Proc. of the 14th ACM Conference on Computer and Communications Security* (Alexandria, VA, 2007).

[9] Friedman, B., Hurley, D., Howe, D., Nissenbaum, H., and Felten, E. Users' conceptions of risks and harms on the web: A comparative study. In *Extended Abstracts of the Conference on Human Factors in Computing Systems* (Minneapolis, MN, USA, 2002).

[10] Frigault, M., and Wang, L. Measuring network security using bayesian network-based attack graphs. In *Proc. of the 32nd Annual IEEE International*

Table 2: Results showing how evidence set at specific user nodes (top) and user/system/attack nodes (bottom) impacts the DOS Exploit probability.

	UserA		UserB		UserB	
Nodes Existence	DOS	Change	DOS	Change	DOS	Change
Baseline	0.121		0.542		0.169	
¬ UsesSocialNetwork	0.120	-0.001	0.509	-0.033	0.107	-0.062
UsesSocialNetwork	0.181	0.059	0.543	0.001	0.172	0.003
OpensFlash + UsesSocialNetwork	0.186	0.065	0.544	0.002	0.174	0.005
¬ UsesLDAP	0.116	-0.005	0.268	**-0.275**	0.163	-0.006
UsesLDAP	0.476	**0.354**	0.557	0.015	0.555	**0.386**
¬ Browsing	0.008	-0.113	0.488	-0.054	0.076	-0.093
Browsing	0.131	0.010	0.544	0.001	0.172	0.003
¬ LoadsPDF	0.120	-0.001	0.491	-0.051	0.077	-0.092
LoadsPDF	0.221	0.100	0.545	0.003	0.175	0.006
Browsing + LoadsPDF	0.221	0.100	0.545	0.003	0.175	0.006

Computer Software and Applications Conference (Washington, DC, 2008).

[11] Furnell, S., Bryant, P., and Phippen, A. Assessing the security perceptions of personal internet users. *Computers & Security 26*, 5 (August 2007), 410–417.

[12] Good, N., Dhamija, R., Grossklags, J., Thaw, D., Aronowitz, S., Mulligan, D., and Konstan, J. Stopping spyware at the gate: A user study of privacy, notice and spyware. In *Proc. of the 2005 symposium on Usable privacy and security* (Pittsburgh, PA, USA, 2005).

[13] Jha, S., Sheyner, O., and Wing, J. M. Two formal analysis of attack graphs. In *Proc. of the 15th IEEE Computer Security Foundations Workshop* (Cape Breton, Nova Scotia, Canada, 2002).

[14] Korb, K. B., and Nicholson, A. E. *Bayesian artificial intelligence.* CRC Press, Boca Raton, FL, 2011.

[15] Lippman, R., Ingols, K., Scott, C., Piwowarski, K., Kratkiewicz, K., and Cunningham, R. Validating and restoring defense in depth using attack graphs. In *Proc. of the Military Communications Conference* (Washington DC, 2006).

[16] Liu, Y., and Man, H. Network vulnerability assessment using Bayesian networks. In *Proc. of the SPIE – Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security* (Orlando, FL, USA, 2005).

[17] Mell, P., Scarfone, K., and Romanosky, S. *CVSS – A Complete Guide to the Common Vulnerability Scoring System Version 2.0.* FIRST - Forum of Incident Response and Security Teams, June 2007.

[18] Milne, G. R., Labrecque, L. I., and Cromer, C. Toward an understanding of the online consumer's risky behavior and protection practices. *Journal of Consumer Affairs 43*, 3 (Fall 2009), 449–473.

[19] Ng, B.-Y., Kankanhalli, A., and Xu, Y. Studying users' computer security behavior: A health belief perspective. *Decision Support Systems 46*, 4 (March 2009), 815–825.

[20] Noel, S., Jajodia, S., O'Berry, B., and Jacobs, M. Efficient minimum-cost network hardening via exploit dependency graphs. In *Proc. of the 19th Annual Computer Security Applications Conference* (Las Vegas, NV, 2003).

[21] Pew Internet. What internet users do online | Pew research center's internet & american life project. http://www.pewinternet.org/Static-Pages/Trend-Data/Online-Activites-Total.aspx, October 2011.

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

[23] Rapid7. Nexpose Community Edition. Available from `http://www.rapid7.com/products/nexpose-community-edition.jsp`.

[24] Ray, I., and Poolsappasit, N. Using attack trees to identify malicious attacks from authorized insiders. In *Proc. of the 10th European Symposium on Research in Computer Security* (Milan, Italy, 2005).

[25] Ritchey, R., and Ammann, P. Using model checking to analyze network vulnerabilities. In *Proc. of the IEEE Symposium on Security and Privacy* (Oakland, CA, 2000).

[26] Rosenstock, I., Strecher, V., and Becker, M. Social learning theory and the health belief model. *Health Education Quarterly 15*, 2 (Summer 1988), 175–183.

[27] Rosenstock, I. M. Why people use health services. *Milbank Memorial Fund Quarterly 44*, 3 (1966), 94–127.

[28] SANS Institute. SANS top-20 2007 security risks (2007 annual update). Available at `http://www.sans.org/top20/2007/top20.pdf`, October 2008.

[29] Sheyner, O., Haines, J., Jha, S., Lippmann, R., and Wing, J. M. Automated generation and analysis of attack graphs. In *Proc. of the IEEE Symposium on Security and Privacy* (Oakland, CA, USA, 2002).

[30] Szewczyk, P., and Furnell, S. Assessing the online security awareness of Australian Internet users. In *Proc. of the 8th Annual Security Conference: Discourses in Security Assurance & Privacy* (Las Vegas, NV, USA, 2009).

[31] Wash, R. Folk models of home computer security. In *Proc. of the Sixth Symposium on Usable Privacy and Security* (Redmond, WA, USA, 2010).