Goal-Oriented Security Threat Mitigation Patterns: A Case of Credit Card Theft Mitigation

Sam Supakkul
Dept. of Computer Science
The Univ. of Texas at Dallas
ssupakkul@ieee.org

Tom Hill
HP Enterprise Services
HP
tom.hill@hp.com

Ebenezer Akin Oladimeji
Architecture and eServices
Verizon Communications
ebenezer.oladimeji@verizon.com

Lawrence Chung
Dept. of Computer Science
The Univ. of Texas at Dallas
chung@utdallas.edu

ABSTRACT

Most attacks on computer and software systems are caused by threats to known vulnerabilities. Part of the reason is that it is difficult to possess necessary broad and deep knowledge of security related strategic knowledge to choose mitigating solutions suitable for a specific application or organization. This paper presents three patterns that use goal-oriented concepts to capture knowledge of security problems and their corresponding mitigating solutions. Each pattern captures three kinds of problems, including undesirable outcome that negatively affects a security goal, threat that could lead to an undesirable outcome, and vulnerability that could be exploited by a threat. Alternative mitigating solutions are captured in relation to the problems, including vulnerability risk transfer, threat prevention, threat containment, undesirable outcome recovery, and undesirable outcome impact prevention and control. The alternatives are identified with consequences against other non-functional requirements (NFRs) such as cost and usability, which are then used as selection criteria in associated selection patterns. The patterns illustrate how knowledge of security incidents and security standards may be captured and used to help avoid the security problems suffered by TJX in one of the largest credit card theft incident in history.

Categories and Subject Descriptors

I.5.2 [**Computing Methodologies**]: Pattern Recognition-Design Methodologies; D.2.1 [**Software Engineering**]: Requirements/Specifications—*Patterns*

Keywords

Goal-Oriented, Pattern, Non-Functional Requirements, Security, Threat Mitigation

1. INTRODUCTION

Most attacks on computer and software systems are caused by threats to known vulnerabilities [24]. Part of the reason is that it is difficult to possess necessary broad and deep knowledge of security related knowledge to understand the security problems, alternative mitigating solutions, and the selection of suitable solutions that not only sufficiently mitigate the security problems, but also meet other organizational needs. A large body of knowledge is needed to answer important questions that can help prevent security problems, such as those suffered by TJX in one of the largest credit card theft in history. Examples of such questions are:

- What does credit card security really mean?
- What were the problems suffered by TJX?
- What were the causes of those problems?
- How did the problems and causes occur in details?
- What are the alternatives for mitigating the the problems and their causes?
- How to choose mitigating solutions that also sufficiently meet other, often conflicting, organizational needs?
- How to implement particular solutions in details?

However, it is difficult to possess such broad knowledge at the strategic level and deep knowledge at the operational level. Patterns have been used to capture knowledge of specific threats [13, 31] and solutions at the analysis level [14] and architecture/design level [45, 12, 8, 36, 33, 26, 13, 31, 27, 29] using semi-formal representation such as object-oriented notation. However, the strategic level knowledge has not been captured in a similar formal manner. For example, most security standards often describe recommended security measures informally with little context and rationale to guide us when and why specific security measure should be used, and whether they could indeed help prevent a known security incident. As a result, we face an insurmountable secure-systems design task to choose the most desirable solutions to sufficiently mitigate the threats, at the same time also sufficiently achieve other, often conflicting, organizational concerns such as cost and usability non-functional requirements (NFRs).

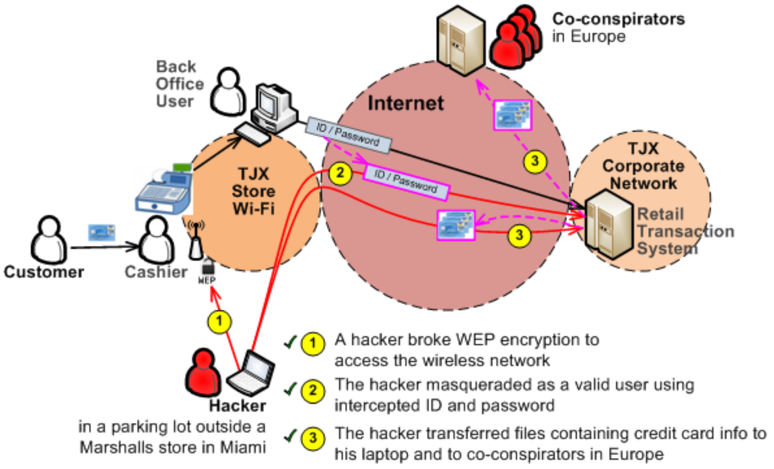
Co-conspirators
in Europe
Back
Office
User
Internet
ID / Password
ID / Password
TJX
Store
Wi-Fi
TJX
Corporate
Network
Retail
Transaction
System
Customer
Cashier
WEP
1
2
3
3
Hacker
in a parking lot outside a
Marshalls store in Miami
✓ 1 A hacker broke WEP encryption to access the wireless network
✓ 2 The hacker masqueraded as a valid user using intercepted ID and password
✓ 3 The hacker transferred files containing credit card info to his laptop and to co-conspirators in Europe

Figure 1: The Attack Scenario and Three Realized Security Threats in the TJX Incident

This paper presents goal-oriented security threat mitigation patterns for capturing cohesive strategic knowledge of security problems and solutions that capture three aspects of security problems including undesirable outcomes, their causal threats, and the exploited vulnerabilities, which are used as the basis for exploring mitigating solutions. A solution may be a risk transfer solution that mitigates a vulnerability by changing the security asset or facility so that the risk associated with the new environment is more acceptable. A solution may be a threat prevention or containment solution that prevents a threat from being realized or contains the impacts caused by a realized threat. The alternative solutions are also noted with consequences against other non-functional requirements (NFRs), which are then used as criteria for selecting suitable mitigating solutions. Three concrete patterns are presented to illustrate how knowledge of security incidents, security standards and other best practices may be captured and used to help avoid the security problems in the TJX incident.

The rest of the paper is structured as follows. Section 2 describes the TJX incident as the basis for the patterns presented in this paper. Section 2 describes the meta-pattern as a visual template for the patterns presented in Sec. 5, 6, and 6. Section 7 discusses related work and observations. Section 8 summarizes the paper and future work.

2. THE TJX CASE

TJX, the holding company of a number of large retail chains such as TJ Maxx and Marshalls, suffered one of the largest credit card theft in history between 2003 and 2008 that was initially estimated to have affected 45 million credit cards, identity information of 451,000 customers, $20 millions in fraudulent transactions, and to cost TJX $1 billion over 5 years excluding lawsuits [18, 21, 22, 32, 39, 28]. This incident became an infamous case study in the retail and credit card industries, and in the security community [6]. We have studied over 30 news articles, investigation reports, and a court indictment, which are mostly informal description of the case. We found it difficult to get a clear picture of the scenario how the incident occurred. Based on the publicly available information with some assumptions, we developed a diagram to describe the attack scenario as shown in Fig. 1. This paper presents three patterns for mitigating the three security threats described in the scenario, including wireless WEP (Wired Equivalent Privacy) hacking, remote access masquerading, and malicious transfer of sensitive data problems. Since specific detailed attacks of these threats are not reported in the publicly available sources, we therefore capture many other possible causes as well as alternative solutions from the literature and the PCI DSS security standard [30].

3. A VISUAL META-PATTERN FOR THE SECURITY MITIGATION PATTERNS

Figure 2 depicts a visual meta-pattern that serves as a visual template for understanding the patterns shown in Fig. 3, Fig. 6, and Fig. 9 based on Alexander's "three-part rule" that expresses "a relation between a certain context, a problem, and its corresponding solution(s)" [3]. In each pattern, the outer area (left , top, and right) represents the context in terms of asset and forces (security goal and other NFRs). The context provides a frame of reference for the security problems and their corresponding alternative mitigating solutions, each may have positive and/or negative consequences against the forces (NFRs). Each positive or negative contribution link (a green or red directed solid or dash line) indicates how one solution/problem contributes to the achievement or denial of the related goal/problem [9, 38]. The types of contribution link include SomePlus (S+), Make(++), Help(+), SomeMinus(S-), Break(−), and Hurt(-). SomePlus(S+) represents a general positive contribution, Make (++) a positive contribution that fully achieves the goal or causes a problem, and Help (+) a somewhat positive contribution that partially helps achieve the goal or cause a problem. Conversely, Break (−) and Hurt (-) represent negative contributions that fully or partially deny a goal/problem, while SomeMinus (S-) represents a general negative contribution with unknown certainty. A goal/problem may be refined with an AND/OR decomposition (denoted by a single- or double-bar line) to more specific goals/problems [9, 40, 38].

Security asset is either a primary asset (simply called "*asset*" in the meta-pattern) or *facility*. Primary asset is a piece of information, an operation, a physical entity, or human,

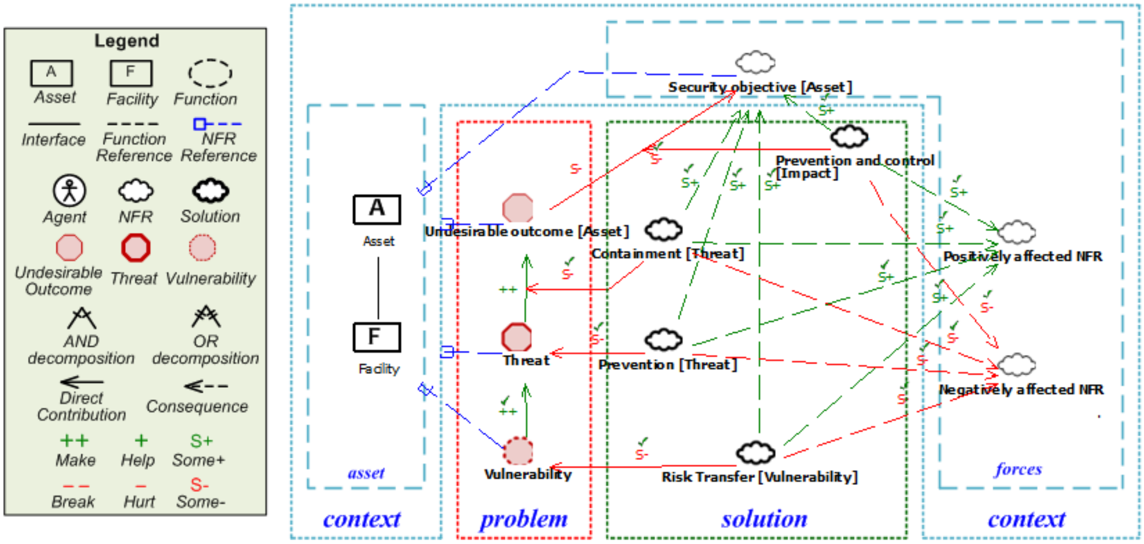

Figure 2: Visual Meta-Pattern for Security Mitigation Patterns

that is of high value that needs to be protected, for example, credit card information or an e-commerce web site. Facility is a secondary asset that when compromised could lead to an undesirable outcome against the primary asset. For example, network and server that have direct or indirect access to credit card information, when attacked by hackers, could lead to unauthorized access of the credit card information.

Security *objective* or goal is a security concern that may be different for different assets or organizations. For example, a US law defines information or system security as confidentiality, integrity, and availability security objectives [2], while the payment card industry is concerned with only the confidentiality of the payment card information. For instance, the PCI DSS recommends minimal card information storage and card authentication information not to be stored [30]. These recommendations in fact hurt the availability aspect of the information. The standard also includes no recommendations to address the integrity aspect of the information.

Security *problem* is an *undesirable outcome*, a *threat*, or a *vulnerability*. Undesirable outcome is a situation that negatively affects a security goal/objective. For example, stolen credit card information is an undesirable outcome that hurts the confidentiality of credit card information. Threat is an operation or technique that may be used to exploit a vulnerability, which is a weakness of an asset or facility. For example, the hacker in the TJX case broke the wireless network WEP encryption key by exploiting the weak WEP encryption vulnerability.

Mitigating *solution* is an *impact prevention and control*, threat *prevent and containment*, or vulnerability *risk transfer*. Impact prevention and control is a solution that prevents a realized undesirable outcome from inflicting a negative impact on the security goal. For example, PCI recommends that card authentication data (e.g. card verification code) be not stored to make the rest of credit card information useless if stolen. Threat prevention and control is a mitigating solution that helps prevent a threat from being realized, reduces or eliminates the possibility for a realized threat from causing an undesirable outcome. Vulnerability risk transfer is a solution that changes the environment (e.g. asset or facility) so that the risk associated with the existing environment is transferred to the risk of the new facility that is more acceptable. For example, the high risk associated with wireless WEP encryption could be transferred to a lower risk of a more secure wireless WPA (Wi-Fi Protected Access) encryption. The notion of risk transfer is adapted from a risk management technique discussed in [5].

4. WIRELESS WEP HACKING MITIGATION PATTERN

The Wireless WEP Hacking mitigation pattern in Fig. 3 is concerned with the confidentiality of an intranet (security goal) that can be compromised by wireless WEP hacking, which may be passive or active attacks. Passive attacks include brute-force key trying, fragmentation attack, or weak IV attack (threats). Active attacks include replay attack and insider attack (threats) that can exploit shared key authentication, short key length, known characteristics of ARP packets, and weak WEP encryption algorithm (vulnerabilities). The vulnerabilities can be mitigated by not using shared key authentication, restricting access based on MAC addresses, using at least 104-bit encryption key, using more secure encryptions such as WPA, WPA2, IPSEC, VPN, or SSL/TLS (risk transfer measures). The insider attack (threat) may be prevented by frequently changing WEP keys manually every quarter or every major personnel change, or automatically every few hours (preventive measures). These mitigation solutions have positive and/or negative consequences against cost, administration need, and usability (NFRs).

4.1 Context

4.1.1 Asset

Intranet: An internal network that has direct or indirect access to sensitive information that is desirable by hostile agents.

4.1.2 Facility

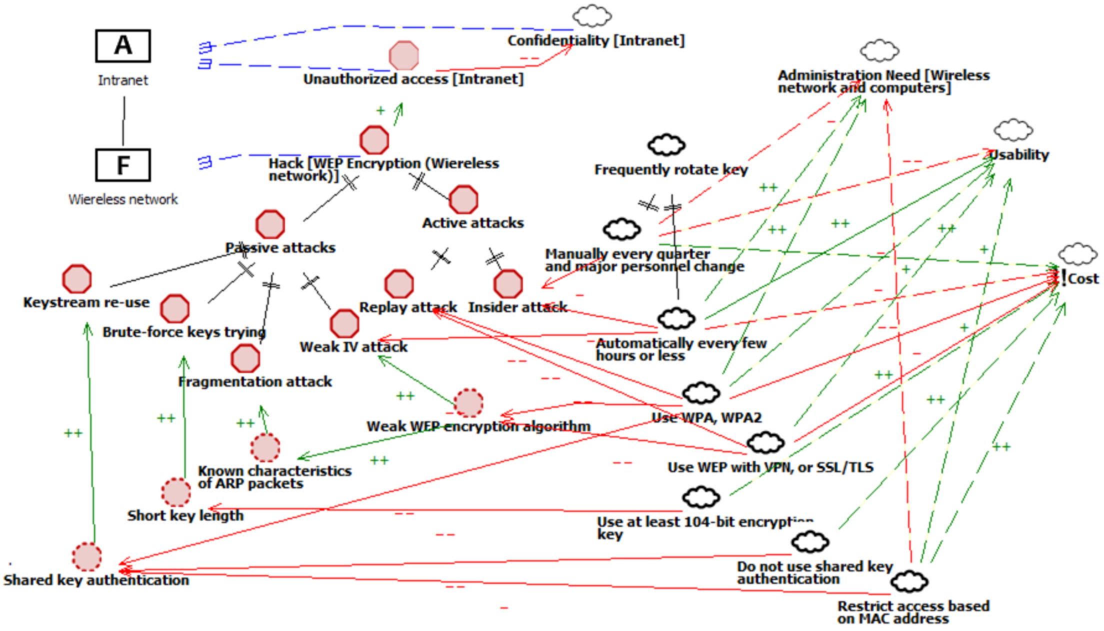
A
Intranet
F
Wiereless network
Confidentiality [Intranet]
Unauthorized access [Intranet]
Hack [WEP Encryption (Wiereless network)]
Active attacks
Passive attacks
Keystream re-use
Brute-force keys trying
Fragmentation attack
Weak IV attack
Replay attack
Insider attack
Weak WEP encryption algorithm
Known characteristics of ARP packets
Short key length
Shared key authentication
Administration Need [Wireless network and computers]
Usability
!Cost
Frequently rotate key
Manually every quarter and major personnel change
Automatically every few hours or less
Use WPA, WPA2
Use WEP with VPN, or SSL/TLS
Use at least 104-bit encryption key
Do not use shared key authentication
Restrict access based on MAC address
++
+
--
-

Figure 3: A Wireless WEP Hacking Mitigation Pattern

Wireless network: Wireless network provides a convenient and cost effective means for data communication based on IEEE 802.11 standard. Because wireless signal can be easily intercepted by hostile agents within the signal range, it is recommended that private wireless networks encrypt all transmitted data. WEP is an encryption that is widely used by the first generation wireless networks. It uses RC4 encryption algorithm that generates keystreams of 64 or 128 bits in length from a seed, which is a concatenation of a fixed 24 bits device generated Initial Vector (IV) and a user defined 40-bit or 104-bit shared key.

4.1.3 Security goal

Confidentiality [Intranet]: The intranet should only be accessible to only authorized personnel and devices.

4.2 Problems

4.2.1 Undesirable Outcome

Unauthorized access [Intranet]: Unauthorized access to the intranet by hostile agents negatively affects the confidentiality of the intranet.

4.2.2 Threat

Hack [WEP Encryption(Wireless network)]: A threat that is achieved by "Passive attacks" or "Active attacks". "Passive attacks" in turn may be achieved by "Keystream re-use", "Brute-force keys trying", "Fragmentation attack", or "Weak IV attack", while "Active attacks" may be achieved by "Replay attack" or "Insider attack".

Keystream re-use: A hacker may eavesdrop shared key authentication sessions and recover the keystreams being used. With a sufficiently large library of keystreams, encrypted data using any re-used keystreams can be decrypted by the hacker.

Brute-force key trying: A hacker may try every key combination until the key is found.

Fragmentation attack: A hacker may use the knowledge of 802.11 fragmentation feature to transmit or decrypt information without having to know the shared key [4].

Weak IV attack: A hacker may eavesdrop and collect a large number of IVs used in a wireless network. If weak IVs are re-used in the network, they can be used to crack the encryption key [15].

Replay attack: Based on the 802.11 protocol, the first 8 bytes of every packet is known, thus can be used to recover the first 8 bytes of the key. A hacker may recover an additional byte in the key by sending a 9-byte encrypted message using the recovered 8-byte key with an extra byte encrypted using a guessed 1-byte key. The hacker repeats the process with a different guessed 1-byte key until the Access Point (the wireless network controller) relays the message, indicating that the guessed key is correct. The hacker gradually lengthens the message until the entire key is recovered [4].

4.2.3 Vulnerability

Shared key authentication: Before a device may use a wireless network, the responsible Access Point may be configured to require the device to go through a shared key authentication process, a four-way challenge-response handshake testing whether the device has a correct shared key. The communication between the Access Point and the device exposes a pair of clear text and its corresponding encrypted text that can be eavesdropped by hostile agents.

Short key length: A key of 40 bits can be cracked by trying all combinations in less than a month [4].

Known characteristics of ARP packets: Address Resolution Protocol (ARP) is used by devices in an Internet Protocol (IP) network to look up a physical hardware address from a logical network address. ARP data packets have a fixed length format, and a portion of the packets is transmitted in clear text even when WEP is used. Knowing these characteristics allows hackers to learn about and hack a wireless network.

Weak WEP encryption algorithm: RC4 encryption algorithm used by WEP has flaws that allow certain Initial Vectors (IVs) to reveal more information about the rest of key than others [15].

4.3 Solutions

4.3.1 Threat Prevention

Frequently rotating the WEP key: The WEP key should be frequently rotated, either manually every quarter or whenever there are changes in personnel that have access to keys (PCI DSS Requirement 4.1.1 [30]) or automatically every few hours.

Manually rotating the key every quarter or every personnel change: This solution is to prevent attacks from insiders who have access to the key.

Positive Consequences

> *Help (+) cost*: This is a procedural solution; therefore, no additional equipment cost is required.

Negative Consequences

> *Hurt (-) administration needs*: A system administrator needs to manually change or rotate the key.
>
> *Break (–) usability*: User intervention may be required to reconfigure the wireless computers or devices whenever the key is changed.
>
> *Hurt (-) availability*: During the key rotation, the wireless network is not available to the users until their wireless devices or computers have been updated with the new key.

Automatically rotating the key every few hours or less: Weak IV attack generally needs to eavesdrop over 1,000,000 packets to crack a key. Changing the key every few hours would make previously collected encrypted packets useless, thus, spoiling the attack before it could succeed. Since frequently changing key every few hours is impractical to perform manually, a number of products provide automated dynamic keys distribution [44].

*Positive Consequence*s

> *Make (++) administration needs*: Automated key rotation requires little or no intervention from a system administrator.
>
> *Make (++) usability*: User intervention is not required during the key rotation.

Negative Consequences

> *Hurt (-) cost*: This solution requires a key distribution tool such as RADIUS that may incur additional cost.

Using WEP with more secure supplemental technologies: More secure encryption such as IPSEC, VPN, and SSL/TLS should be used in conjunction with the insecure WEP (PCI DSS Requirement 4.1.1 [30]).

Positive Consequences

> *Make (++) administration needs*: Once wireless computers or devices have been configured, there is little need for administration.
>
> *Make (++) usability*: User intervention is not required once the wireless computer and devices have been set up.

Negative Consequences

> *Hurt (-) cost*: Moderate cost may be incurred if hardware or software is needed to support technologies such as VPN.

4.3.2 Risk Transfer

Using a more secure encryption: Encryption methods, such as WPA, WPA2, VPN, or SSL/TLS are harder to crack, therefore, should be used instead of WEP (PCI DSS Requirement 4.1 [30]).

Positive Consequences

> *Make (++) usability*: Connected network devices and computers do not require re-configuration once they have been set up to use the new encryption method.

Negative Consequences

> *Hurt (-) cost*: New equipment that supports WPA, WPA2, or VPN software may be required.

Using at least 104-bit encryption key: A key of 104 bits is generally sufficient in preventing a brute-force attack as it would take millions of years to crack in theory [43].

Positive Consequences

> *Make (++) cost*: Most wireless access point equipment support 104-bit keys for WEP encryption.

Negative Consequences

> None. No negative consequences to other NFRs of interest.

Restricting access by MAC addresses: Wireless access point should be configured to restrict access by the MAC addresses of authorized computers (PCI DSS Requirement 4.1.1 [30]).

Positive Consequences

> *Make (++) cost*: Most wireless access points support this technique, therefore, no additional equipment is required

Negative Consequences

> *Hurt (-) administration need*: A network administrator needs to re-configure the wireless access point whenever a device is added or removed from the network.

4.4 Mitigation Selection

4.4.1 Cost Driven Mitigation

Figure 4 depicts recommended solutions based on the alternatives shown in Fig. 3 with an emphasis on cost (denoted by “!!Cost”). The recommended solutions are made at

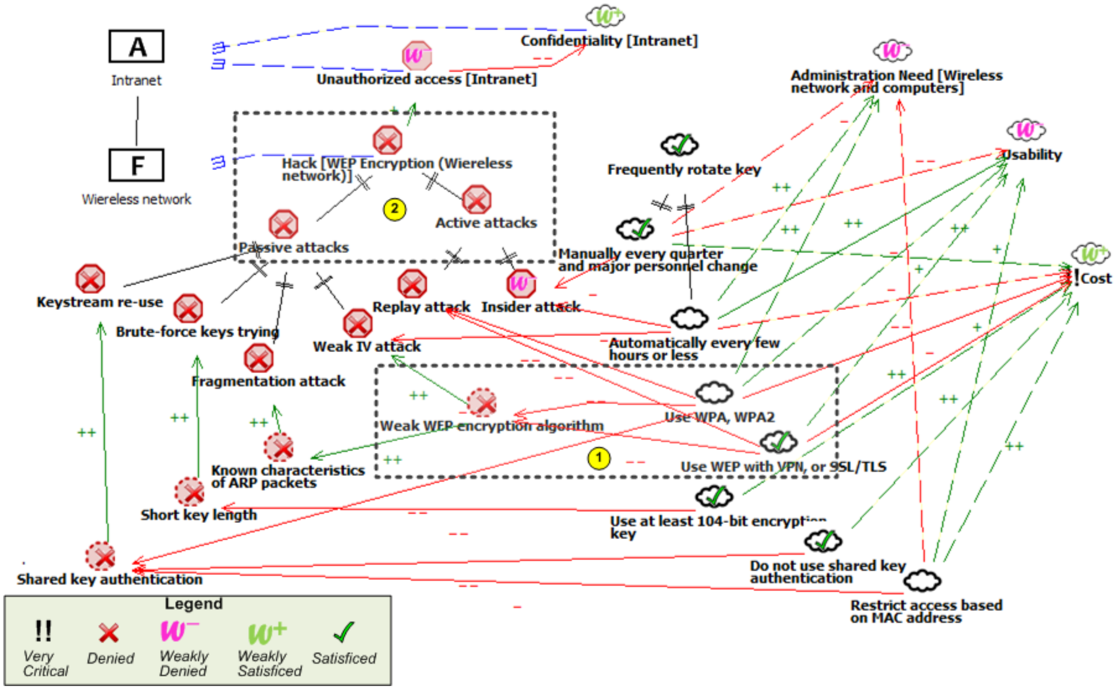
A
Intranet
F
Wiereless network
Confidentiality [Intranet]
Unauthorized access [Intranet]
Administration Need [Wireless network and computers]
Usability
!Cost
Hack [WEP Encryption (Wiereless network)]
2
Active attacks
Passive attacks
Frequently rotate key
Manually every quarter and major personnel change
Keystream re-use
Brute-force keys trying
Replay attack
Insider attack
Weak IV attack
Automatically every few hours or less
Fragmentation attack
Weak WEP encryption algorithm
Use WPA, WPA2
Known characteristics of ARP packets
1
Use WEP with VPN, or SSL/TLS
Short key length
Use at least 104-bit encryption key
Shared key authentication
Do not use shared key authentication
Restrict access based on MAC address
Legend
!! Very Critical
Denied
Weakly Denied
Weakly Satisficed
Satisficed

Figure 4: Cost Driven Mitigation for Wireless WEP Hacking

the leaf-level solutions (those solutions that are not the target of any decompositions or contributions) and are denoted by satisficed labels or check marks. To mitigate the web hacking problem, its undesirable outcomes, threats, and/or vulnerabilities should be mitigated in order to satisfice[1] the security objective. The denial of security problems are denoted by a cross mark. Partially satisfied goals or denied problems are denoted by weakly satisfied (W+) and weakly denied (W-) labels respectively. To determine whether the problems are sufficiently mitigated and whether the ultimate security goal is sufficiently satisfied, the label of the recommended solutions are propagated upward the goal/problem graph by consulting a label evaluation catalog, such as the one shown in Appendix A.

For illustration purposes, area 1 in Fig. 4 shows that "Use WEP with VPN, or SSL/TLS" solution is recommended (denoted by a Satisficed label or a check mark) over "Use WPA, WPA2" because the former is less costly (denoted by a Hurt/- contribution vs. a Break/– contribution toward Cost respectively). By consulting the evaluation catalog, the satisficed label of "Use WEP with VPN, or SSL/TLS" solution is changed to a denied label for "Weak WEP-encryption algorithm" problem over the Break(–) contribution that links the two. In area 2, "Hack[WEP Encryption...]" problem is evaluated to be denied because both of its OR-decomposed sub-problems ("Passive attacks" and "Active attacks") are denied [38]. The evaluation is repeated until the root goals are evaluated. In this case, "Confidentiality [Intranet]" and "Cost" are evaluated to be weakly satisficed, the best trade-off results for the available alternatives against the security and cost concerns.

In summary, the following are the recommended mitigation for the cost concern that should be used conjunctively:

- Manually [rotate WEP key] every quarter or every major personnel change
- Use at least 104-bit encryption key
- Do not use shared key authentication
- Use WEP with VPN, or SSL/TLS

4.4.2 Usability Driven Mitigation

Figure 5 depicts recommended solutions with an emphasis on usability or friendliness of the solutions (denoted by "!!Usability"). For example, "Frequently rotate key" can be used to counter "Insider attack" by either rotating the key "Manually every quarter" or "Automatically every few hours". The latter is recommended because it does not require user intervention, it is therefore more friendly. Other solutions are recommended to counter other sub-problems of "Hack [WEP EncryptionĚ]" threat so that ultimately, "Confidentiality [Intranet]" is evaluated to be weakly satisficed and "Usability" to be satisficed.

The following are the recommended mitigation that should be used conjunctively:

- Automatically [rotate WEK key] every few hours or less
- Use WPA or WPA2
- Use at least 104-bit encryption key
- Do not use shared key authentication

[1] *Satisfice* has an ancient origin (circa 1600), it is a Scottish variant of satisfy, referring to the notion of "good enough" [37]. It is adopted by the NFR Framework [9] for noting the achievement of non-functional requirements (NFRs), which, for their subjective nature, are usually satisficed rather than satisfied in an absolute sense.

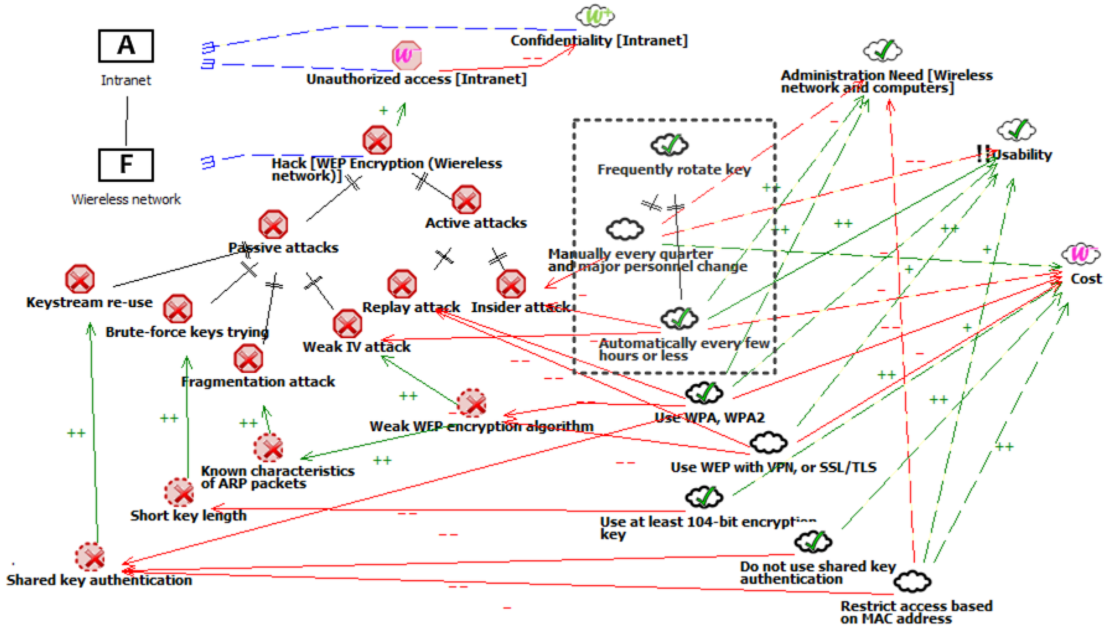
A
Intranet
F
Wiereless network
Confidentiality [Intranet]
Unauthorized access [Intranet]
Hack [WEP Encryption (Wiereless network)]
Active attacks
Passive attacks
Keystream re-use
Brute-force keys trying
Fragmentation attack
Weak IV attack
Replay attack
Insider attack
Weak WEP encryption algorithm
Known characteristics of ARP packets
Short key length
Shared key authentication
Administration Need [Wireless network and computers]
Usability
Cost
Frequently rotate key
Manually every quarter and major personnel change
Automatically every few hours or less
Use WPA, WPA2
Use WEP with VPN, or SSL/TLS
Use at least 104-bit encrypt key
Do not use shared key authentication
Restrict access based on MAC address
++
--
+
-

Figure 5: Usability Driven Mitigation for Wireless WEP Hacking

4.5 Known Uses

Over 700 organizations are using and in compliance with the PCI DSS [41].

5. REMOTE ACCESS MASQUERADING MITIGATION PATTERN

The Remote Access Masquerading (mitigation pattern) in Fig. 6 is concerned with the confidentiality of a corporate server (security goal) that can be compromised by a hacker masquerading as a valid user using stolen ID and password, which can be accomplished by illegally obtaining the ID and password and using them for login. Login information could be obtained via (1) social engineering, (2) intercepting, or (3) passwords guessing (threats) that exploit (1) some undesirable user behaviors, (2) interceptable IDs and passwords, and (3) simple passwords being used (vulnerabilities).

Undesirable user behaviors, interceptable IDs and passwords, and simple passwords being used (vulnerabilities) can be mitigated by securing IDs and passwords via (1) user education, training, and enforcement, (2) encrypting stored and transmitted IDs and passwords, (3) using strong passwords that are frequently changed non-dictionary words (risk transfer measures). Alternatively, hacker masquerading as a valid user using stolen ID and password (threat) may be prevented by using two-factor authentication that uses ID/password with certificate/token or ID/password with biometrics (preventive measures). Unauthorized access to the server (undesirable outcome) can be recovered by resetting the compromised passwords (impact prevention or recovery) to limit the impacts on the unauthorized access (security goal). These mitigation solutions have positive and/or negative consequences upon cost and usability (NFRs).

5.1 Context

5.1.1 Asset

Corporate server: A corporate computer server that has access to sensitive data.

5.1.2 Facility

Remote login facilities: Computing facilities such as an intranet, the Internet, and an ID/password based login process allow remote users outside the corporate network to login to a corporate server.

5.1.3 Security goal

Confidentiality [Corporate server]: Every corporate server should be accessible to only authorized personnel and devices.

5.2 Problems

5.2.1 Undesirable Outcome

Unauthorized access to the server: A server that is accessible to hostile agents.

5.2.2 Threat

Masquerade [Remote login using ID/password]: A hostile agent may masquerade as an authorized user logging into a server by illegally obtaining an ID and a password, then login to the server using the stolen ID and password. ID and password may be illegally obtained by social engineering attack, interception, or passwords guessing.

5.2.3 Vulnerability

Undesirable user behaviors: Certain user behaviors are undesirable, such as posting note with a login ID and password nearby the computer. Some users may be easily tricked into revealing their ID and password by social engineering attacks and phishing

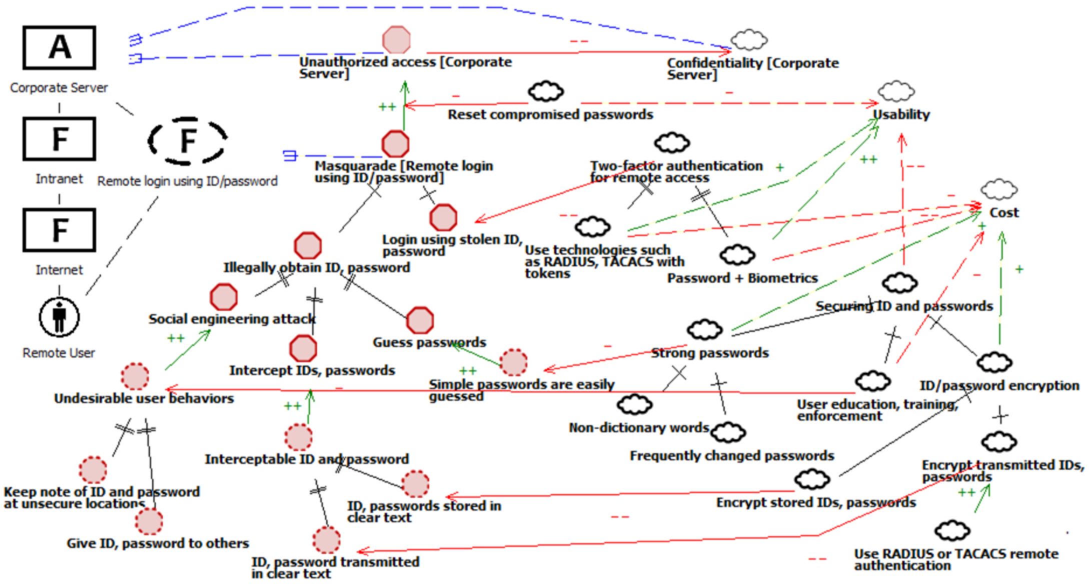
A
Corporate Server
F
Intranet
F
Internet
Remote User
F
Remote login using ID/password
Unauthorized access [Corporate Server]
Confidentiality [Corporate Server]
Reset compromised passwords
Usability
Masquarade [Remote login using ID/password]
Two-factor authentication for remote access
Cost
Login using stolen ID, password
Use technologies such as RADIUS, TACACS with tokens
Password + Biometrics
Illegally obtain ID, password
Securing ID and passwords
Social engineering attack
Guess passwords
Strong passwords
Intercept IDs, passwords
Simple passwords are easily guessed
ID/password encryption
Undesirable user behaviors
Non-dictionary words
User education, training, enforcement
Frequently changed passwords
Interceptable ID and password
Encrypt transmitted IDs, passwords
Keep note of ID and password at unsecure locations
ID, passwords stored in clear text
Encrypt stored IDs, passwords
Give ID, password to others
ID, password transmitted in clear text
Use RADIUS or TACACS remote authentication

Figure 6: Remote Access Masquerading Mitigation Pattern

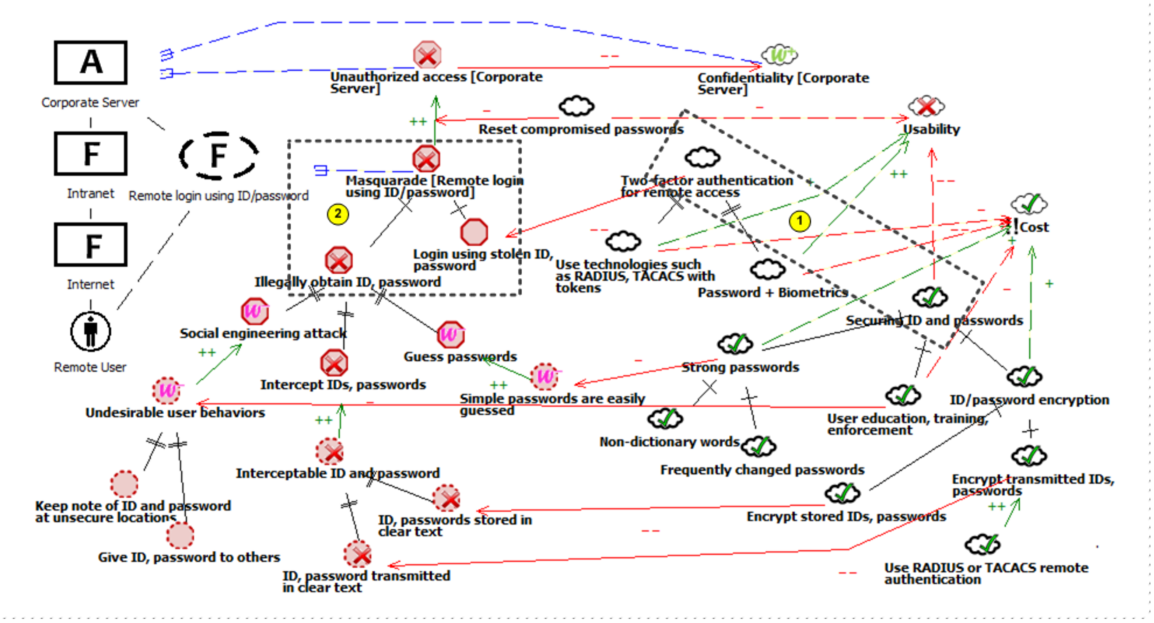
A
Corporate Server
F
Intranet
F
Internet
Remote User
F
Remote login using ID/password
Unauthorized access [Corporate Server]
Confidentiality [Corporate Server]
Reset compromised passwords
Usability
Masquarade [Remote login using ID/password]
2
1
Two-factor authentication for remote access
Cost
Login using stolen ID, password
Use technologies such as RADIUS, TACACS with tokens
Password + Biometrics
Illegally obtain ID, password
Securing ID and passwords
Social engineering attack
Guess passwords
Strong passwords
Intercept IDs, passwords
Simple passwords are easily guessed
ID/password encryption
Undesirable user behaviors
Non-dictionary words
User education, training, enforcement
Frequently changed passwords
Interceptable ID and password
Encrypt transmitted IDs, passwords
Keep note of ID and password at unsecure locations
ID, passwords stored in clear text
Encrypt stored IDs, passwords
Give ID, password to others
ID, password transmitted in clear text
Use RADIUS or TACACS remote authentication

Figure 7: Cost Driven Mitigation for Remote Access Masquerading

Interceptable ID and password: IDs and passwords can be intercepted from a repository or during a transmission if they are stored or transmitted in clear text.

Simple passwords are easy to guess: Simple words, such as those exist in a dictionary, are easy to remember for the user, but at the same are also easy to guess and attack.

5.3 Solutions

5.3.1 Undesirable Outcome Recovery

Resetting compromised passwords: Passwords of the accounts that are suspected to have been compromised may be reset by the account owner or the account may be disabled by the system.

Positive Consequences

> *Make (++) cost*: This solution should be easy to implement without additional hardware or software cost.

Negative Consequences

> *Hurt (-) usability*: Affected users need to remember the new password or re-activate the account if disabled by the system.

5.3.2 Threat Prevention

Use two-factor authentication: Typical authentication requires the user to provide some forms of credentials, including what she knows (e.g. a password), what she has (e.g. a security token), or what she is (e.g. being identifiable by her biometrics). Two-factor authentication requires the user to provide two forms of credentials, which can be facilitated by technologies such as RADIUS (PCI DSS Requirement 8.3 [30]).

Positive Consequences

> *Make (++) usability*: Two-factor authentications using security token or biometrics require the users to memorize little or no information.

Negative Consequences

> *Hurt (-) cost*: Two-factor authentication using password with security token or biometrics is more costly than one-factor password based authentication due to the cost of the security tokens and the supporting hardware/software, or the cost of collecting and maintaining biometrics samples.

5.3.3 Risk Transfer

Not available. It is unlikely that ID and password based authentication could be replaced since it is widely used, cost effective, and universally supported by most computing platforms.

5.4 Mitigation Selection

5.4.1 Cost Driven Mitigation

Figure 7 depicts recommended mitigating solutions with an emphasis on the cost of the solutions. Because "Masquerade [Remote login using ID/password]" is AND-decomposed to "Illegally obtain ID, password" and "Login using stolen ID, password" sub-problems, only one of the sub-problems needs to be mitigated in order to mitigate the parent problem [38]. In this case, only "Illegally obtain ID, password" sub-problem is mitigated since its corresponding solutions (i.e. Securing ID and passwords) are less costly than "Login using stolen ID and password" sub-problem (i.e. Two-factor authentication for remote access).

The following are the recommended mitigation that should be used conjunctively:

- Securing ID and passwords, which consists of:
  - Strong passwords (e.g. frequently changed non-dictionary words)
  - User education, training, enforcement

5.4.2 Usability Driven Mitigation

Figure 8 depicts recommended solutions with an emphasis on the usability of the solutions. For illustration purposes, area 1 in Fig. 8 shows that "Password & Biometrics two-factor authentication for remote access" is chosen over "securing ID and passwords", as the latter would require the users to memorize frequently changed non-dictionary passwords. Similar to the cost driven mitigation, countering only "Login using stolen ID, password" sub-problem (in area 2) is sufficient for preventing the "Masquerade [Remote login...]" problem because of the AND-decomposition [38].

For the usability concern, two-factor authentication using biometrics is recommended. The label evaluation concludes that "Confidentiality [Corporate Server]" is weakly satisficed and "Usability" satisficed.

5.5 Known Uses

Over 700 organizations are using and in compliance with the PCI DSS [41].

6. MALICIOUS TRANSFER OF SENSITIVE DATA MITIGATION PATTERN

The Malicious Transfer of Sensitive Data (mitigation pattern) in Fig. 9 is concerned with the confidentiality of sensitive data (security goal) that can be compromised by a malicious transfer of sensitive data to an external host (threat), a threat that exploits a vulnerability that allows connections to be made to any external hosts.

The vulnerability of allowing connections to external hostile hosts can be mitigated by building a firewall configuration to restrict outbound traffic to that which is necessary for the sensitive data environment, to restrict outbound traffic from authorized applications to IP addresses within the DMZ, and to deny all outbound traffic that is not specially allowed (risk transfer measures). Alternatively, the malicious transfer of sensitive data (threat) can be contained by recovering the affected facility that can be achieved by logging network activities, frequently monitoring the logs, and isolating the affected facility when a malicious transfer has been detected (containment measures). The stolen sensitive data undesirable outcome can be prevented from negatively affecting the confidentiality of sensitive data (security goal) by not storing complete data or storing the data in a different form that is not useful to hostile agents (impact prevention and control). These mitigation solutions have positive and/or negative consequences against cost and availability (NFRs).

6.1 Context

6.1.1 Asset

Sensitive data: Sensitive data such as credit card information that needs to be protected.

6.1.2 Facility

Internal host, Intranet, Internet, External host: Internal hosts in the intranet and external hosts in the Internet are involved in a malicious transfer of sensitive data.

6.1.3 Security goal

Confidentiality of the sensitive data: Sensitive data should be accessible to only authorized personnel and devices.

6.2 Problems

6.2.1 Undesirable Outcome

Stolen sensitive data: Sensitive data may be obtained by a hostile agent.

6.2.2 Threat

Malicious transfer of sensitive data to an external host: The threat is composed of two sub-problems: mali-

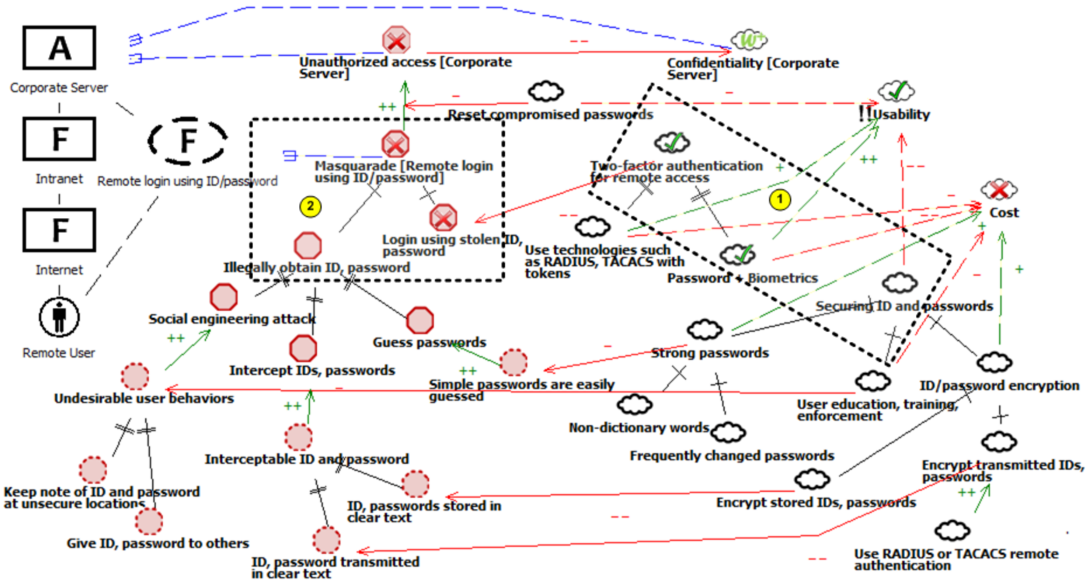
A
Corporate Server
F
Intranet
F
Internet
Remote User
F
Remote login using ID/password
Unauthorized access [Corporate Server]
Confidentiality [Corporate Server]
Reset compromised passwords
!!Usability
Masquarade [Remote login using ID/password]
Two-factor authentication for remote access
Login using stolen ID, password
Use technologies such as RADIUS, TACACS with tokens
Password + Biometrics
Cost
Illegally obtain ID, password
Social engineering attack
Guess passwords
Intercept IDs, passwords
Simple passwords are easily guessed
Strong passwords
Securing ID and passwords
ID/password encryption
Undesirable user behaviors
User education, training, enforcement
Non-dictionary words
Frequently changed passwords
Interceptable ID and password
Encrypt transmitted IDs, passwords
Keep note of ID and password at unsecure locations
ID, passwords stored in clear text
Encrypt stored IDs, passwords
Give ID, password to others
ID, password transmitted in clear text
Use RADIUS or TACACS remote authentication
1
2
++
--
+
-

Figure 8: Usability Driven Mitigation for Remote Access Masquerading

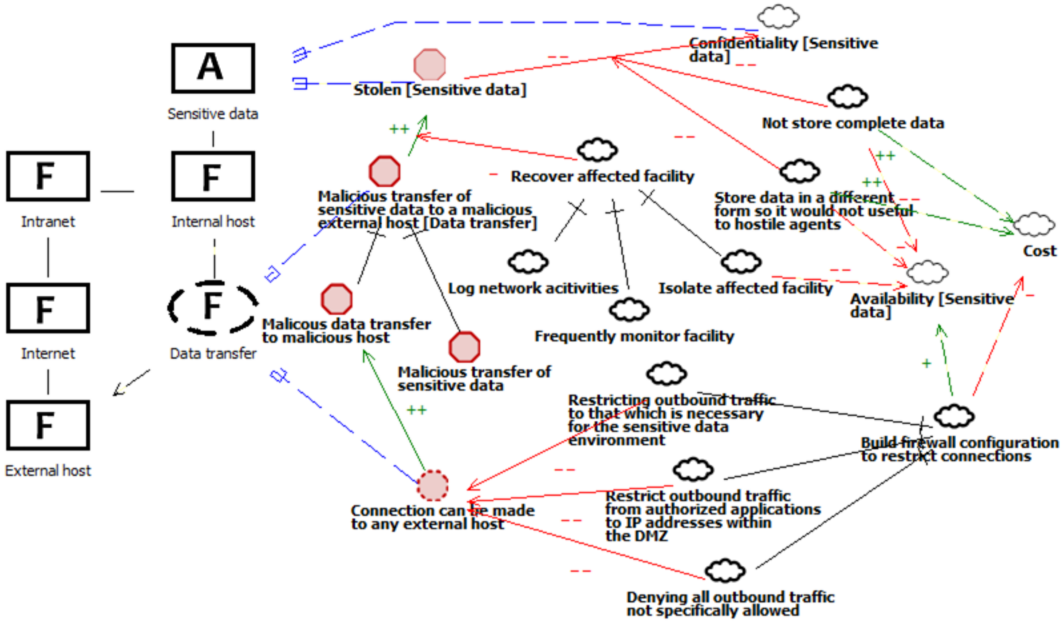
A
Sensitive data
Confidentiality [Sensitive data]
Stolen [Sensitive data]
Not store complete data
F
Intranet
F
Internal host
Recover affected facility
Malicious transfer of sensitive data to a malicious external host [Data transfer]
Store data in a different form so it would not useful to hostile agents
Cost
F
Internet
F
Data transfer
Log network acitivities
Isolate affected facility
Availability [Sensitive data]
Malicous data transfer to malicious host
Frequently monitor facility
Malicious transfer of sensitive data
F
External host
Restricting outbound traffic to that which is necessary for the sensitive data environment
Build firewall configuration to restrict connections
Connection can be made to any external host
Restrict outbound traffic from authorized applications to IP addresses within the DMZ
Denying all outbound traffic not specifically allowed
++
--
+
-

Figure 9: Malicious Transfer of Sensitive Data Mitigation Pattern

cious transfer of sensitive data and malicious transfer to a malicious external host.

6.2.3 *Vulnerability*

Connection can be made to any external host: A typical network connection can be made between any internal host on the intranet and any host in the Internet by default, for example, via an "ftp" operation.

6.3 Solutions

6.3.1 *Impact Prevention*

Not storing complete data: certain pieces of important information should not be stored with the rest of the sensitive data, for example, card authentication code should not be stored (PCI DSS Requirement 3.2 [30]) so that the rest of credit card information would not be useful to hackers if stolen.

Positive Consequences

Make (++) cost: No equipment and effort is required.

Negative Consequences

Break (–) availability: The missing pieces of data would not be available for other legitimate uses.

Storing data in a different form that is not useful to hostile agents: Information, such as driver license number, that is needed for authentication purposes may be stored in a different form, for example, as one-way hashed values that can be used for the same purposes (e.g. authenticating customers), but would not be useful to hostile agents if stolen. If the original information needs to be retained, it may be stored in an encrypted form. The use of hashed values and encryption are recommended by the PCI DSS Requirement 3.4 [30].

Positive Consequences

Make (++) cost: No additional equipment is required

Negative Consequences

Break (–) availability: The missing data would not be available for other legitimate purposes.

6.3.2 Threat Containment

Recovering the affected facility: Organizations should maintain an audit trail of network activities (PCI DSS Requirement 10.2 [30]), frequently monitor the facility (PCI DSS Requirement 10.6 [30]), and isolate the affected facility when a malicious transfer has been detected.

Positive Consequences

None to other NFRs of interest besides the indirect weakly satisficing of the confidentiality of sensitive data.

Negative Consequences

Break (–) availability [sensitive data]: Legitimate use of the sensitive data is inconvenient or not possible if it is not stored or stored with an encryption.

6.3.3 Risk Transfer

Build a firewall configuration to restrict connections: The risk of malicious connections can be mitigated by a network firewall configuration that restricts outbound network connections to those with destinations that are necessary for the sensitive data environment, to those with origins only from authorized applications, and to those with target IP addresses in the demilitarized zone (DMZ). The configuration should also deny all outbound traffic that is not specially allowed (PCI DSS Requirement 1.3.5 [30])

Positive Consequences

None, other than the indirect weakly satisficing of the confidentiality of sensitive data.

Negative Consequences

Hurt (-) availability: Unplanned legitimate network connections and data transfer of data would be restricted.

Hurt (-) cost: Additional equipment such as network firewalls to restrict connections and to provide a demilitarized zone (DMZ) are required.

6.4 Mitigation Selection

6.4.1 Cost Driven Mitigation

Figure 10 depicts recommended solutions with an emphasis on the cost of the solution. In this case, “Not store complete data” prevents “Stolen [Sensitive data]” undesirable outcome from affecting “Confidentiality [Sensitive data]” without any associated cost. The prevention is represented by the denial of the Break(–) contribution between “Stolen [Sensitive data]” and “Confidentiality [Sensitive data]”. Since the contribution is denied, regardless whether “Stolen [Sensitive data]” problem is satisficed or not, “Confidentiality [Sensitive data]” cannot be denied [9]. Therefore, there is no need to counter the “Stolen [Sensitive data]” problem.

6.4.2 Availability Driven Mitigation

Figure 11 depicts recommended solutions with an emphasis on the availability of the sensitive data. To mitigate “Malicious transfer of sensitive data to a malicious external host” that is AND-decomposed sub-problems, we need to mitigate one or more of the sub-problems [38]. Because it is difficult to prevent “Malicious transfer of sensitive data sub-problem” as it is intentional in nature, we therefore mitigate “Malicious transfer to a malicious host” sub-problem by mitigating its root cause using a firewall configuration to restrict outbound connections.

The following are the recommended mitigation that should be used conjunctively:

- Build a firewall configuration to restrict connections that breaks/– Connection can be made to any external host vulnerability. Specific firewall configuration includes:
  - Restricting outbound traffic to that which is necessary for the sensitive data environment
  - Restricting outbound traffic from authorized applications to IP addresses within the DMZ
  - Denying all outbound traffic not specifically allowed

6.5 Known Uses

Over 700 organizations are using and in compliance with the PCI DSS [41].

7. RELATED WORK AND DISCUSSION

Knowledge patterns were first introduced to capture solutions of recurring town and home building problems [3], which were later adapted for software design [17]. The use of pattern has later spread to other types of software knowledge including requirements [16] and architecture [45], with many patterns specializing in the area of software and system security [45, 12, 8, 36, 33, 26, 13, 31, 27, 29] for capturing knowledge of security mechanisms such as access controls at the analysis, architecture, and design levels. Patterns are also used to capture security problems such as security misuses/attacks [13, 31] to help understand how each problem could happen. Most of these patterns are represented using object-oriented notations. Other security patterns capture security solutions from the perspective of agents and their collaborations [27, 29], or high-level domain and phenomena of security problems [25].

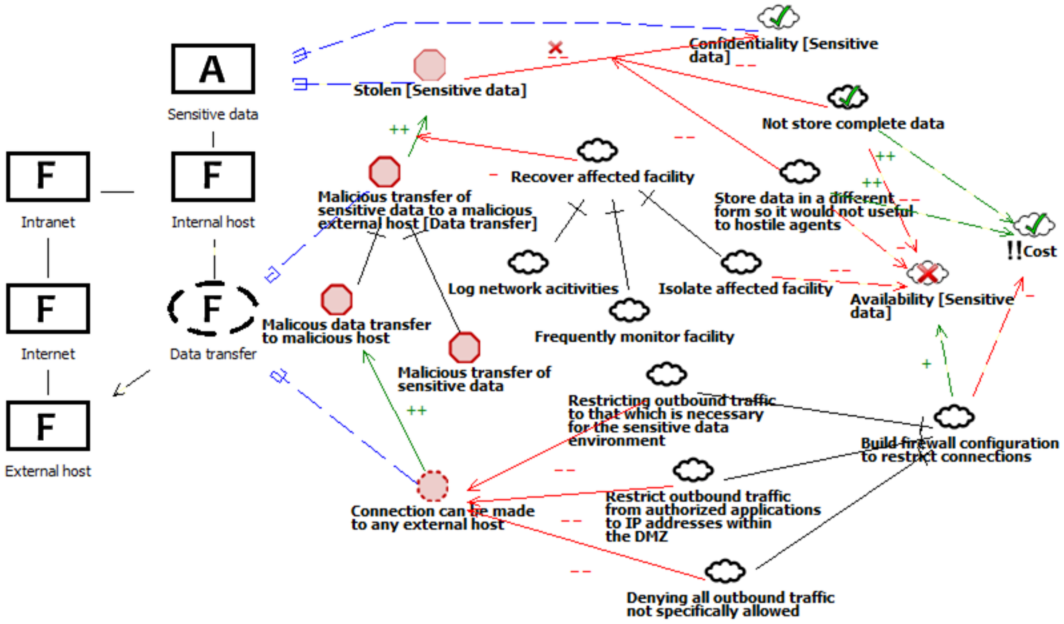
A
Sensitive data
F
Intranet
F
Internal host
F
Internet
F
Data transfer
F
External host
Confidentiality [Sensitive data]
Stolen [Sensitive data]
Not store complete data
Recover affected facility
Malicious transfer of sensitive data to a malicious external host [Data transfer]
Store data in a different form so it would not useful to hostile agents
!!Cost
Log network acitivities
Isolate affected facility
Availability [Sensitive data]
Malicous data transfer to malicious host
Frequently monitor facility
Malicious transfer of sensitive data
Restricting outbound traffic to that which is necessary for the sensitive data environment
Build firewall configuration to restrict connections
Connection can be made to any external host
Restrict outbound traffic from authorized applications to IP addresses within the DMZ
Denying all outbound traffic not specifically allowed

Figure 10: Cost Driven Mitigation for Remote Access Masquerading

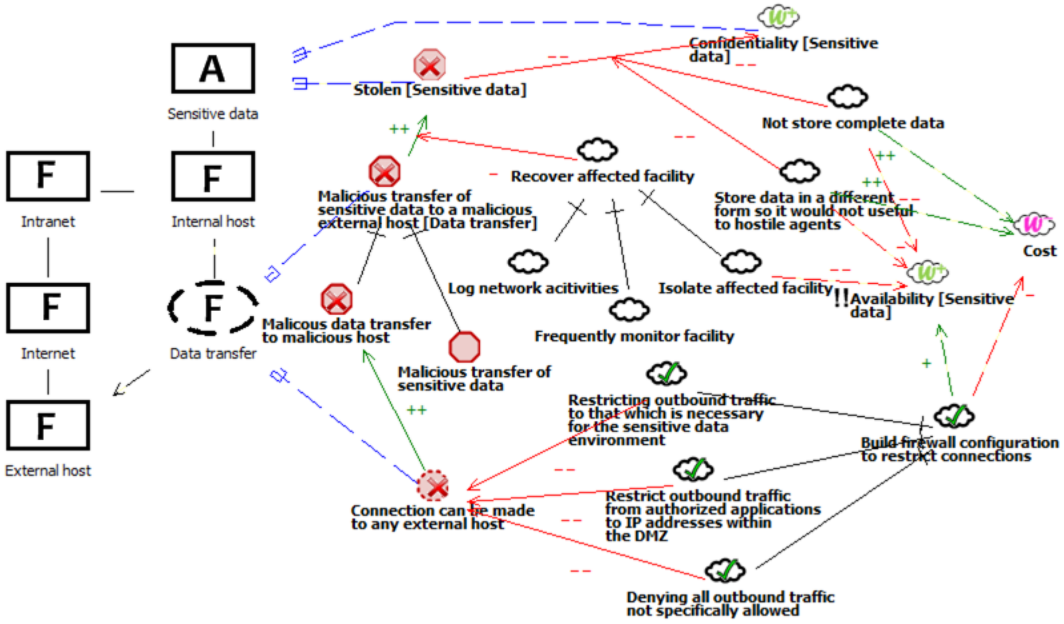
A
Sensitive data
F
Intranet
F
Internal host
F
Internet
F
Data transfer
F
External host
Confidentiality [Sensitive data]
Stolen [Sensitive data]
Not store complete data
Recover affected facility
Malicious transfer of sensitive data to a malicious external host [Data transfer]
Store data in a different form so it would not useful to hostile agents
Cost
Log network acitivities
Isolate affected facility
!!Availability [Sensitive data]
Malicous data transfer to malicious host
Frequently monitor facility
Malicious transfer of sensitive data
Restricting outbound traffic to that which is necessary for the sensitive data environment
Build firewall configuration to restrict connections
Connection can be made to any external host
Restrict outbound traffic from authorized applications to IP addresses within the DMZ
Denying all outbound traffic not specifically allowed

Figure 11: Availability Driven Mitigation for Remote Access Masquerading

Unlike the traditional security/attack patterns that describe operational level details of specific security mechanisms or attacks, patterns in this paper may seem to simply list a number of problems, solutions, and their contribution relationships without much details, in the manner that may not be considered patterns to some. However, these patterns capture important broad strategic knowledge of security that is necessary for making decisions at the organizational or requirements engineering level. The difference in the level of abstraction leads to the difference in the naming convention: the traditional security/attack patterns tend to have more memorable names that reflect the mechanisms or attacking techniques such as “Role-Based Access Control” [26] or “VoIP Denial-of-Service Attack” [13] while the names of the patterns in this papers tend to be more generic. However, these two kinds of security patterns can be used in an integrated and complementary manner. For example, the alternative solutions in the strategic patterns may refer to the operational level patterns. The integrated pattern system would provide the necessary knowledge to help security engineers make decisions at the strategic level where they can inspect more details about specific problems or solutions by consulting the associated operational level patterns.

The non-functional requirements driven tradeoff and satisficing (good enough) seem to reflect some of real-life practices where organizations do not always aim for maximum security regardless of other concerns. For example, despite being a highly secure authentication method, biometric authentication is generally not used by most e-commerce web sites, due to its high cost and the privacy concern that the potential customers may have concerning the possession of their biometric samples by the web sites. But it is noted that the NFRs based consequences, although desirable, are difficult to evaluate [11]. They are therefore assigned and evaluated qualitatively in our patterns. The NFRs driven selection has been introduced for design patterns and security patterns [20, 42] as conceptual guidelines, where the patterns in this paper capture concrete application-specific selections. The use of security standards as the source of security patterns has previously been discussed in [35].

During the development of the patterns in this paper, we found that the integrated goal-oriented analysis of problems and solutions provides the rationale for why or if a solution set is adequate. For example, since the PCI Data Security Standard [30] does not seem to address the social engineering and phishing attacks, how are organizations assured that the PCI DSS is adequate in protecting the confidentiality of credit card information? Figure 6 shows that the user masquerading problem is AND-decomposed to illegally obtaining ID/password (e.g. via social engineering) and logging in using the stolen ID/password. Using goal-oriented problem analysis, such as the approach described in [38], only one of of the sub-problems needs to be mitigated in order to mitigate the parent problem. In the case of PCI DSS, although it does not seem to address the social engineering attack, but its two-factor authentication recommendation could mitigate the logging in using the stolen ID/password sub-problem. It could, as the result, also mitigate the user masquerading parent problem. On the other hand, Fig. 9 shows that "Recover affected facility" consists of "Log network activities", "Frequently monitor facility", and "Isolate affected facility" solutions, but PCI DSS recommends only the first two measures without the measure to isolate the affected facility, which suggests that PCI DSS recommendations may be incomplete. This illustrates that goal/problem based rationale could be helpful in justifying and ensuring that the recommendations in security standards are sound (i.e. why the recommendations can mitigate certain problems) and complete (i.e. adequate recommendations to mitigate certain problems).

The patterns in this paper were modeled using the RE-Tools [1] that supports Problem Frames [23], the NFR Framework [9], and the Problem Interdependency Graph [38] notations used in this paper, as well as other related notations including the i* Framework [46], and KAOS [10]. Many of the notations are closely related. For example, the NFR Framework was originally embraced by the i* Framework [46], which was later adopted by Tropos [7], which in turn became the foundation for Secure Tropos [19].

8. CONCLUSION

In this paper, we have presented security mitigation patterns that use goal-oriented techniques to capture and relate security context, problems, and alternative mitigating solutions with recommendations that based on different organizational goals such as non-functional requirements (NFRs). Three specific patterns have been presented using the TJX largest credit card theft in history as a case study.

9. ACKNOWLEDGMENTS

We thank our shepherd, Eduardo B. Fernandez, for his valuable and insightful comments that led to significant improvement of this paper during the shepherding process. We also thank Ralph Johnson, Linda Rising, Bob Hanmer, and the authors participating in the security group writers' workshop at the PLoP'09 conference, including Eduardo B. Fernandez, Paul Talaga, Geert Monsieur, Young-Su Son, and Jung-sun Kim, for their valuable comments.

10. REFERENCES

[1] www.utdallas.edu/~supakkul/tools/RE-Tools.

[2] The federal information security management act of 2002. 44 U.S.C. §3541, 2002.

[3] C. Alexander. *The timeless way of building*. Oxford University Press, USA, 1979.

[4] A. Bittau, M. Handley, and J. Lackey. The final nail in WEP's coffin. In *IEEE Symposium on Security and Privacy*. Citeseer, 2006.

[5] B. Boehm. *Software Risk Management*. IEEE Computer Society Press, 1989.

[6] T. Bradley, A. Chuvakin, A. Elberg, and B. Koerner. PCI Compliance: Understand and Implement Effective PCI Data Security Standard Compliance. 2007.

[7] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Sytems*, 8:203–236, 2004.

[8] B. Cheng, L. Campbell, and R. Wassermann. Using security patterns to model and analyze security requirements. In *Proc. Requirements for High Assurance Systems Workshop (RHAS 03)*, 2003.

[9] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.

[10] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50, 1993.

[11] E. Fernandez. Shepherding comments. Personal communication, Jun. 2009.

[12] E. Fernandez and R. Pan. A pattern language for security models. In *Conference on Pattern Languages of Programs (PloP 2001)*. Citeseer, 2001.

[13] E. Fernandez, J. Pelaez, and M. Larrondo-Petrie. Attack patterns: A new forensic and design tool. *International Federation for Information Processing-Publications-IFIP*, 242:345, 2007.

[14] E. Fernandez and X. Yuan. Securing analysis patterns. In *Proceedings of the 45th annual southeast regional conference*, page 293. ACM, 2007.

[15] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes in Computer Science*, pages 1–24, 2001.

[16] M. Fowler. *Analysis Patterns: reusable object models*. Addison-Wesley, 2000.

[17] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.

[18] S. Gaudin. Banks Hit TJ Maxx Owner With Class-Action Lawsuit. *InformationWeek, Apr*, 25, 2007.

[19] P. Giorgini, H. Mouratidis, and N. Zannone. Modelling Security and Trust with Secure Tropos. *Integrating Security and Software Engineering: Advances and Future Vision*, pages 160–189, 2006.

[20] D. Gross and E. Yu. From non-functional requirements to design through patterns. *Requirements Engineering Journal*, 6(1):18–36, 2001.

[21] S. Hilley. *Computer Fraud and Security*, Apr. 2007.

[22] S. Hilley. *Computer Fraud and Security*, Jan. 2008.

[23] M. Jackson. *Problem frames: analyzing and structuring software development problems*. Addison-Wesley, 2000.

[24] M. Johnston. Known vulnerabilities are no. 1 hack exploit. *CNN*, Dec. 17 1999. `http://archives.cnn.com/1999/TECH/computing/12/17/hack.exploit.idg/index.html`.

[25] L. Lin, B. Nuseibeh, D. Ince, and M. Jackson. Using abuse frames to bound the scope of security problems. In *12th IEEE International Requirements Engineering Conference, 2004. Proceedings*, pages 354–355, 2004.

[26] M. Markus, E. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad. *Security Patterns: Integrating Security and System Engineering*. John Wiley & Sons, 2006.

[27] H. Mouratidis, P. Giorgini, M. Schumacher, and M. Manson. Security patterns for agent systems. In *Proceedings of the eighth european conference on pattern languages of programs (EuroPLoP'03)*. Citeseer, 2003.

[28] Office Of The Privacy Commissioner Of Canada And Office Of The Information And Privacy Commissioner Of Alberta. Report of an Investigation into the Security, Collection and Retention of Personal Information of TJX Companies Inc. and Winners Merchant International L.P., Sep. 2007.

[29] T. Okubo and H. Tanaka. Identifying security aspects in Early development stages. In *Proc. 3rd International Conference on Availability, Reliability and Security (ARES'08), Barcelona, Spain*, pages 1148–1155, 2008.

[30] Payment Card Industry. Data security standard v1.1, Sep. 2006.

[31] J. Pelaez, E. Fernandez, and M. Larrondo-Petrie. Misuse patterns in VoIP. *Wiley's Security and Communication Networks Journal*, 2009.

[32] J. Pereira. Breaking the code: How credit-card data went out wireless door. *The Wall Street Journal*, May 4 2007.

[33] T. Priebe, E. Fernandez, J. Mehlau, and G. Pernul. A pattern system for access control. In *Research directions in data and applications security XVIII: IFIP TC11/WG11. 3 eighteenth annual Conference on Data and Applications Security, July 25-28, 2004, Sitges, Catalonia, Spain*, page 235. Springer, 2004.

[34] R. Reiter. *On closed world data bases*. Morgan Kaufmann Publishers Inc., 1987.

[35] M. Schumacher. Security patterns and security standards. In *Proceedings of the 7th European Conference on Pattern Languages of Programs (EuroPLoP), July*. Citeseer, 2002.

[36] M. Schumacher. *Security engineering with patterns: origins, theoretical model, and new applications*. Springer-Verlag, 2003.

[37] W. Stirling. *Satisficing Games and Decision Making: with applications to engineering and computer science*. Cambridge Univ Press, 2003.

[38] S. Supakkul and L. Chung. Extending problem frames to deal with stakeholder problems: An agent- and goal-oriented approach. In *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*, pages 389–394, New York, NY, USA, 2009. ACM.

[39] United States District Court of Massachusetts. United States of America v. Albert Gonzalez. 18 U.S.C. §371, Aug. 5 2008.

[40] A. van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proc. 5th Intl. Symp. Requirements Engineering*, pages 249–262, 2001.

[41] Visa Inc. `http://usa.visa.com/download/merchants/cisp-list-of-pcidss-compliant-service-providers.pdf`.

[42] M. Weiss and H. Mouratidis. Selecting Security Patterns that Fulfill Security Requirements. *16th IEEE International Requirements Engineering, 2008. RE'08*, pages 169–172, 2008.

[43] Wikipedia. Brute force attack. `http://en.wikipedia.org/wiki/Brute_force_attack`.

[44] S. Wong. The evolution of wireless security in 802.11 networks: Wep, wpa and 802.11 standards, 2003. `http://www.sans.org/rr/whitepapers/wireless/1109.php`.

[45] J. Yoder and J. Barcalow. Architectural patterns for enabling application security. In *Conference on Pattern Languages of Programs (PLoP 1997)*, 1997.

[46] E. Yu and J. Mylopoulos. Understanding why in software process modelling, analysis, and design. In *Proc. 16th Intl. Conf. on Soft. Eng.*, pages 159–168, May 16-21, 1994.

APPENDIX

A. A LABEL EVALUATION CATALOG

Figure 12 shows a catalog of goal/problem label evaluation under the closed-world assumption [34]. The label is propagated upward based on the label of the offspring goals/problems (e.g. satisficed, denied) and the contribution against the parent goal/problem (e.g. Make/++, Help/+, Hurt/-, Break/++).

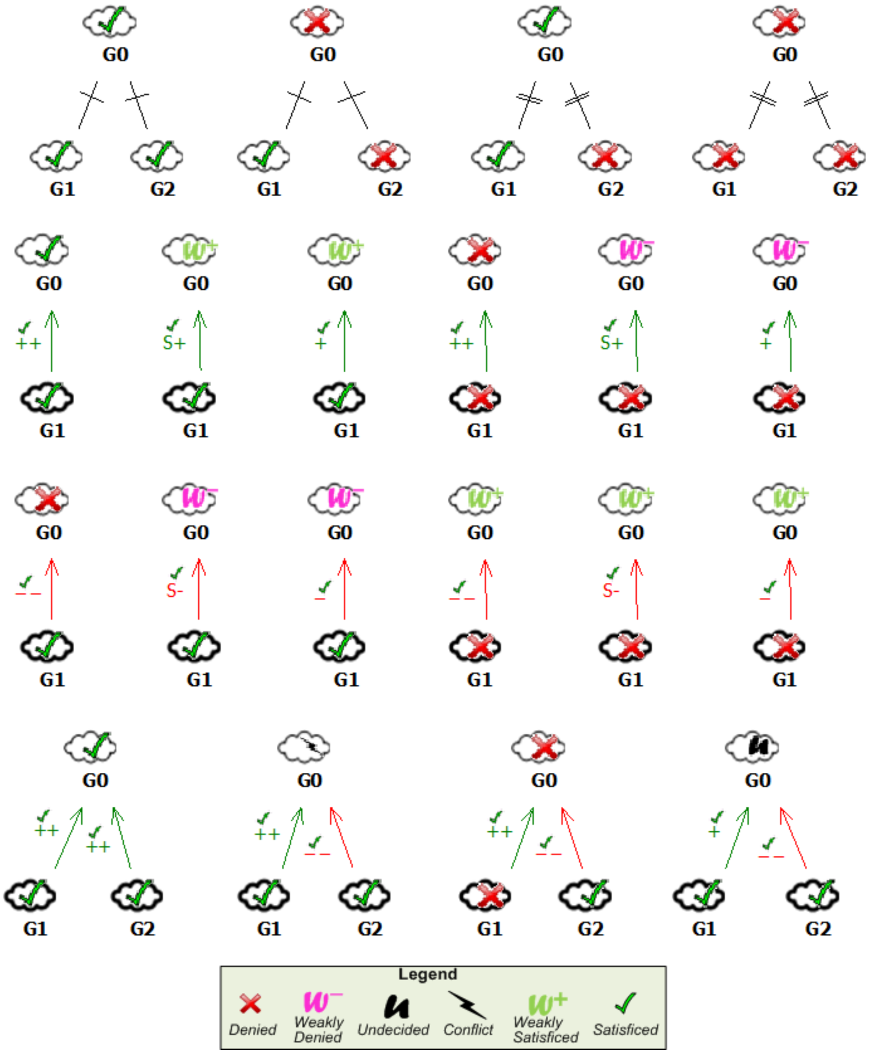

Figure 12: A Goal/Problem Label Evaluation Catalog