UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

DIONATAN DE SOUZA MOURA

DESIGN DE INTERAÇÃO VISANDO SEGURANÇA EM SISTEMAS DE COMPUTAÇÃO

Trabalho de Conclusão de Graduação.

Prof. Dr. Orientador Marcelo Soares Pimenta

Porto Alegre, Dezembro de 2009

AGRADECIMENTOS

Agradeço a todos os professores do curso de Ciência da Computação da Universidade Federal do Rio Grande do Sul, pela disposição e qualidade de ensinar aos alunos. Agradeço também ao time do Google Docs por prover a excelente ferramenta web de escrita que facilitou este trabalho.

SUMÁRIO

LISTA DE FIGURAS

Página

LISTA DE TABELAS

Página

RESUMO

Com o alto crescimento do uso de sistemas de computação para diversos tipos de tarefas, a necessidade de segurança desses sistemas aumentou na mesma proporção. No contexto de software, segurança é a área que estuda diversas formas de proteger tais dados e informações contra o acesso não autorizado, bem como contra ações inseguras de usuários. Embora existam muitos procedimentos que podem ser realizados durante o desenvolvimento para aumentar a segurança do sistema sendo desenvolvido, uma possibilidade nem sempre levada em conta é a melhoria da interação do usuário com o sistema. Neste trabalho discutiremos por que a interação do usuário no sistema é relevante para a segurança do sistema e deve ser considerada no desenvolvimento de software de qualidade, onde segurança é um fator-chave.

Este trabalho descreve os princípios de design de interação visando segurança de sistemas de computação através de fatores e critérios de usabilidade. Cada princípio é descrito e posteriormente exemplificado para um melhor entendimento. Esses princípios formam um conjunto de recomendações que visam auxiliar o processo de design de interação de sistemas com foco nos aspectos relacionados à segurança do software, podendo ser utilizado como um framework para o desenvolvimento de interfaces seguras com o usuário.

Palavras-chaves: Design de Interação, Segurança de Sistemas de Computação, Usabilidade.

Interaction Design Aiming Security In Computer Systems

ABSTRACT

With the high growth in the use of computer systems for various types of tasks, the need for security of these systems has increased proportionately. In the context of software, security is the area that studies various ways to protect data and information against unauthorized access, as well as unsafe user actions. Although there are many procedures that can be performed during development to enhance the security of the system that is being developed, a possibility that is not always taken into account is to improve the user interaction with the system. This work will discuss why the user interaction with the system is relevant to system security and should be considered in quality software development, where security is a key factor.

This work describes the principles of interaction design aiming security of computer systems through the factors and criteria of usability. Each principle is described and illustrated for a better understanding. These principles form a set of recommendations to assist the design process of interaction systems, focusing on aspects related to security software and it also can be used as a framework for the development of secure user interfaces.

Keywords: Interaction Design, Computer Systems Security, Usability.

1. INTRODUÇÃO

Com o expressivo crescimento da taxa de computadores utilizados para diversos tipos de tarefas nas últimas décadas, a demanda de desenvolvimento de software cresceu juntamente na mesma razão. Uma grande parte dessas tarefas que cresceu expressivamente é relacionada a serviços baseados na Internet (Holmström, 1999). Com essa alta demanda, uma quantidade de software significativa foi - e ainda é - desenvolvida com curtos prazos e baixa qualidade. Diversos problemas consequentes dessa baixa qualidade fizeram o desenvolvimento de software preocupar-se com a qualidade criando a área de Engenharia de Software. A área de Engenharia de Software preocupa-se com a qualidade durante todo o processo de desenvolvimento do software.

Diversas tarefas realizadas por software necessitam que se haja proteção e sigilo dos dados e das informações contidas nos sistemas que as executam. Essa necessidade criou o ramo de segurança de software, no qual estuda diversas formas de proteger tais dados e informações contra o acesso não autorizado dos mesmos. Esse ramo estuda a segurança de software em nível de sistema, implementando o software corretamente no quesito de segurança, com criptografia dos dados e autenticação do sistema.

Um sistema baseado em software é basicamente dividido em duas partes; sistema e interface com o usuário. O sistema é responsável por prover funções para realizar as tarefas do escopo e necessidades do usuário em que foi implementado. A interface é responsável pela interação do usuário com o sistema. Um grande problema no desenvolvimento de software é o de que a segurança é implementada apenas na parte do sistema, esquecendo-se da segurança na interface do sistema com o usuário, para que o usuário realize apenas tarefas seguras e não leve o sistema a um estado inseguro.

A área multidisciplinar que trata da interação entre pessoas e computadores é a IHC (Interação Homem-Computador). Baseada em IHC, Design de Interação é a área que pode ser utilizada no desenvolvimento de software para tratar da qualidade de interfaces, facilitando a interação com o usuário através de características de usabilidade. De acordo com (PREECE, 2005), Design de Interação é entendido como "design de produtos que fornecem suporte às atividades cotidianas das pessoas, seja no lar ou no trabalho". Design de Interação tem como base diversas áreas do conhecimento humano, áreas que lidam com sistemas mecânicos, sistemas eletrônicos, sistemas computacionais, áreas de design, e com diversas áreas humanas. Design de Interação é uma área nova do conhecimento que aborda o uso. Pode-se também dizer que é a intersecção entre o escopo do sistema e o escopo do usuário, ou seja, é o uso do sistema pelo o usuário.

Este trabalho objetiva solucionar os problemas de segurança na interação do usuário com o sistema, aprimorando a interface através das características de Design de Interação adaptadas ao escopo de segurança. O framework dos fatores e critérios de usabilidade descrito em (ABOWD, 1992) será utilizado para detalhar os princípios de interação segura descritos em (YEE, 2002). Cada princípio de interação segura será apresentado através dos critérios de usabilidade que se relacionam diretamente com tal princípio.

Em contraponto a (1993, BASTIEN & SCAPIN), os fatores e critérios de (ABOWD, 1992) foram escolhidos para detalhar os princípios de interação segura por terem um enfoque de qualidade no desenvolvimento de software, já que em (1993, BASTIEN & SCAPIN) tem-se um enfoque maior para o usuário e menor para o desenvolvimento de software.

No capítulo 3, serão descritos e exemplificados com detalhes os problemas de segurança relacionados às interfaces. Também serão descritas as abordagens de tratar tais problemas. Já no capítulo 4, serão descritos os fatores de usabilidade e seus critérios, para a partir disso detalhar e exemplificar os princípios de interação segura.

2. INTERAÇÃO INSEGURA, SUAS CONSEQUÊNCIAS E SOLUÇÕES

Interação insegura é o resultado de desenvolver interfaces e interações com o usuário sem considerar características de usabilidade relacionadas à segurança. As conseqüências de uma interação insegura variam desde um arquivo importante excluído até catástrofes nucleares, tal como ocorrida em Three Mile Island nos Estados Unidos em 1979 (U.S.NRC, 2009). Soluções para interação insegura são baseadas no estudo de relacionar características de design de interação e usabilidade com características de segurança e aplicá-las no desenvolvimento de software.

2.1. Potenciais Problemas de Interação Insegura

A segurança de um sistema de computação utilizado por um usuário é diretamente dependente da interação do mesmo. Mensagens ao usuário, alertas de segurança, perguntas para decisões, todo tipo de interação entre o usuário e o sistema pode ser válido em questões de segurança. Em diversos programas e sistemas operacionais – sistemas de computação, a interação com o usuário não é bem cuidada em aspectos de segurança, pois muitas vezes os problemas de segurança são relacionados à especificação, ao projeto e à codificação do sistema, e não da interface. Alguns exemplos desses problemas são estouro de buffer, uso de algoritmo de criptografia fraco, e proteção contra invasão por um atacante.

Acreditar que aumentar a segurança diminui a usabilidade, e vice-versa, é um erro cometido comumente em desenvolvimento de software, segurança e usabilidade não são mutuamente exclusivas. É possível moldar um sistema seguro que seja fácil de usar a partir

de recomendações de usabilidade e recomendações de segurança, sem desmerecer os dois lados.

Um exemplo do sistema correto - e não seguro - é o correio eletrônico. O sistema de correio eletrônico comporta-se de acordo com a sua implementação, enviando e recebendo arquivos corretamente. Porém, o usuário espera que os arquivos recebidos sejam de acordo com o que ele espera, por exemplo, uma foto. Muitas vezes, em mensagens de s*pam* o usuário recebe um arquivo *malware* disfarçado de uma foto, no qual o sistema de correio eletrônico comportou-se adequadamente com a sua função, porém o comportamento do sistema não é o que o usuário deseja. Visto isso, o uso correto do software é tão importante quanto o software estar correto.

Outro bom exemplo em que a usabilidade afeta totalmente a segurança é sobre os resultados de uma análise do programa de criptografia PGP 5.0, contida em (WHITTEN, 1999). A interface do PGP 5.0 confunde o usuário e o faz utilizá-lo incorretamente. Um dos problemas constatados a partir da pesquisa contida nessa análise é o de vazamento de confidencialidade. Entre vinte participantes que utilizaram o PGP 5.0 com uma mensagem secreta ao seu time de membros, três participantes acidentalmente enviaram pelo correio eletrônico a mensagem secreta sem criptografia, ou seja, a mensagem secreta estava vulnerável a ser descoberta facilmente. Mesmo com o melhor algoritmo de criptografia disponível no PGP, a segurança do usuário estava vulnerável por causa da interação dele com o programa.

2.2. Abordagens para Interação Segura

Por serem duas áreas distintas, usabilidade e segurança raramente são relacionadas em formações de desenvolvimento de software. Em geral, formações de usabilidade não dão a importância devida à interação em segurança, mas sim, por exemplo, à facilidade de uso. Geralmente em formações de segurança, mal se toca no assunto de usabilidade e interfaces seguras, e sim, por exemplo, a algoritmos de criptografia seguros e prevenção de ataques ao sistema.

Os estudos de usabilidade e de design de interação relacionados com segurança baseiam-se em princípios de segurança. Uma boa descrição de princípios de segurança pode ser encontrada em (SALTZER, 1975). O guia de segurança de software em (NASA, 2004) contém princípios de design de interação e segurança descritos de um modo bastante implícito e entrelaçado no que se deve fazer criar uma interface de software segura. Por descrever boas práticas, acaba se esquecendo de muitos componentes diferentes que a interface pode ter. Se tais princípios fossem explicitamente descritos e facilmente exemplificados, teria uma abrangência mais completa para todos os tipos de componentes da interface com o usuário.

Baseada em princípios de segurança e design de interação, uma solução para interfaces seguras é dada em (YEE, 2002). Tal solução lista dez princípios que auxiliam o design de interação de interfaces seguras no desenvolvimento de software. Tais princípios são listados de uma maneira indireta e textual. O próximo capítulo desse texto aprimora a explanação desses princípios de uma maneira mais clara, a partir dos fatores e critérios de usabilidade encontrados em (ABOWD, 1992). Desse modo, tem-se uma lista de princípios de design de interação em segurança descritos de uma forma mais clara, tomando como base características de usabilidade. Para cada princípio de design de interação em segurança, é listado pelo menos um exemplo relacionado ao princípio, facilitando o entendimento de cada princípio.

3. DESIGN DE INTERAÇÃO VISANDO SEGURANÇA

Esse capítulo descreve cada um dos dez princípios de design de interação em segurança de sistemas de computação contidos em (YEE, 2002) através dos fatores e critérios de usabilidade contidos em (ABOWD, 1992).

Dentre os vários critérios possíveis para caracterizar usabilidade - nos quais destacam-se os critérios em (NIELSEN, 1994), em (BASTIEN & SCAPIN, 1993), e em (PREECE, 2005) - o conjunto escolhido foi o de (ABOWD, 1992) por 2 razões. A primeira razão é a de que estabelece uma correspondência clara e explícita entre critérios de usabilidade (de IHC) e critérios de qualidade (Engenharia de Software), e por consequência serve como um ponto de partida mais adequado para uma correspondência entre critérios de usabilidade e critérios de segurança, usando os critérios de engenharia de software como intermediários para isto. A segunda razão é a de que tem definições precisas para muitos critérios que os outros trabalhos deixam vagamente definidos.

3.1. Fatores de Usabilidade

De acordo com (ABOWD, 1992), usabilidade pode ser dividida em três fatores. Esses fatores são a aprendizagem, a flexibilidade interação e a robustez de interação. Cada fator tem critérios relacionados auxiliam na sua definição.

O fator de aprendizagem preocupa-se em como o usuário irá utilizar o software a partir da primeira vez até que possa alcançar um bom nível de desempenho no uso do software, realizando a sua tarefa sem dificuldades. Divide-se em quatro principais critérios a seguir:

- Previsibilidade (*Predictability*). Suporte para o usuário determinar o efeito de uma interação futura baseada nas interações passadas.
- Síntese. Suporte para o usuário estimar o efeito das operações realizadas até o momento.
- Familiaridade. Quão parecido é interagir com a realidade ou com outras interfaces, através da interação do sistema dado.
- Generalização (*Generalizability*). Suporte para o usuário ampliar seu conhecimento na aplicação específica através de outras aplicações.

O fator de flexibilidade de interação descreve como o usuário e o sistema podem interagir através da interface entre diferentes maneiras de troca de informações. Divide-se em seis principais critérios a seguir:

- Iniciativa em Diálogo. Permissão de maior liberdade ao usuário através de restrições impostas pelo sistema sobre diálogos de entrada.
- Multitarefabilidade (*Multithreading*). Habilidade do sistema que permite que o usuário interaja com mais de uma tarefa simultaneamente.
- Migrabilidade de Tarefas. Habilidade do sistema que passa o controle de uma tarefa do sistema para o usuário.
- Substituitividade (*Substitutivity*). Permitir valores de entrada e saída semanticamente iguais.
- Multimodalidade (*Multimodality*). Usar múltiplos canais humanos de comunicação para entrada e para saída, tal como pelo teclado, mouse, voz, áudio, vídeo, tato.
- Personalização (*Customizability*). Possibilidade do usuário ou do sistema alterar a interface.

O fator de robustez de interação fornece suporte através da interação com o usuário para que a sua tarefa seja realizada e avaliada. Divide-se em quatro principais critérios a seguir:

- Observabilidade (*Observability*). Habilidade de o usuário avaliar o estado interno do sistema a partir da representação da interface.
- Recuperabilidade (*Recoverability*). Habilidade de o usuário corrigir uma ação quando um erro foi identificado.

- ❖ Tempo de resposta. Como o usuário percebe a taxa de comunicação com o sistema.
- ❖ Conformidade de Tarefas. Habilidade de o sistema suportar todas as tarefas necessárias e desejadas pelo usuário da maneira que ele as compreende.

3.2. Princípios de Design de Interação Segura

Os princípios formam um guia para se obter segurança em interfaces através do design de interação. A validação de cada princípio pode ser provada na forma direta ou indireta. Ou seja, para provarmos ter segurança por completo, implica que temos tal princípio, podemos utilizar a prova direta mostrando como o princípio afeta a segurança, ou podemos utilizar a prova indireta, provando que se não temos tal princípio, que não temos segurança por completo. Em outras palavras, um princípio de segurança pode ser demonstrado através da vulnerabilidade de segurança que a falta do princípio causa.

A seguir serão apresentados os princípios de design de interação visando segurança em sistemas de computação, nos quais descrevem a segurança com a perspectiva do usuário.

3.2.1. Caminho de Menor Resistência

O caminho de menor resistência se aplica na forma de como o usuário realizará a tarefa. Na maioria das vezes, o usuário irá realizar a sua tarefa da forma mais natural pelo caminho de menor resistência.

A característica de **Caminho de Menor Resistência** relaciona-se com os critérios de usabilidade:

- Familiaridade. O caminho natural é semelhante com outras aplicações do mesmo escopo para a mesma tarefa. Geralmente o usuário irá executar sua tarefa a partir do caminho natural das aplicações familiares que já utilizou. Caso não haja esse caminho existente em outras aplicações, e também não haja outro caminho tão natural quanto, o usuário pode ficar frustrado com o software e tentar executar sua tarefa de maneira não segura.

- Substituitividade. A interface deve aceitar entradas que podem ser esperadas pelo usuário para realizar sua tarefa. Números devem ser aceitos além de fórmulas, bem como palavras devem ser aceitas além de expressões regulares. Caso não seja aplicada a substituitividade, o usuário pode não conseguir realizar sua tarefa, ou pode não utilizar um caminho tão seguro para realizar sua tarefa.
- Multimodalidade. Um dos canais de comunicação entre o usuário pode participar na maior parte do caminho de menor resistência, muitas vezes através de um atalho por combinações de teclas, ou através de um ícone de atalho a ser clicado. Tais atalhos devem ser familiares às outras aplicações e devem ser naturalmente memorizados. Os ícones devem ser condizentes com a ação que ele for tomar ao ser clicado, e deve parecer clicável.
- Personalização. O usuário pode desejar personalizar o caminho de menor resistência para que seja mais fácil e mais rapidamente acessado. As principais formas de personalização para facilitar a realização da tarefa são através de uma barra de tarefas personalizada, de menus personalizados, de atalhos personalizados.
- Observabilidade. O usuário pode decidir naturalmente o caminho de menor resistência de acordo com a representação, através interface, do estado interno do sistema.
- Tempo de Resposta. O tempo que se leva para realizar uma tarefa é diretamente proporcional à escolha do caminho de menor resistência. Se o tempo de resposta de uma função do sistema fica instável, o usuário pode escolher outro caminho para realizar sua tarefa. Muitas vezes, tais caminhos são mais arriscados, porém são mais fáceis. O tempo de resposta deve ser estável, em caso de instabilidade, o sistema deve avisar o usuário. Caso o tempo de resposta se torne grande demais, o sistema deve propor ajuda ao usuário e alguma outra forma segura para o usuário realizar sua tarefa.

3.2.1.1 Caminho de menor resistência na exclusão de arquivos

Os sistemas operacionais oferecem um sistema de armazenamento de arquivos excluídos para uma possível recuperação dos mesmos, geralmente denominado de lixeira. No sistema operacional Windows, a lixeira implementa corretamente a sua funcionalidade, mas não

facilita o seu uso, muitas vezes fazendo com que o usuário não a utilize, excluindo os arquivos diretamente.

Nesse caso, o caminho de menor resistência acaba sendo o de excluir os arquivos diretamente - segurando a tecla SHIFT ou desativando a lixeira - sem utilizar a lixeira, o que pode ser muito inseguro no fato do usuário excluir um arquivo sem querer, ou excluir sem saber que o arquivo era importante. Visto isso, o caminho de menor resistência para a tarefa de excluir um arquivo deve ser seguro, a partir da facilidade de uso da lixeira.

As imagens a seguir descrevem as ações tomadas pelo usuário no uso da lixeira.

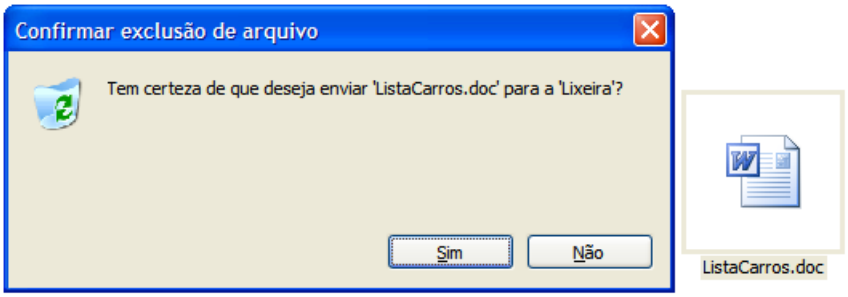

Figura 3.1: Excluindo arquivo, enviando-o para a lixeira.

Na figura 3.1, o desenho da lixeira poderia ser melhor visível se fosse maior, ocupando uma área não utilizada da tela da mensagem.

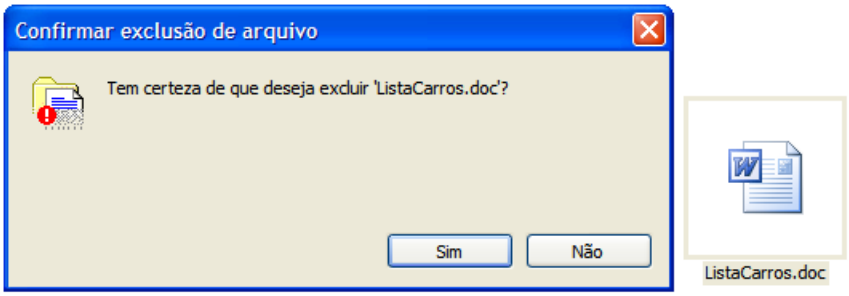

Figura 3.2: Excluindo arquivo definitivamente, não enviando para lixeira.

Na figura 3.2, o desenho de documento e pasta sendo excluídos também é pequeno, assim como na figura 3.1. O nome do arquivo ou pasta a ser excluído é embutido na mensagem, o que dificulta a visualização do nome do arquivo.

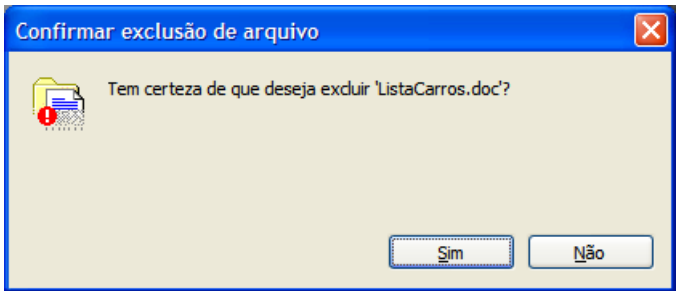

Figura 3.3: Esvaziando a lixeira, na qual contém um arquivo.

A mensagem do caso da figura 3.3 é idêntica à mensagem do caso da figura 3.2, o que pode gerar confusão ao usuário.

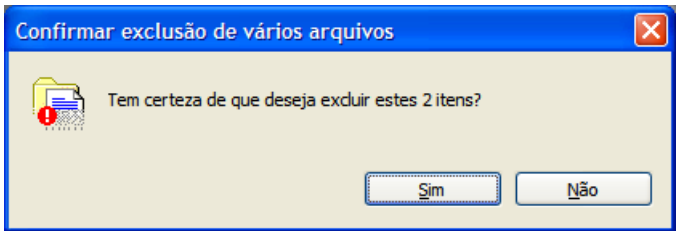

Figura 3.4: Esvaziando a lixeira, na qual contendo dois itens (arquivos ou pastas).

Na figura 3.4, diz que vai se excluir vários arquivos, o que dá uma impressão de exclusão de muitos arquivos, porém são apenas dois itens a ser excluídos.

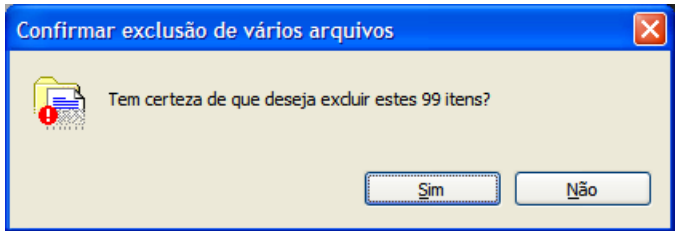

Figura 3.5: Esvaziando a lixeira, na qual contendo 99 itens.

As mensagens para exclusão de 2 a 99 itens referenciam os itens a serem excluídos, porém não os exibe, gerando dúvida em quais arquivos realmente estão na lixeira para serem excluídos.

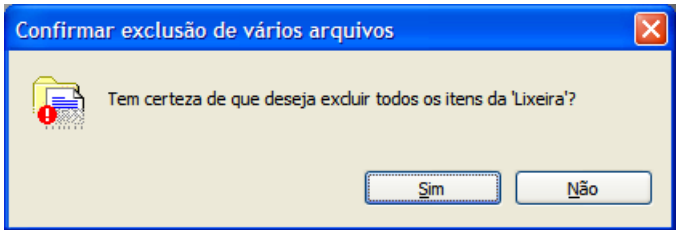

Figura 3.6: Esvaziando a lixeira, na qual contendo 100 ou mais arquivos.

No caso da figura 3.6, a lógica da mensagem é alterada, pois a partir de 100 itens na lixeira não é mais exibida a quantidade de arquivos e pastas contidos na lixeira. Essa mensagem é incoerente, porque a quantidade de arquivos e pastas contidas na lixeira continua sendo importante para a decisão do usuário.

Figura 3.7: Lixeira vazia, sem arquivos.

Figura 3.8: Lixeira não-vazia, com arquivos.

Descrito na figura 3.8, o ícone da lixeira contendo itens gera dúvida ao usuário, de quão cheia está a lixeira. A quantidade de itens na lixeira e a capacidade utilizada da lixeira deveriam ser exibidos juntamente ao ícone, deixando assim o usuário com um maior controle e uma melhor observabilidade do sistema da lixeira.

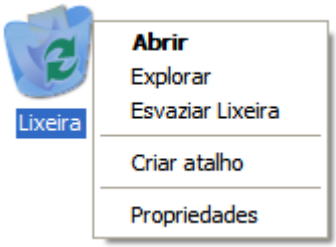

Figura 3.9: Opções ao clicar com o botão direito do mouse na lixeira.

Os itens de menu da figura 3.9 contém um problema de interface segura. No qual é gerado pela distância da opção de Esvaziar Lixeira, que está ao lado da Opção de Explorar sem distinções, o que permite facilmente um clique errado em Esvaziar Lixeira quando o usuário quer realmente clicar em Explorar. Isso pode se tornar um risco aos arquivos armazenados na lixeira. Além desse problema, a opção de Restaurar arquivos da lixeira não existe nesse menu.

Uma proposta segura e usável desse menu da lixeira, no qual teria a opção de esvaziar lixeira separada das demais, seria:

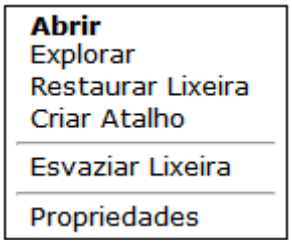

Figura 3.10: Uma forma mais segura de agrupar os itens de menu da lixeira.

Visto tais problemas contidos na interação com a lixeira, o caminho de menor resistência dos usuários muitas vezes acaba sendo a exclusão direta do arquivo, sem a utilização da lixeira.

3.2.2. Limites Adequados (*Appropriate Boundaries*)

As diferenças entre objetos e ações devem ser expostas pela interface e reforçadas pelo sistema. As permissões dos atores do sistema devem estar limitadas e definidas adequadamente para executar apenas ações sobre objetos de um modo seguro.

Tudo que compõe a interação é considerado objeto. Existem dois tipos de objetos; o tipo inanimado e o tipo ator. Um objeto do tipo inanimado sofre ação de um ator, e um objeto do tipo ator realiza a ação sobre um objeto inanimado. Tais permissões de ações devem ser limitadas adequadamente de uma forma segura para cada ator do sistema.

A característica de **Limites Adequados** relaciona-se com os critérios de usabilidade:

- Previsibilidade. As ações possíveis de se executar devem ser compatíveis e não gerar surpresa ao usuário, estando de acordo com suas ações passadas. Um contra-exemplo é, após a interação do usuário de excluir um arquivo e sempre perguntar se deseja mesmo excluir tal arquivo, o usuário excluir uma pasta e o sistema não perguntar se deseja excluir tal pasta, o usuário irá sentir que algo de errado aconteceu.
- Familiaridade. Os limites adequados de permissões devem ser similares aos de outras aplicações, desde que tais aplicações sejam seguras. O usuário deve

facilmente reconhecer quais permissões ele e outros atores tem em relação aos objetos, e a familiaridade com outras interfaces auxiliam nesse reconhecimento.

- Multitarefabilidade. Nesse caso, podem existir ações que proíbem ser acessadas se outras ações estiverem sendo realizadas, ou seja, ferindo totalmente o conceito de multitarefabilidade. Existem também objetos que proíbem sofrer mais de uma ação simultaneamente, por exemplo, se uma conta corrente for vista como um objeto, não se pode realizar mais de uma operação ao mesmo tempo para segurar que a transação seja completa, e que ninguém mais está acessando-a ao mesmo tempo.
- Observabilidade. As permissões, nas quais estão sendo obrigados pelo estado interno do sistema, devem ser facilmente representados através da interface. Esse fator de usabilidade define a característica dos limites adequados, pois devem a sua visualização deve ser reforçada pela observabilidade da interface.
- Conformidade de Tarefas. Relacionado à suportar todas as tarefas necessárias para o usuário completar sua tarefa. Conformidade de tarefas é mais amplo que limites adequados, pois é a junção de limites adequados e habilidade esperada, que são características de design de interação segura fortemente relacionadas entre si.

3.2.2.1. Limites adequados no Internet Explorer 7 com Windows Vista

Diversas páginas da Internet tentam atacar o computador do usuário através de códigos maliciosos. O que o navegador Internet Explorer 7 rodando no Windows Vista faz, é utilizar o princípio do menor privilégio (1975, SALTZER), dando o menor conjunto de permissões e privilégios, e através de autorizações explícitas fazer tais permissões se ajustar de acordo com a necessidade do usuário. Esse sistema, baseado no princípio do menor privilégio é um componente integrado ao Windows Vista, é denominado de UAC (User Access Control - Controle de Acesso do Usuário). Em teoria, isso torna a navegação mais segura, confiável, e fácil de usar após algum tempo de uso do software.

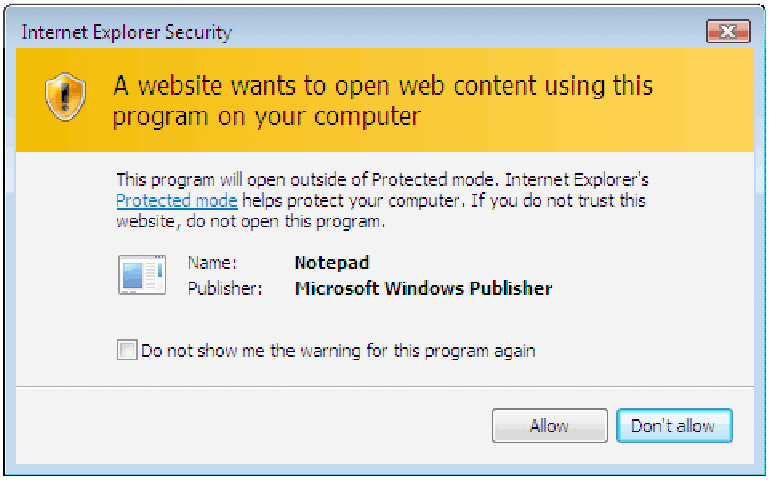

Figura 3.11: Internet Explorer requisita de forma errada autorização ao usuário.

Há privilégios que não precisam ser requisitados ao usuário, pois o usuário já requereu implicitamente tal ação a ser executada. Um exemplo é quando o usuário abre o código-fonte da página, o Internet Explorer por padrão utiliza o Notepad para exibir o texto do código. Então o Internet Explorer pergunta ao usuário se ele autoriza que a página abra o seu conteúdo pelo Notepad, o que é um erro de conformidade da tarefa, pois foi o usuário quem requisitou a abertura do conteúdo da página.

3.2.3. Autorização Explícita

Uma permissão que não esteja no conjunto de permissões de acodo com os limites adequados, só deve ser provida para outros atores pela ação explícita do usuário. Tal ação explícita ao usuário deve ser facilmente entendida que seja de concessão de permissão. O princípio da Autorização Explícita relaciona-se diretamente com o critério de usabilidade de Iniciativa em Diálogo.

A característica de **Autorização Explícita** relaciona-se com os critérios de usabilidade:

- Previsibilidade. O usuário espera que, no momento de algum objeto realizar uma ação relevante à segurança, seja requisitada a ele a autorização do objeto realizar tal ação. Por exemplo, na situação que o usuário deseja realmente que o sistema que ele está utilizando exclua um arquivo.

- Familiaridade. A solicitação de autorização deve ser semelhante às solicitações de outras aplicações. Assume-se que tais aplicações familiares façam solicitações de um modo seguro e bom excelente entendimento. Caso não façam de um modo seguro, a interface deverá ignorar esse conceito de familiaridade. A disposição das confirmações na interface deve sempre ser a mesma. Se o botão de confirmação positiva estiver sempre à esquerda nas aplicações familiares, o botão na interface referida deve também estar sempre à esquerda, não havendo assim confusão e uma possível confirmação errada do usuário, na qual pode comprometer a segurança do usuário.
- Iniciativa em Diálogo. Autorização explícita é a instância da característica de iniciativa em diálogo no escopo de segurança de sistemas. Autorização explícita dá maior flexibilidade ao usuário poder realizar ações que podem afetar a segurança, mas ficando com responsabilidade do usuário de definir se é seguro ou não tais ações.
- Migrabilidade de Tarefas. O sistema passa o controle de decidir se a ação é segura ou não para realizar a tarefa através da autorização explícita, e o usuário avalia se a tarefa é segura, autorizando ou não dependendo da sua avaliação.
- Multimodalidade. O diálogo de confirmação para o usuário autorizar ou não outro ator, pode ter diversas formas de confirmação, através do teclado, mouse, temporalidade, entre outros. O meio de confirmação positiva pelo teclado é comumente a tecla ENTER, e a tecla ESC para confirmação negativa. Para casos críticos de segurança, devem ser utilizadas combinações de teclas com SHIFT, ALT e/ou CONTROL e alguma tecla comum, evitando que o usuário confirme sem querer ou que autorize diversas confirmações sem ler o que é a autorização, e acabe autorizando um caso crítico com descuido. O foco da interface deve estar no botão que é o caso mais seguro para o usuário, em casos críticos de segurança muitas vezes é o botão "Não". O meio de confirmação pelo mouse é mais simples que os outros, porém deve-se ter o cuidado de colocar as opções de autorização brevemente distante das outra, para que o usuário não selecione a opção errada ao clicar. A temporalidade da confirmação pode ser utilizada, desde que a segurança não seja comprometida. Tal temporalidade

utiliza um contador regressivo para escolher a opção que é a mais comum a ser escolhida, no final da contagem a opção é escolhida automaticamente. A temporalidade é aconselhada a ser utilizada em aplicações que não afetem a segurança do usuário e que precisem de uma confirmação imediata.

- Personalização. A interface pode oferecer multiplicidade na opção do sistema armazenar as permissões de ações para os atores. A multiplicidade de opções depende da finalidade do sistema, em geral a multiplicidade dá as opções de dar permissão para o ator executar a ação apenas uma vez e no momento dado, para o ator executar a ação apenas na sessão aberta da aplicação, para o ator executar a ação na execução atual e em todas as outras execuções do sistema, e para o ator não executar a ação.
- Observabilidade. A interface deve mostrar uma visualização interna do sistema, graficamente ou através da mensagem da autorização explícita, para que o usuário possa decidir se a autorização não afeta a segurança.
- Recuperabilidade. A recuperabilidade de uma permissão dada pelo usuário por uma autorização explícita ocorre através da revogabilidade.
- Conformidade de Tarefas. Os requerimentos de autorizações ao usuário devem estar em conformidade com a tarefa que o usuário deseja realizar. O princípio de menor privilégio deve ser considerado, porém deve-se levar em consideração também as ações tomadas pelo usuário, nas quais autorizam implicitamente permissões a atores.

3.2.3.1. Autorização implícita ao abrir arquivos

Ao abrir um documento num programa através de um navegador de arquivos, o usuário já está autorizando implicitamente que o programa abra o arquivo ao selecionar o arquivo e pressionando o botão de abrir do navegador de arquivos. Uma autorização explícita para abrir o arquivo selecionado não é necessária quando o usuário seleciona o arquivo a ser aberto na aplicação. Esse caso é ilustrado na figura a seguir.

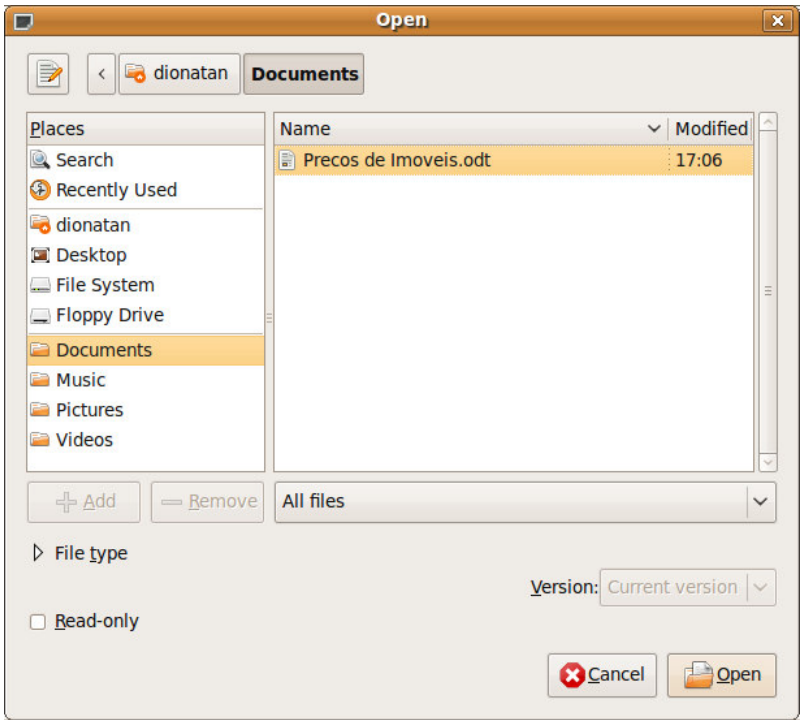

Figura 3.12: Interface de escolha de arquivo para abertura no Ubuntu.

3.2.3.2 UAC do Windows Vista e Windows 7

O Windows Vista e o Windows 7 tem como base de segurança para o usuário o UAC (User Account Control - Controle de Contas de Usuário), que é um componente de segurança presente no Windows Vista. O UAC fornece inicialmente privilégios limitados, então se faz necessário que o usuário autorize a concessão de privilégios ao programa através de autorização explícita. Tais autorizações explícitas são feitas quando, por exemplo, se deseja rodar uma aplicação como um administrador, alterar configurações ou arquivos na pasta de instalação do Windows ou na pasta de arquivos de programas, instalar ou desinstalar aplicações, rodar o Agendador de Tarefas, etc.

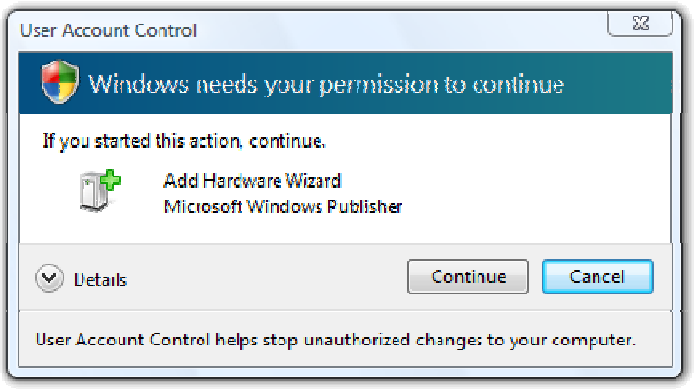

Figura 3.13: Autorização via UAC. O UAC pede a autorização explícita ao usuário para que se abra o assistente de adicionar hardware.

No Windows 7 o UAC pode ser personalizado com 4 níveis de segurança. Tais níveis são para deixar o usuário escolher o que é melhor para a segurança, em quatro escalas de sempre ser notificado até nunca ser notificado. Essa personalização foi criada após muitos usuários desativarem por completo o UAC no Windows Vista, por ficarem aborrecidos com tantas mensagens de autorização explícitas.

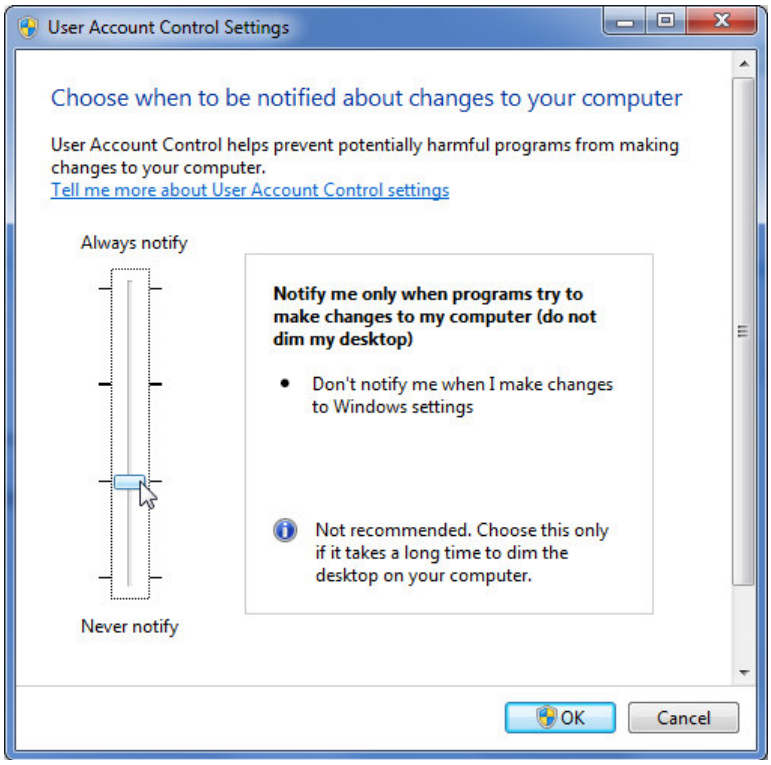

Figura 3.14: Escolhendo níveis de segurança em autorizações explícitas do UAC.

3.2.3.3 Autorização explícita para substituir arquivos

Os sistemas operacionais mais utilizados implementam um sistema de cópia de arquivos e pastas. Ao detectar que o diretório já contém tais arquivos e pastas com os mesmos nomes, é requisitada uma autorização para substituir tais itens. Tal substituição é implementada de uma maneira fracamente segura, pois não há opção dos arquivos substituídos serem colocados na lixeira.

No sistema operacional Windows XP, a mensagem para substituir arquivos ou pastas é mostrada na imagem a seguir.

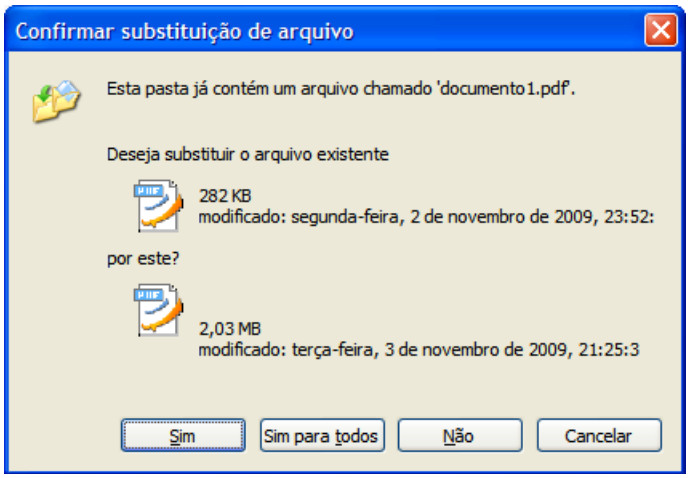

Figura 3.15: Mensagem de confirmação para substituir arquivos.

Nessa mensagem da figura 3.15, não há opção de não copiar apenas os arquivos que o diretório de destino não contém (poderia ser um botão denominado "Não para todos"), o que facilitaria bastante na tarefa de copiar os arquivos não existentes no diretório de destino. Essa facilidade é escondida do usuário, para realizá-la deve-se segurar a tecla SHIFT ao clicar no botão "Não".

Não há o conceito seguro de previsibilidade, pois não é provida a opção de visualizar quais arquivos serão substituídos caso seja selecionada a opção de substituir todos os arquivos (botão Sim para todos). Outro problema de previsibilidade é que, ao clicar em "Sim para todos", as pastas com mesmo nome também serão excluídas, mesmo a mensagem sendo destinada à substituição de arquivos.

3.2.4. Visibilidade

Todos os relacionamentos relevantes à segurança de permissões a autores devem ser facilmente revisados pelo usuário através da interface. Não se pode assumir que o usuário deva se lembrar de todas as autorizações realizadas aos atores e ações do sistema, pois a memória humana é limitada e falha, além de que não se deve sobrecarregar a memória do usuário. Devem ser visíveis apenas as autorizações necessárias para o usuário ter o controle e segurança da aplicação. O usuário não precisa enxergar todas as permissões de todos os componentes, a modo de ter uma visibilidade de baixo nível, sim enxergar apenas as permissões relevantes à segurança. O princípio da Visibilidade relaciona-se diretamente com o critério de usabilidade Síntese.

A característica de **Visibilidade** relaciona-se com os critérios de usabilidade:

- Previsibilidade. O princípio de visibilidade auxilia o usuário a definir se o sistema está num estado seguro e se estará num estado seguro, pois permite analisar que atores do sistema tem quais permissões. A previsivibilidade depende diretamente da visibilidade do momento atual, pois é a partir da visão do estado atual do sistema que o usuário irá fazer decisões e realizar ações seguras.
- Síntese. Analisar as permissões concedidas até o dado momento é o conceito de visibilidade no escopo de permissões. Determinar a segurança do estado do sistema depende diretamente da síntese e da visibilidade do que cada ator do sistema pode ter realizado até o momento.
- Familiaridade. Visibilidade é um princípio de segurança difícil de ser encontrando em aplicações, então provavelmente aplicar o conceito de familiaridade possa ser ignorado, necessitando-se implementar um modelo aprimorado de visão das permissões aos atores do sistema.
- Multitarefabilidade. O usuário deve poder executar suas tarefas paralelamente com o que a interface provê de visibilidade. A interface não deve forçar o usuário a interromper sua tarefa para poder analisar quais permissões os atores do sistema tem. Tal interrupção pode conduzir o usuário a parar de analisar as permissões por aborrecimento desse papo a ser tomado, realizando as tarefas ignorando tais permissões, o que provavelmente irá corromper a segurança.
- Personalização. Adaptar e facilitar a visibilidade do estado do sistema relacionado às permissões devem ser possíveis para que o usuário escolha a melhor forma de analisar a segurança do sistema. A interface pode prover opções de cores, sons, localidades da tela, tamanhos e formas de componentes que irão dar a visão interna das permissões.
- Observabilidade. A interface deve representar o estado interno do que os atores do sistema podem realizar através das permissões para decidir e definir a segurança atual do estado sistema.

- Tempo de resposta. A interface não pode delongar a apresentação das permissões do sistema para apresentar ao usuário, pois dependendo do tempo e necessidade de urgência do usuário a realizar alguma tarefa, o usuário pode acabar realizando tarefas sem aguardar que o sistema disponha da informação, podendo realizar tarefas indevidas e inseguras.
- Conformidade de Tarefas. A visão e a revisão que o usuário tem das permissões através da interface devem ser fidedignas com o estado atual do sistema. Qualquer erro ou atraso na informação pode conduzir o usuário a realizar ações que comprometam toda a segurança da sua tarefa e do sistema.

3.2.4.1 Visibilidade em navegador web para conexões seguras

Boa parte dos navegadores web utiliza o conceito de cadeado para mostrarem que o usuário está numa conexão segura com a página que está sendo visitada. Caso a conexão tenha a criptografia fraca, o cadeado ficará aberto, caso a conexão tenha sua criptografia confiável, o cadeado ficará fechado. O provimento de Visibilidade pelos navegadores web são importantes para que o usuário identifique que está realizando a sua tarefa seguramente. Também é importante pelo motivo de que o navegador implicitamente é autorizado para abrir páginas nas quais contém certificados validados pelas unidades certificadoras conhecidas pelo navegador.

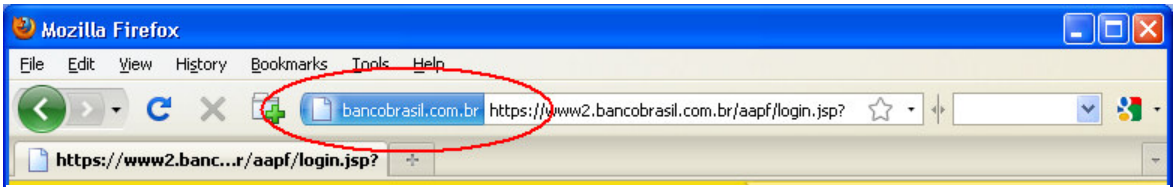

Figura 3.16: Conexão segura no navegador Firefox, barra superior. Caixa colorida (azul ou verde) no Firefox simbolizando que o certificado da página aberta é seguro. O cadeado fica no canto inferior direito da tela, indicado na figura 3.17.

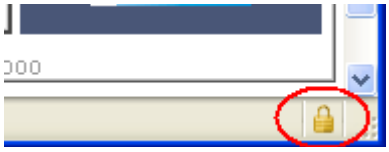

Figura 3.17: Conexão segura no navegador Firefox, barra inferior. Cadeado simbolizando conexão segura, a página é certificada por uma unidade certificadora reconhecida pelo navegador web, e é utilizado algum sistema de criptografia.

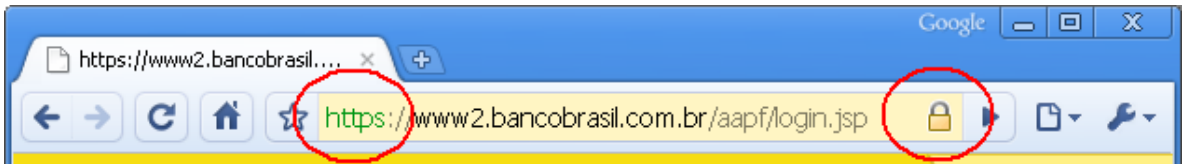

Figura 3.18: Conexão segura no navegador Chrome. O protocolo HTTPS é destacado em verde para simbolizar que a conexão é segura. Ao lado direito do diretório do site contém o cadeado fechado, simbolizando que o certificado é da página é confiável e que a conexão é segura. O cadeado não aparece no canto inferior direito tal como no Firefox. O fundo do link da página é destacado em amarelo fraco para realçar que se está numa conexão segura.

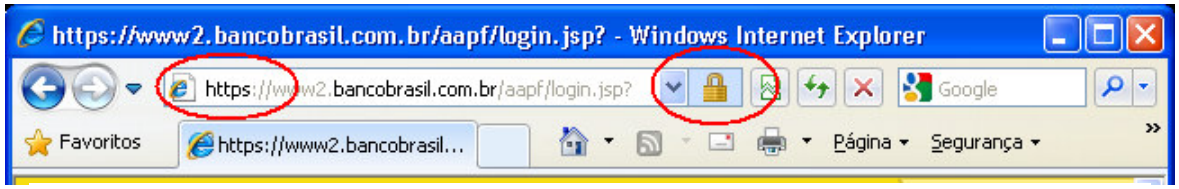

Figura 3.19: Conexão segura no navegador Internet Explorer. O cadeado é exibido num lugar semelhante ao Chrome, porém o fundo do link continua em branco. Não há um destaque no tipo do protocolo utilizado.

Figura 3.20: Certificado inválido no navegador Internet Explorer. O cadeado não é mais exibido, gerando certa confusão. No lugar do cadeado aberto, para contrastar com o cadeado fechado, é exibido um escudo vermelho simbolizando erro e uma descrição "Erro no Certificado" ao lado. O fundo do link nesse momento ficou vermelho, avisando erro. Não há um destaque no tipo do protocolo utilizado.

Figura 3.21: Conexão insegura no navegador Chrome. O protocolo HTTPS é destacado em vermelho com um risco em cima para simbolizar que a conexão não é segura. Ao lado direito do diretório do site contém um símbolo de aviso amarelo com um ponto de exclamação, simbolizando que o certificado da página não é confiável e que a conexão não é segura. O fundo do link da página não é destacado para destacar que se está numa conexão insegura.

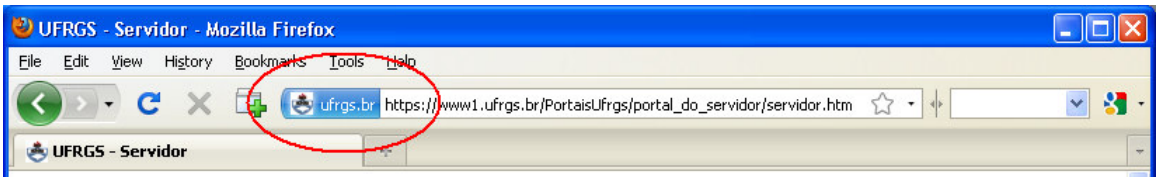

Figura 3.22: Certificado não confiável pelo navegador autorizado pelo usuário. Mesmo que o certificado seja tomado inicialmente como não confiável pelo navegador porque a unidade certificadora é desconhecida, o Firefox permite que se adicione a unidade certificadora como confiável. Isso fará com que não sejam mais dados avisos ao usuário nos certificados e sites que ele confia, facilitando o acesso aos sites que o usuário confia.

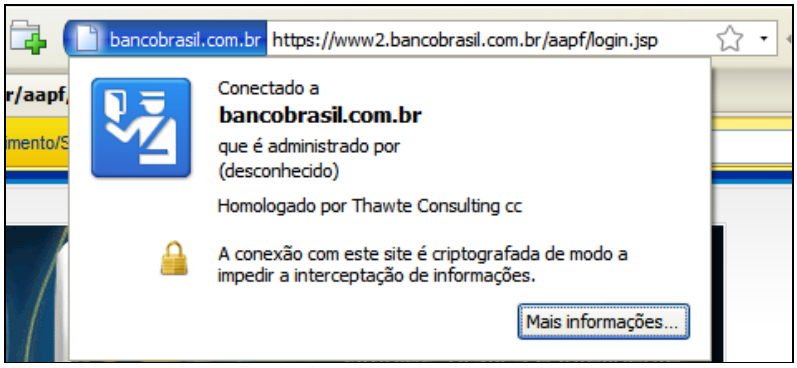

Figura 3.23: Caixa de certificado colorida clicável no Firefox. Além de se poder clicar no cadeado para examinar a segurança da conexão e o certificado, o Firefox ainda permite que a caixa colorida seja clicada para examinar-se o certificado.

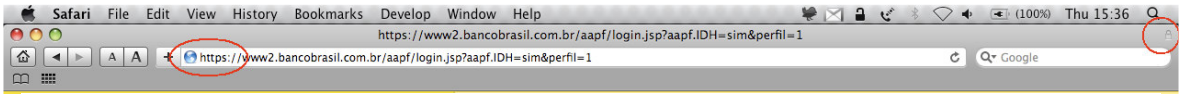

Figura 3.24: Conexão segura no navegador Safari. O navegador Safari exibe um cadeado discreto no canto superior direito, escondendo parcialmente do usuário a visibilidade da permissão do navegador se conectar à página.

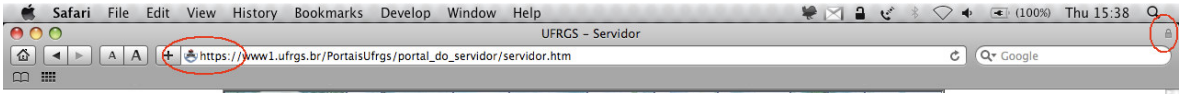

Figura 3.25: Adição de certificado não seguro no navegador Safari. O navegador Safari permite o usuário adicionar confiabilidade nas unidades certificadoras de certificados não reconhecidos como seguros, porém continua ocultando visualmente que se está numa conexão segura.

	Google Chrome	Internet Explorer	Mozzila Firefox	Safari
Realce do protocolo	Sim, HTTPS em verde ou riscado em vermelho	**Não**	Sim, com caixa colorida do estado do certificado	**Não**
Realce clicável do protocolo	**Não**	**Não**	Sim, caixa do certificado	**Não**
Endereço seguro destacado	Sim, com fundo em amarelo	**Não**	**Não**	**Não**
Endereço não seguro destacado	**Não**	Sim, com fundo em vermelho	**Não**	**Não**
Cadeado de segurança facilmente visível	Sim, na barra de endereços	Sim, na barra de endereços	Sim, na barra de status	Semi-oculto

Tabela 3.1. Visibilidade de conexão segura em quatro principais navegadores.

De acordo com a tabela 3.1, os navegadores Google Chrome e Mozzila Firefox são os que mais atendem o princípio de visibilidade na conexão segura. O Safari praticamente não distingue conexão segura de conexão comum, o que pode ser perigoso pelo fato do usuário não identificar se está numa conexão onde seus dados estarão criptografados e protegidos.

3.2.4.2. Processos transparentes em sistemas operacionais

Os sistemas operacionais executam programas através de processos. Um processo é uma instância de execução de um programa. Os sistemas operacionais atuais são multi-tarefas, nos quais executam diversos processos simultaneamente. Os processos por serem considerados como objetos internos do sistema operacional, são escondidos de certa forma do usuário. Visto isso, o usuário não sabe o que está sendo executado no momento, muito menos quais processos tem quais permissões para realizar quais ações.

Com o crescimento significativo da quantidade de usuários que utilizam computadores para realizarem tarefas monetárias através de sites de bancos, a quantidade de programas maliciosos que são executados invisivelmente pelo usuário também

aumentou. Assim gerou-se uma modalidade de sistemas de proteção denominados anti-spywares, no qual analisa os programas contidos no computador e os processos em execução para identificar o que é malicioso para o usuário.

Tal necessidade não seria crescente e grande se os sistemas operacionais executassem os processos de um modo não transparente para o usuário. Os processos que fazem parte do sistema operacional deveriam continuar sendo transparentes, pois não deve interessar às tarefas do usuário. Todos os outros processos que não fazem parte do sistema operacional deveriam ser visíveis de uma forma simples pelo usuário, para que ele veja que processos estão sendo executados e quais as suas permissões de ações sobre quais objetos do sistema.

No Windows XP, o Gerenciador de Tarefas exibe os programas (denominados como aplicativos) e processos que estão sendo executados, no qual sua interface provê pouca informação a respeito de permissões que os aplicativos e processos que estão sendo executados têm.

Figura 3.26: Aplicativos abertos pelo usuário no Windows XP.

Por ser um sistema multi-tarefas, o Windows executa os processos relacionados aos usuários do sistema simultaneamente. A imagem a seguir identifica os processos sendo executados no Windows XP.

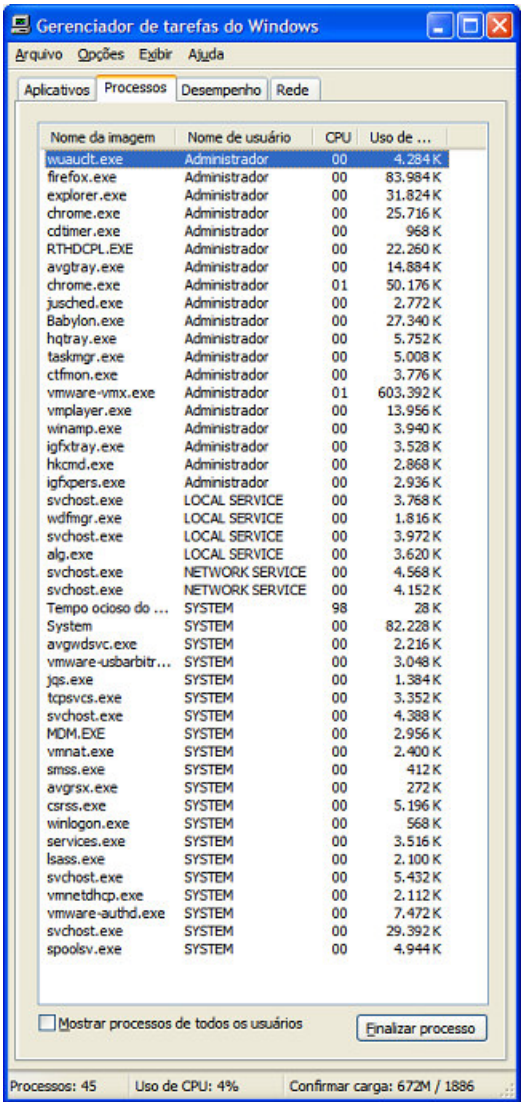

Figura 3.27: Processos sendo executados pelo sistema operacional Windows XP.

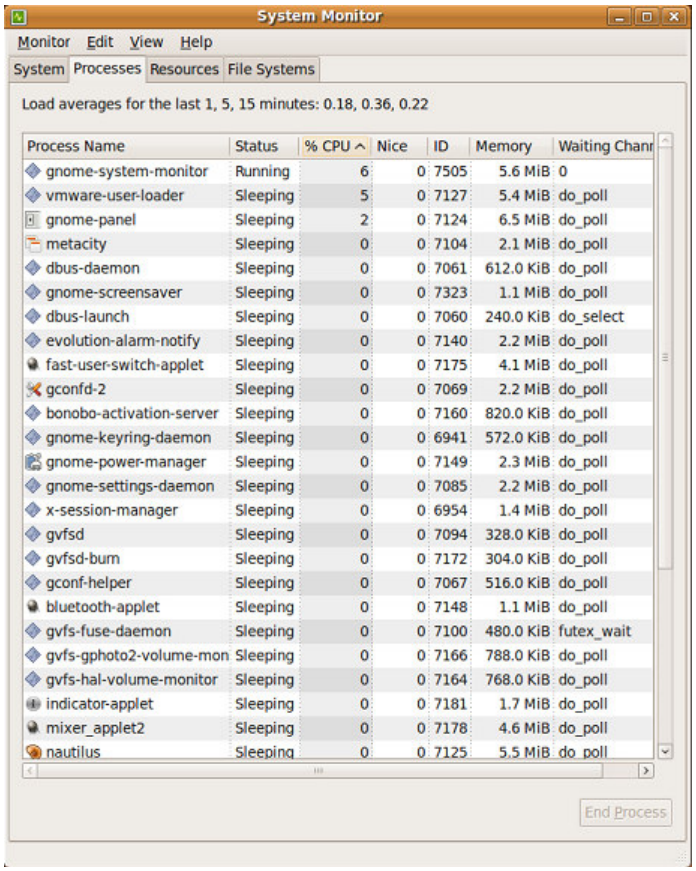

Figura 3.28: Processos sendo executados pelo sistema operacional Ubuntu.

Os sistemas operacionais não utilizam o conceito de visibilidade para processos, o que é perigoso para a segurança do usuário, pois processos podem estar sendo executados em segundo plano no sistema operacional, sem a consciência do que o processo é e executa. Nas figuras anteriores foram exibidos a única maneira de como as interfaces gráficas dos sistemas operacionais Windows XP e Ubuntu exibem os processos: num monitor de processos no qual não provê informações relacionadas a permissões, de um modo muito interligado ao desempenho dos processos, além de estar numa janela que não provê constante monitoramento dos processos.

Por uma tentativa de aprimoramento, o Windows Vista adicionou um campo de descrição a cada processo, de acordo com a figura a seguir.

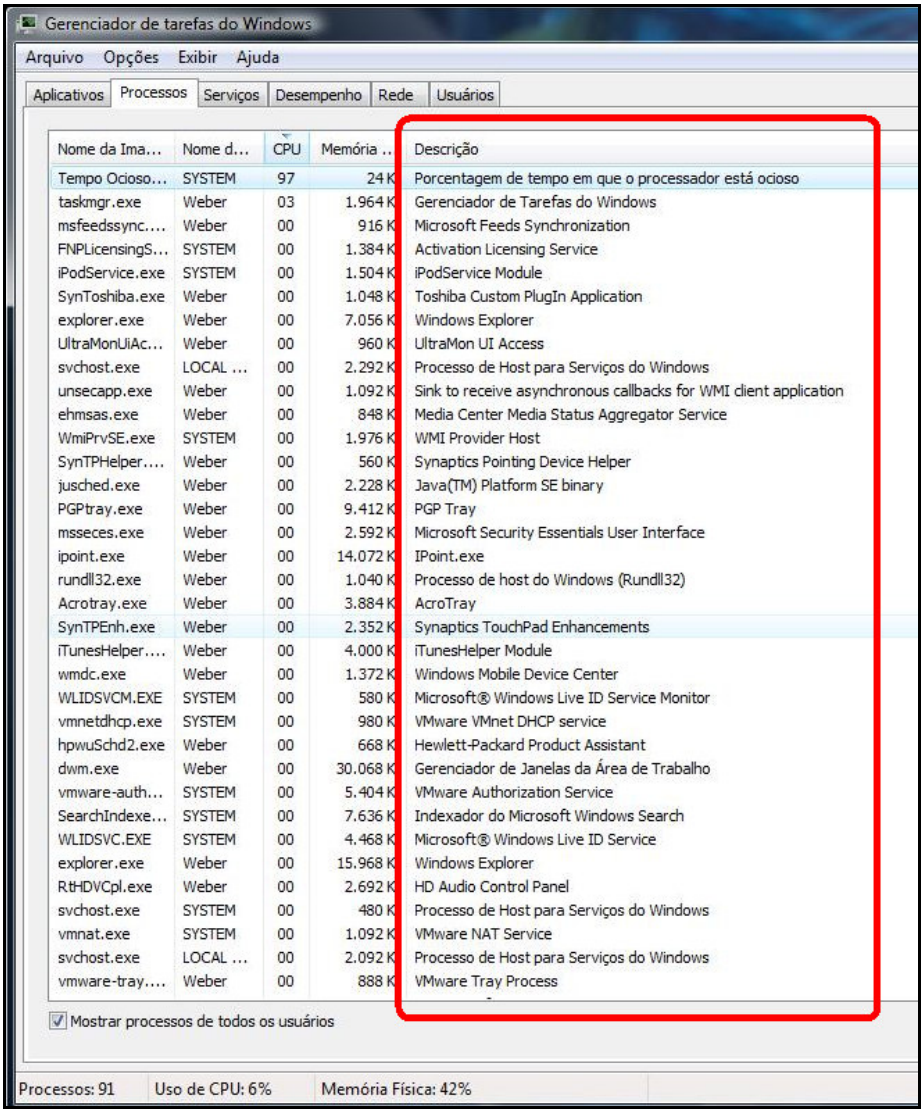

Figura 3.29. Processos sendo executados pelo sistema operacional Windows Vista.

De acordo com a figura 3.29, a descrição de cada processo aprimora o entendimento do usuário do que se trata cada processo. Porém, por ser um campo texto, qualquer processo malicioso pode personalizar sua descrição fazendo com que o usuário seja enganado, como agravante de o processo já estar rodando de um modo quase invisível para o usuário. Muitos programas maliciosos podem se aproveitar desse campo de descrição para induzir o estado de segurança para o usuário.

3.2.5. Revogabilidade

A interface deve permitir que o usuário facilmente anule as permissões de outros atores, dentro do possível. Dar o controle ao usuário sobre as permissões é essencial para que o usuário não deixe que algum ator faça ou continue fazendo ações inseguras. O princípio da Revogabilidade relaciona-se diretamente com o critério de usabilidade de Recuperabilidade.

A característica de **Revogabilidade** relaciona-se com os critérios de usabilidade:

- Previsibilidade. A revogabilidade deve ser possível sempre em ações semelhantes e relacionadas, pois o usuário pode prever a revogabilidade de uma permissão através de outras que ele já confirmou que podem ser revogadas.
- Síntese. O efeito da remoção das permissões através revogabilidade devem ser visto de alguma forma através da interface, podendo ser por uma lista de permissões, ou até mesmo ser por um balão de informação no canto da tela.
- Familiaridade. A interface deve prover revogabilidade de um modo semelhante aos modos que outras aplicações revogam permissões.
- Personalização. O usuário pode desejar que a permissão dada a um ator seja limitada em tempo ou partes. A permissão limitada por tempo é revogada automaticamente após um tempo, no qual o usuário permite que um ator execute uma ação sobre um objeto do sistema durante um tempo personalizado. A permissão é limitada em partes quando se revoga parcialmente a permissão, tendo-se uma permissão dada ao ator mais restringida.
- Observabilidade. O usuário deve perceber a possibilidade de revogabilidade nas permissões dadas aos atores do sistema nas quais podem ser revogadas.
- Recuperabilidade. É o critério de usabilidade diretamente relacionado à Revogabilidade, no qual se pode recuperar o estado seguro no qual não tinha tal permissão a um ator potencialmente malicioso.
- Tempo de resposta. Anular uma permissão deve ser prioritário sobre toda ação do ator no qual contém tal permissão. É importante que o ator tenha a sua

permissão anulada antes que ele tome alguma ação que corrompe a segurança do usuário.

- Conformidade de Tarefas. A Revogabilidade deve estar presente em todas as permissões que são possíveis de ser anuladas. O usuário irá contar com que a interface esteja habilidade para que tarefas possíveis de ser revogadas sejam anuladas: muitas vezes ele dará permissões que ele não sabe isso implicará na segurança do sistema, esperando que a permissão possa ser cancelada caso seja identificada um problema na segurança do sistema relacionado a essa permissão.

3.2.5.1 Anulando a Execução de Processos

Ao identificar que um aplicativo é ou pode ser malicioso, o usuário deveria poder remover a permissão de o processo ser executado, bloqueando o processo ou bloqueando a árvore de processos relativa ao processo. Esse deveria ser o primeiro passo para que o sistema volte a um estado seguro. Porém os sistemas operacionais não dispõem dessa opção, ao menos não pela interface gráfica.

3.2.5.2 Permissões de salvar senhas em navegadores web

Diversas páginas utilizam senhas para identificar o usuário. Os navegadores atuais oferecem a opção de salvar os nomes de usuários e senhas, assim os usuários não precisam digitar o nome de usuário e senha sempre que entrar nessas páginas. Assim como tais navegadores pedem a permissão para salvar senhas através de autorização explícita, também fornecem a opção de que o usuário exclua a senha para que o navegador não a guarde, impossibilitando que outro usuário acesse a página com suas credenciais.

A seguir segue um exemplo do navegador Chrome, que tem um sistema de salvamento de senhas:

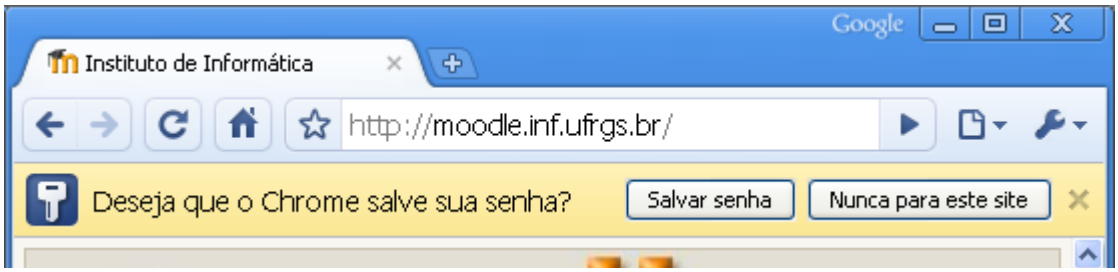

Figura 3.30: Autorização explícita para salvar senha no Chrome. Autorização para que o usuário permita que o Chrome salve a senha recém digitada na página. Selecionando a opção de nunca salvar a senha para esse site, o site vai para uma lista de exceções de perguntar sobre o salvamento de senha.

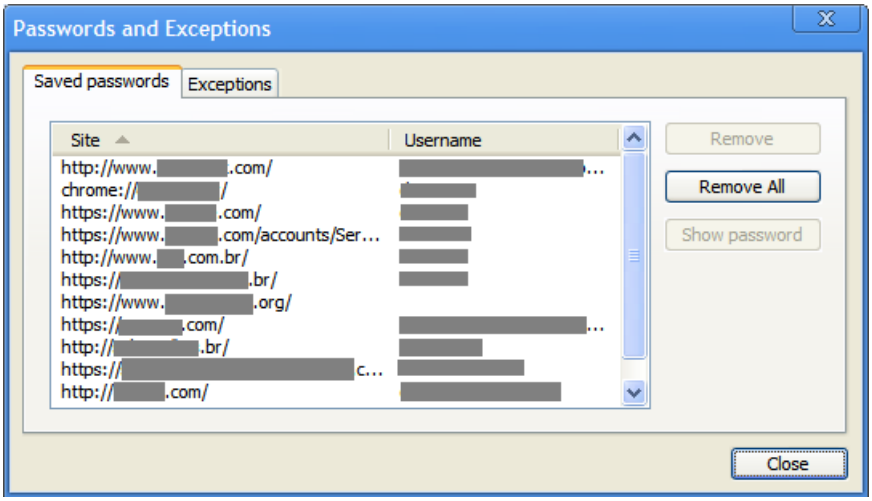

Figura 3.31: Gerência de senhas no Chrome. Interface que permite a gerência de nomes de usuários e senhas salvas, podendo excluí-las. Exceções podem ser adicionadas nessa janela. Essas senhas são armazenadas sem um sistema de proteção, selecionando-se o endereço de uma página, pode-se ver a senha do usuário. Porém é permitido que se exclua a senha, anulando a permissão do navegador de armazenar aquela senha.

3.2.6. Habilidade Esperada (*Expected Ability*)

A interface não deve induzir o usuário de que é possível fazer algo que não pode ser feito, mostrando apenas as habilidades esperadas pelo usuário. O princípio da Habilidade Esperada relaciona-se diretamente com o critério de usabilidade de Conformidade de Tarefas.

A característica de **Habilidade Esperada** relaciona-se com os critérios de usabilidade:

- Previsibilidade. As ações nas quais o usuário pode tomar devem ser compatíveis com o que o usuário já fez, ou seja, não se pode faltar ações inter-relacionadas, ações que se completam ou ações nas quais fazem parte de um conjunto de atividades. A falta de uma ou mais ações podem levar o usuário a realizar operações fora do contexto esperado, podendo comprometer a segurança do sistema.
- Familiaridade. Todas as ações consistentes que contém em aplicações semelhantes devem poder ser realizadas pelo usuário, permitindo que o usuário possa contar com ações para garantir a segurança.
- Multitarefabilidade. A interface deve permitir que ações correlacionadas sejam executadas em paralelo, desde que cada ação permita. Antes de realizar uma ação em certa parte de uma tarefa, o usuário pode desejar rever permissões e rever o estado atual do sistema, certificando-se de que a ação que ele tomará será segura. Caso a interface não deixe o usuário realizar outra tarefa ou ação paralelamente, o usuário talvez possa realizar a ação sem analisar o estado do sistema, arriscando a segurança do sistema.
- Migrabilidade de Tarefas. Os usuários podem esperar que o sistema automatize ações sempre que possível. Em aplicações críticas de segurança, provavelmente o usuário esperará que o sistema não automatize ações, passando a execução da tarefa para usuário a fazer manualmente. Essa necessidade o usuário tem para se certificar que o sistema não tomará ações que comprometam a segurança. Caso o sistema automatize as ações, a interface deve prover Observabilidade da execução da ação para o usuário, e Recuperabilidade se possível.
- Substituitividade. Valores de entrada e saída podem ser descritos de diversas maneiras por diferentes perfis de usuário. A interface deve esperar que valores de entrada e saída sejam utilizados pelo usuário. Em aplicações críticas os valores de entrada e saída devem ser de uma forma apenas, para não deixar dúvidas ao usuário, tendo-se um forte acompanhamento da interface ao usuário

para a entrada e saída desses valores, pois nesse caso um valor utilizado errado pode corromper ou desestabilizar o estado interno seguro do sistema.

- Multimodalidade. Alguns meios de comunicação entre a interface e o usuário fazem parte da resposta de ações do sistema, sendo a maior parte utilizada de meios visuais e auditivos. A falta de uma resposta de uma ação deixará dúvida no usuário, fazendo que ele tente repetir a ação diversas vezes esperando uma resposta, possivelmente sobrecarregando o sistema, ou fazendo que ele tente realizar novamente a ação de outra forma, talvez menos segura.
- Observabilidade. A representação do sistema pela interface conduzirá o usuário a esperar que ele esteja hábil para realizar as ações relativas ao que ele percebe. Tal representação deve ser consistente, de fácil e simples visualização. O erro dessa consistência pode levar o usuário à um estado inseguro do sistema.
- Recuperabilidade. O usuário espera que seja possível corrigir erros assim que identificados. Muitas vezes o usuário conta com a Recuperabilidade do sistema para que ele possa realizar alguma ação seguramente, e caso após a execução dessa ação seja identificado algum erro ou problema de segurança, o usuário irá imediatamente desejar que a interface disponha a ele a correção de erro. A falta de recuperabilidade em ações onde são possíveis recuperar pode gerar um problema crítico de segurança.
- Tempo de resposta. A interface deve fornecer a representação do estado de execução de ações que levem tempo, assim o usuário irá saber que a ação está sendo tomada e aguardará. Caso o tempo de resposta seja muito maior que o esperado e a interface não avisar isso ao usuário, talvez o usuário irá tentar realizar a tarefa através de outros modos inseguros.
- Conformidade de Tarefas. Relaciona-se diretamente com o princípio de Habilidade Esperada. Todas as ações conectadas ao mesmo escopo de tarefa devem ser fornecidas ao usuário pela interface, bem como as ações que o usuário espera realizar.

3.2.6.1 Atualizações de Sistemas Operacionais

Mesmo os sistemas operacionais sendo fortemente testados, continuam com erros, onde muitos são erros de segurança que podem comprometer os dados e os ambientes dos usuários. Como solução parcial, os sistemas operacionais são lançados com um sistema de atualizações, onde a maioria das correções é de segurança. A seguir seguem alguns exemplos de sistemas de atualizações de sistemas operacionais.

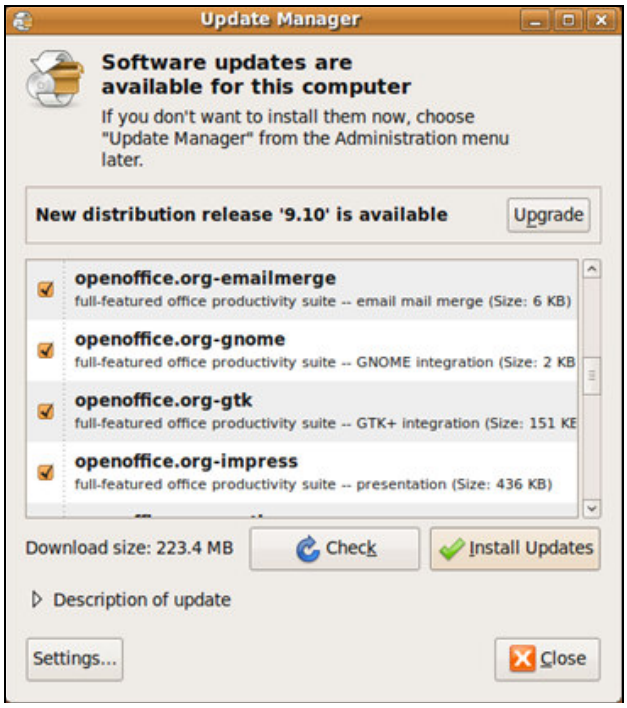

Figura 3.32: Gerenciador de atualizações do Ubuntu. O gerenciador fornece uma lista de todas as atualizações disponíveis.

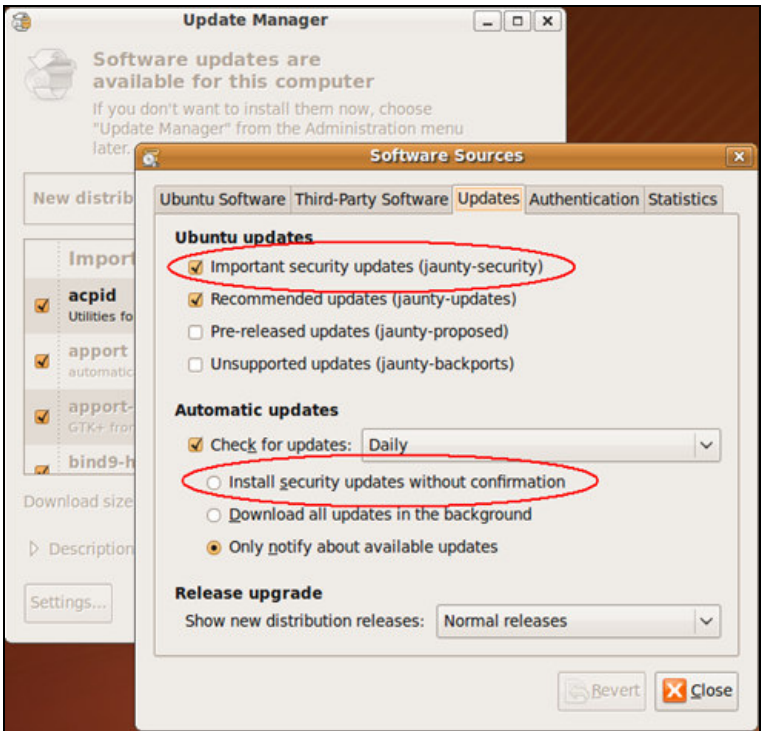

Figura 3.33: Opções de atualização no Ubuntu. Por padrão de configuração, fornecendo segurança, serão exibidas as atualizações de segurança. Uma das opções que o usuário pode selecionar é instalar automaticamente apenas as atualizações relacionadas à segurança.

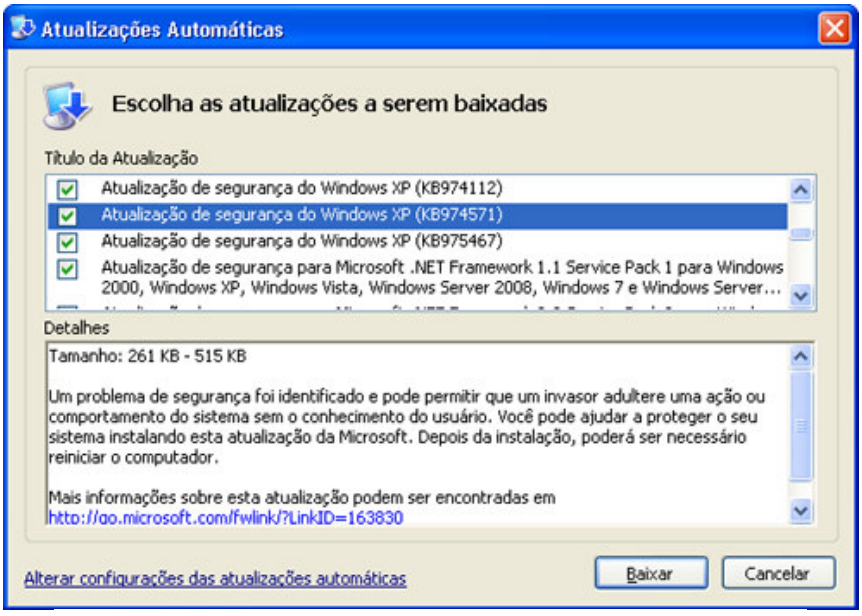

Figura 3.34: Gerenciador de atualizações do Windows XP.

As interfaces desses gerenciadores de atualizações não distinguem as atualizações de segurança das demais, tais atualizações de segurança deveriam ser prioritárias na

exibição e na instalação. Possivelmente por seguir o conceito de familiaridade, as interfaces são semelhantes e cometem quase as mesmas faltas.

Existem diversos problemas nessas interfaces simplificadas. Tais problemas podem fazer o usuário desistir de atualizar imediatamente o sistema operacional, ou até mesmo desabilitar as atualizações automáticas.

Os problemas dessas interfaces são:

- Não distinguir as atualizações de segurança das demais, uma solução seria agrupá-las de acordo com o tipo de atualização.
- Não priorizar as atualizações de segurança, instalando as atualizações de segurança antes das demais.
- Não facilitar ao usuário selecionar apenas as atualizações de segurança. Alguns usuários podem desejar instalar apenas as atualizações de segurança, por falta de tempo, de banda ou de espaço em disco.
- Não prover um botão para selecionar/remover todas as opções de atualização, facilitando a seleção de algumas atualizações apenas.
- Não informar o tamanho total para baixar as instalações, nem informar o espaço em disco estimado a ser ocupado, pois as atualizações podem vir compactadas, nem informar o tempo médio para instalar as atualizações. O sistema operacional Ubuntu exibe o tamanho total para baixar as atualizações.
- Descrever genericamente a atualização, exibindo um link para que o usuário acesse informações avançadas a respeito de cada atualização.
- Não deixar o usuário maximizar a tela de atualizações, dificultando a visualização das atualizações. Esse caso não se aplica para o sistema operacional Ubuntu.

3.2.7. Caminho Confiável

A interface deve prover um caminho de comunicação confiável e sem engano entre o usuário e quaisquer entidades confiadas para manipular permissões sobre o comportamento do usuário. Esse caminho de comunicação deve ser seguro, pois qualquer vulnerabilidade

dele pode bastar para que um invasor altere as permissões, até mesmo criando permissões para o próprio invasor utilizar o sistema e ter acesso total ao sistema.

A característica de **Caminho Confiável** relaciona-se com os critérios de usabilidade:

- Síntese. A interface deve fornece ao usuário um aviso do que está sendo alterado, através da detecção de mudança. Ao manipular permissões, o usuário irá aguardar algum retorno da interface garantindo o que foi alterado por ele.
- Familiaridade. O caminho confiável deve ser similar ao de outras aplicações do mesmo escopo, desde que tais aplicações utilizem corretamente o conceito de segurança.
- Iniciativa em Diálogo. Ao alterar as permissões do sistema, o usuário provavelmente aguardará que seja requisitada a sua senha novamente através de diálogo. Esse conceito se torna mais importante ainda se o usuário for administrador do sistema. A falta dessa requisição para garantir a autenticidade do usuário, pode ser um risco para o sistema, pois alguém pode manipular as permissões sem ser o verdadeiro usuário.
- Multitarefabilidade. Pode ser necessário que o caminho confiável seja manipulado concorrentemente com outras tarefas do usuário, ou seja, o usuário pode desejar alterar permissões durante qualquer execução de tarefa.
- Substituitividade. Para garantir a segurança, deve haver apenas um caminho confiável, não havendo substituitividade relacionada ao caminho confiável.
- Multimodalidade. Podem-se utilizar diversos meios de comunicação com o usuário para estabelecer-se um caminho confiável. Em aplicações críticas de segurança, pode ser necessário que o usuário utilize o mouse para entrar com os caracteres a senha, já que o computador pode estar sendo vigiado por programas maliciosos que armazenam senhas digitadas no teclado. Alguns sistemas utilizam também a biometria como garantia de autenticidade para o caminho confiável.
- Personalização. A necessidade de segurança do sistema pode fazer com que o caminho confiável seja personalizado para que a segurança seja ajustada,

adicionando-se ou removendo-se formas de autenticar o usuário, tais como citadas na multimodalidade.

- Observabilidade. A interface deve representar de algum modo o estado interno do sistema quando for utilizado o caminho confiável, para que o usuário certifique-se que está no caminho confiável.
- Conformidade de Tarefas. O caminho confiável deve ser íntegro, incorruptível, seguro e de fácil uso.

3.2.7.1 Manipulando usuários e permissões de usuários em sistemas operacionais

A maioria dos sistemas operacionais necessita de usuários cadastrados para o seu uso. Apenas usuários com permissões de manipular contas de usuários podem criar, editar e remover usuários do sistema. Por segurança do sistema e dos dados, deve-se ter um caminho confiável ao manipular usuários, autenticando-se o usuário a cada manipulação realizada.

Analisando-se o gerenciamento de contas de usuários em sistemas operacionais, vê-se que existe uma falha grave de segurança: a interface permite criar, editar e remover contas de usuários apenas com a autenticidade da sessão no sistema operacional. Com isso, qualquer pessoa pode acessar o computador enquanto estiver aberta a sessão, e então criar uma conta de usuário invasor para acessar fisicamente ou remotamente o computador, tendo acesso total a todos os dados contidos no computador. A solução para esse problema é pedir que o usuário digite sua senha ao manipular contas de usuários, impedindo que um invasor crie uma conta de usuário para ele discretamente.

A seguir seguem exemplos de caminhos confiáveis, ou a falta deles, em sistemas operacionais.

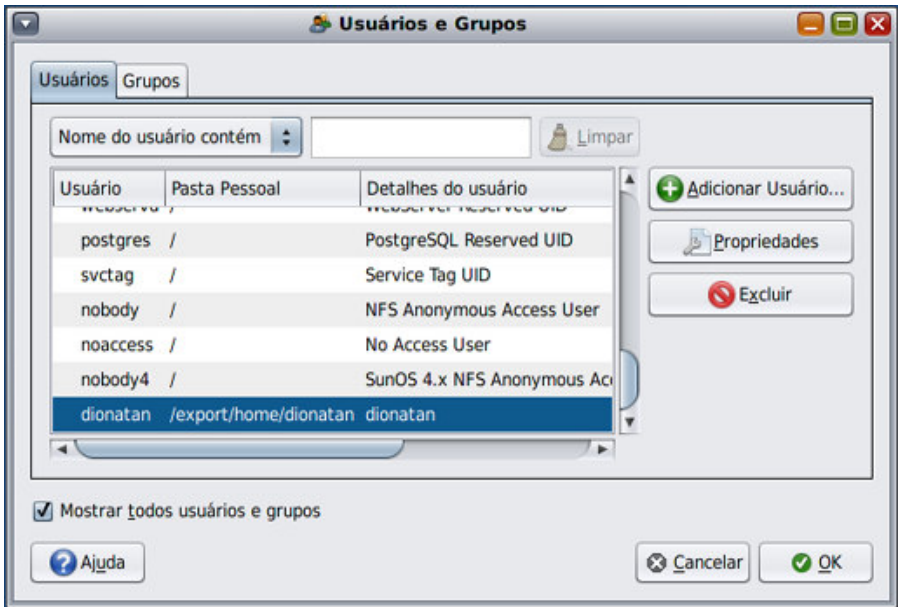

Figura 3.35: Manipulando contas de usuários no sistema operacional OpenSolaris. Nenhuma senha é requerida para manipular as contas de usuários. Qualquer pessoa com acesso físico ao computador com a sessão aberta pode adicionar usuários livremente.

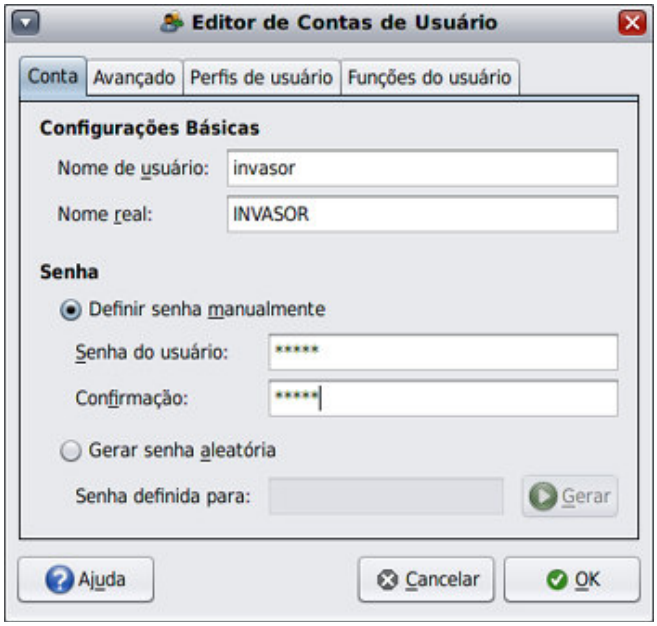

Figura 3.36: Adição de conta no OpenSolaris. O invasor adicionando uma conta de usuário para ele se autenticar no computador depois.

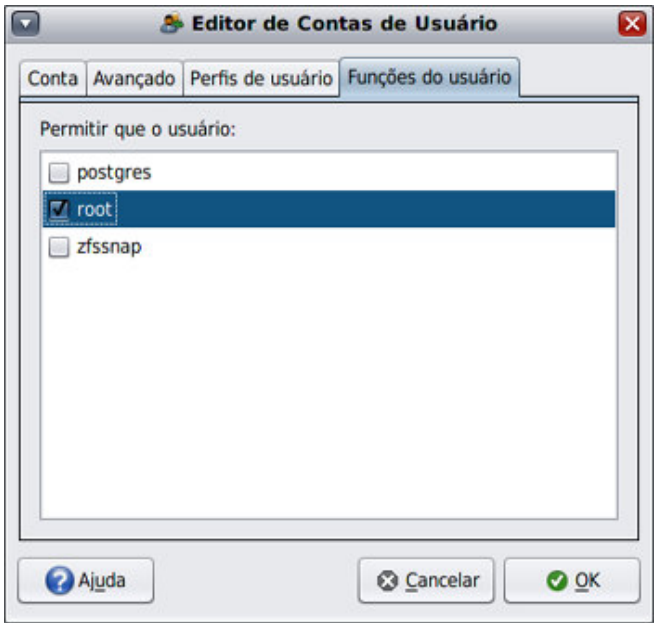

Figura 3.37: Configurando tipo de conta no OpenSolaris. O invasor cria uma conta do tipo *root*, na qual tem permissão total a todas as ações e aos objetos do sistema operacional.

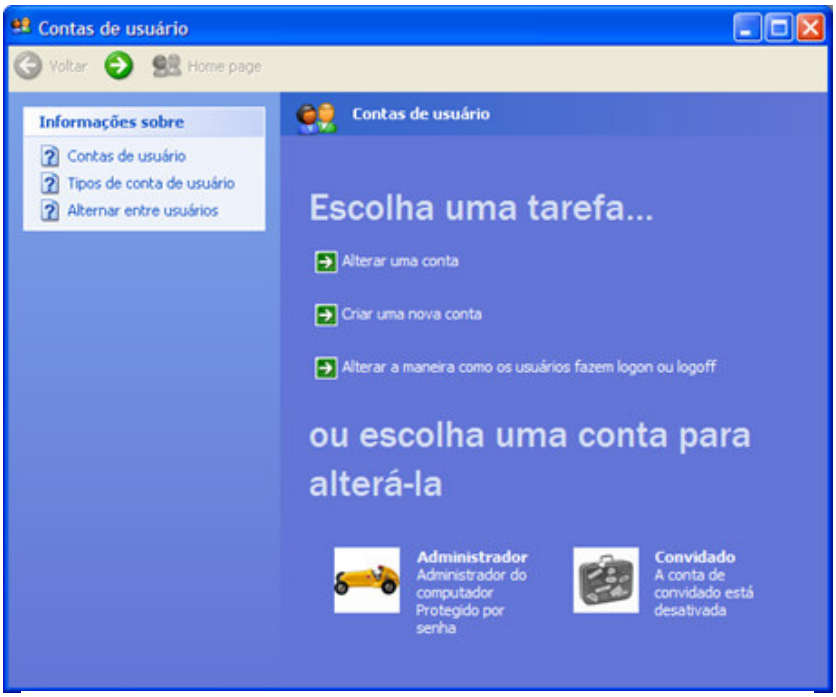

Figura 3.38: Manipulando contas de usuário no Windows XP.

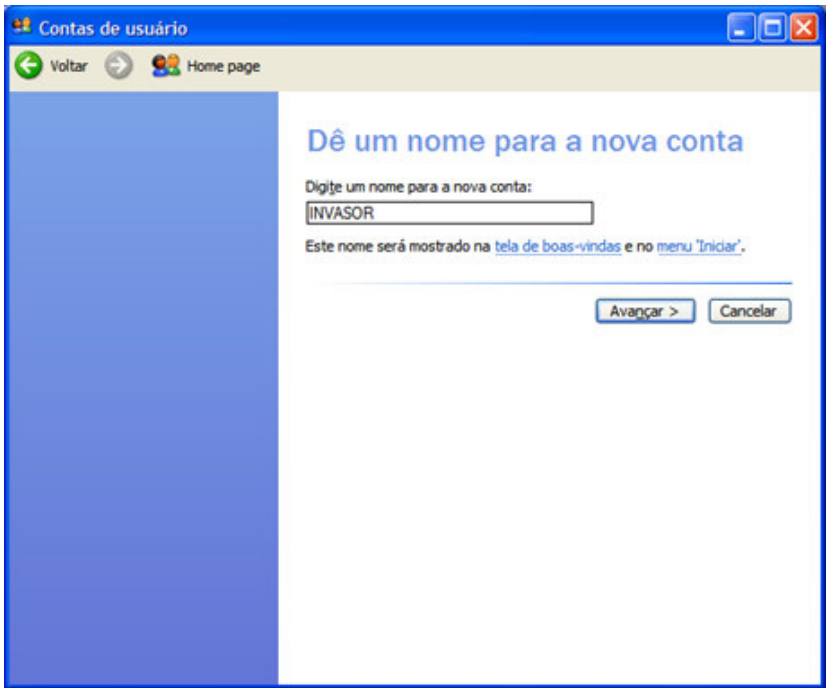

Figura 3.39: Invasor criando uma conta no Windows XP, para utilizar depois.

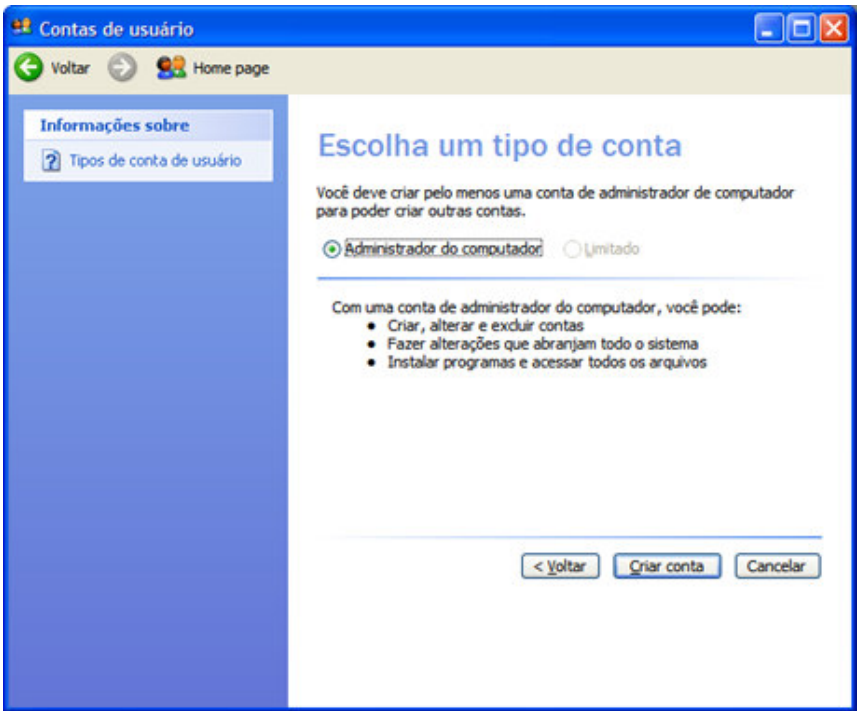

Figura 3.40: O invasor criando a conta com permissões de administrador do computador no Windows XP.

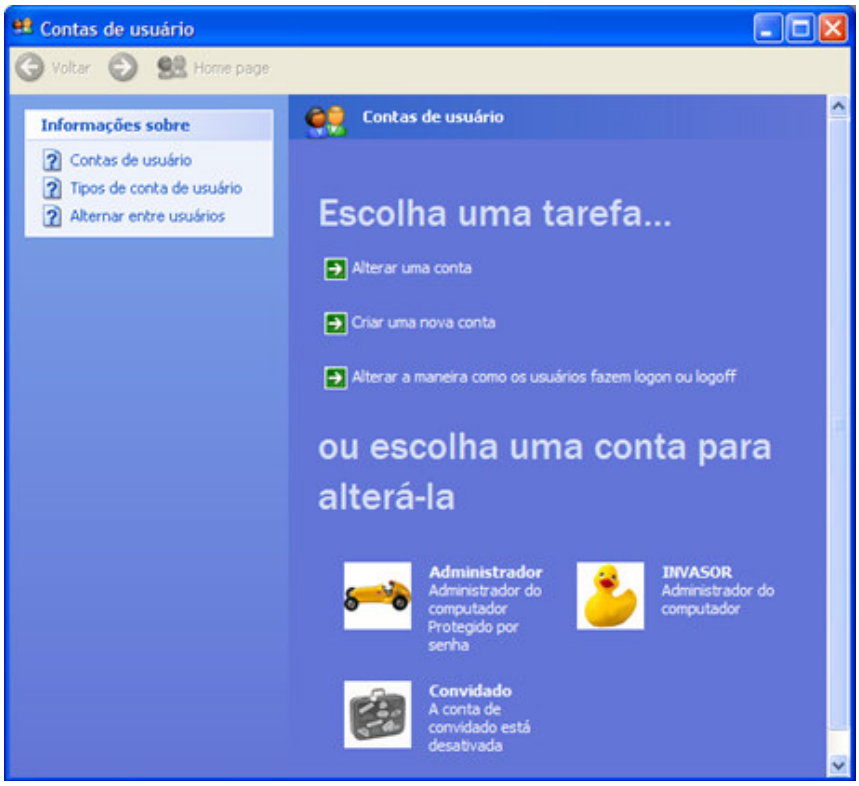

Figura 3.41: O invasor cria uma conta no Windows XP. A partir desse momento o sistema está totalmente vulnerável ao acesso físico ou remoto pelo invasor a qualquer momento.

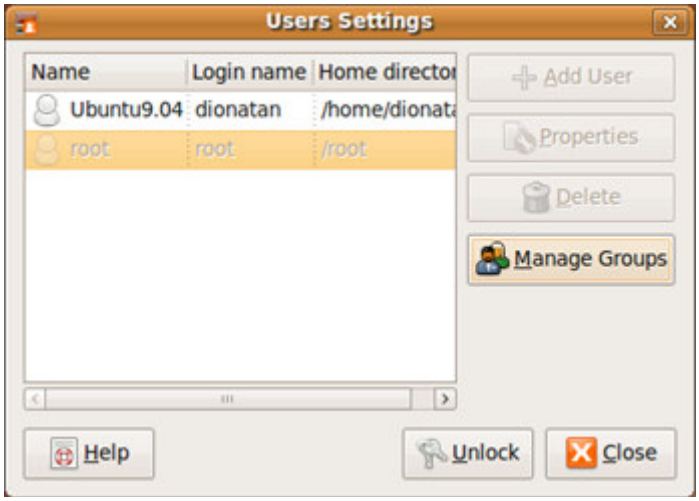

Figura 3.42: Manipulando contas de usuários no Ubuntu. Para manipular as contas de usuários é necessário se autenticar no sistema, clicando em "*Unlock*".

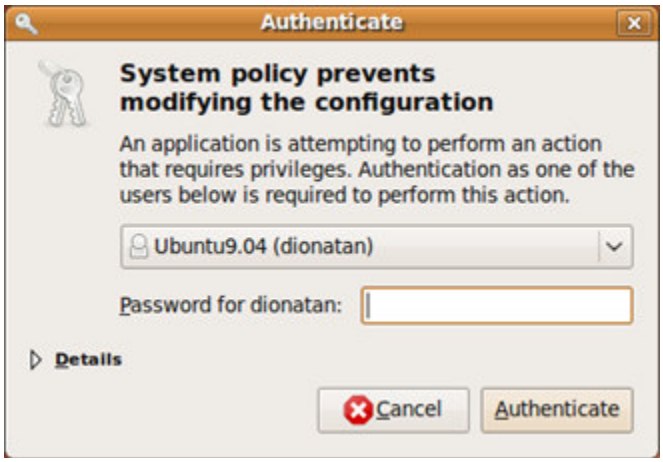

Figura 3.43: Pedido de senha para adição de usuário no Ubuntu. A senha do usuário com a sessão aberta no Ubuntu é requisitada para manipular as contas de usuários, criando-se um caminho confiável para manipular as contas de usuários, e impedindo que um invasor crie uma conta de usuário para ele acessar o computador depois.

Visto tais exemplos de manipulação de contas de usuários em sistemas operacionais, o único sistema operacional que tem um caminho confiável para manipular contas de usuários e permissões de usuários é o Ubuntu. Os sistemas operacionais OpenSolaris e Windows deixam essa falha de segurança em aberto.

3.2.8. Identificabilidade (*Identifiability*)

A interface deve reforçar a distinção entre objetos distintos e entre ações de um modo seguro. Tal distinção deve ser feita com uma identificação sem possibilidade de engano e com representações distinguíveis. A interface deve representar cada objeto igual sempre da mesma forma, para que haja identificação dos objetos, e deve também discernir objetos diferentes. A interface deve aplicar esse mesmo conceito tanto para objetos quando para ações. O princípio da Identificabilidade relaciona-se diretamente com os critérios de usabilidade de Previsibilidade, Síntese e Observabilidade.

A característica de **Identificabilidade** relaciona-se com os critérios de usabilidade:

- Previsibilidade. Todos os objetos do sistema e todas ações dos atores do sistema devem ser identificados através da interface pela suas igualdades, semelhanças ou diferenças.

- Síntese. A interface deve identificar os objetos que sofreram ações ao longo do tempo, assim o usuário poderá ter uma ideia melhor do que está acontecendo no sistema, tendo uma base aprimorada para tirar conclusões de segurança.
- Familiaridade. O conceito de familiaridade facilita a identificabilidade dos objetos, pois o usuário pode já estar acostumado com outras interfaces que implementam a Identificabilidade. Caso o usuário estiver familiarizado com outras interfaces e a interface atual não ser tão consistente quanto as outras interfaces familiares, gerar-se-á confusão e possíveis erros na utilização da interface.
- Generalização. O usuário pode difundir seu conhecimento da interface através do conhecimento de outras interfaces de propósitos semelhantes, fazendo analogia dos objetos e ações com as interfaces que já conhece.
- Iniciativa em Diálogo. Mensagens de autorizações explícitas, bem como mensagens de avisos de segurança e de erros devem ser apresentadas e identificadas facilmente e sem confusão pelo usuário.
- Multitarefabilidade. A interface que permite janelas abertas simultaneamente que tarefas sejam executadas em paralelo deve ter o cuidado para que o usuário não seja enganado através de janelas semelhantes de tarefas diferentes.
- Personalização. É desejável que a interface permita que o usuário a configure para melhor identificabilidade. Tais configurações podem ser por cores, tamanhos, tipos, formas e todas as características possíveis que auxiliem o usuário.
- Observabilidade. O usuário deve perceber a igualdade de objetos análogos, identificando que o comportamento desses objetos serão semelhantes. Da mesma forma, deve perceber a diferença de objetos não análogos, para identificar que seus comportamentos serão desiguais.
- Conformidade de Tarefas. Os objetos e ações devem ser identificados de acordo com as tarefas necessárias para o usuário realizar, de um modo correto que o usuário compreenda.

3.2.8.1 Janelas Órfãs

Um problema frequente em interfaces gráficas de sistemas operacionais é a representação das janelas filhas independentemente das janelas mães. Uma janela mãe é a janela raiz do programa, de onde as janelas filhas são criadas. Por exemplo, toda mensagem de autorização explícita é filha de alguma janela do sistema, considerando-se que a área de trabalho do usuário também é uma janela sem bordas.

Tais janelas filhas sendo representadas independentemente das janelas mães, ficam sem identificação própria e parecem que estão perdidas pela interface. Uma boa ilustração desse caso é o exemplo na figura a seguir.

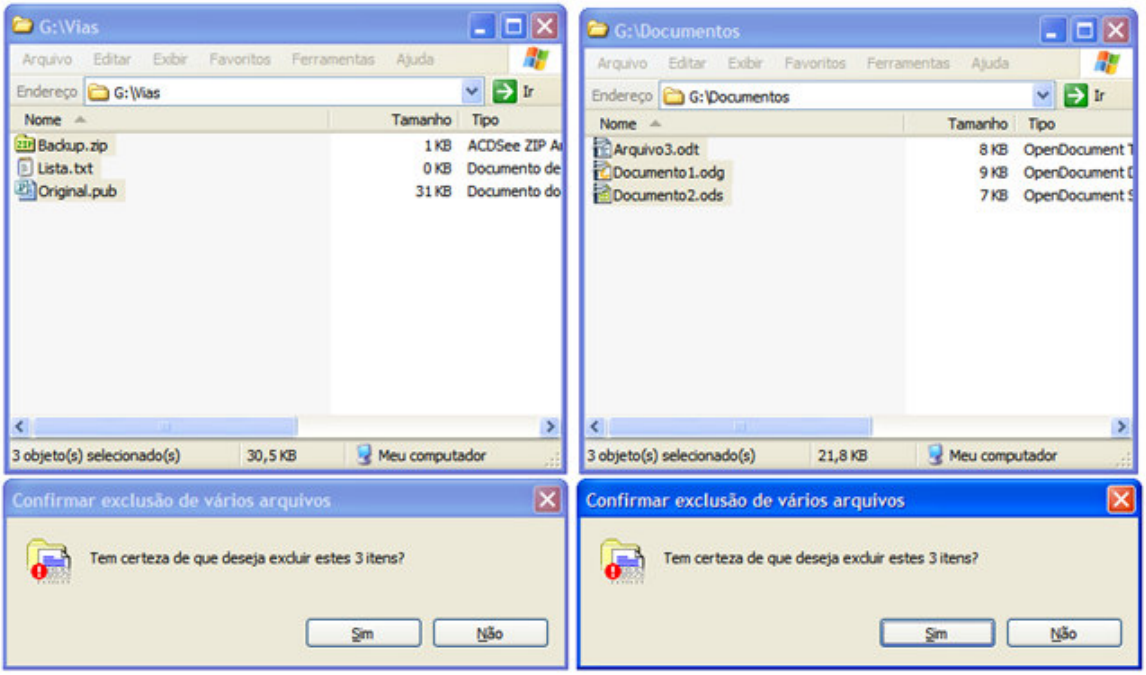

Figura 3.44: Janelas órfãs no Windows XP. Janelas órfãs de confirmação de exclusão de arquivos gerando confusão ao usuário em navegadores de arquivos, pois tais cada janela de confirmação não é identificada pela janela que a criou.

Analisando-se o exemplo da figura 3.44, quando o usuário tenta excluir arquivos de um navegador de arquivos, surge uma mensagem de confirmação para que os arquivos sejam excluídos. Tal mensagem não é identificada como sendo filha da janela do navegador de arquivos, surgindo uma certa dúvida no usuário. Caso o usuário esteja utilizando mais de um navegador de arquivos e tente excluir outros arquivos em algum outro navegador de

arquivos, duas janelas de mensagens de confirmações semelhantes existirão na interface com o usuário.

Como agravante desse exemplo, as mensagens de exclusão de arquivos não identificam quais arquivos serão excluídos, havendo duas mensagens exatamente idênticas de cada navegador de arquivo. Isso pode comprometer a segurança dos arquivos e dados do usuário, pois ele poderá confirmar de modo errôneo a deleção dos arquivos.

Como solução para esse problema, poderia ser utilizado um tipo de realce na janela mãe ao colocar o foco da aplicação na janela filha. Ou como outra possível solução seria tracejar linhas entre a janela filha com o foco da aplicação e a janela mãe respectiva. Identificar as janelas filhas a partir da janela mãe também pode ser necessário.

3.2.9. Expressividade

Políticas de segurança é um conjunto de regras a serem seguidas pelo sistema e pela interface do sistema, que pode ser personalizado de acordo com o nível crítico da aplicação. Elas podem ser descritas em um painel de configurações, ou até mesmo implicitamente nas ações realizadas pelo usuário. Tal política de segurança deve ser explícita ou implicitamente expressa de acordo com a necessidade do usuário.

A interface deve prover suficiente poder expressivo para descrever uma política de segurança sem dificuldade, e também para permitir que os usuários expressem políticas de segurança de acordo com seus objetivos. O princípio da Expressividade relaciona-se diretamente com o critério de usabilidade de Personalização.

A característica de **Expressividade** relaciona-se com os critérios de usabilidade:

- Previsibilidade. Opções de políticas de segurança devem ser fornecidas de acordo com a interação do usuário com a interface. A interação deverá se correlacionar com as opções de políticas de segurança para que o usuário não conte com políticas de segurança que não existam, sendo equivocado pela interface, e podendo realizar ações inseguras imaginando que existem políticas de segurança controlando essas ações.

- Síntese. Quando alteradas as políticas de segurança, o usuário deve sentir concretamente de alguma forma que as políticas de segurança foram aplicadas no sistema e na interface.
- Familiaridade. A interface deve expressar a segurança semelhantemente às políticas de segurança contidas em interfaces e sistemas semelhantes.
- Iniciativa em Diálogo. Mensagens com o usuário podem ser utilizadas para informá-lo das políticas de segurança de acordo com uso da interface. Tais mensagens devem ser utilizadas apenas quando necessário, e com uma alta expressividade em casos críticos de segurança.
- Personalização. Todas as configurações possíveis da política de segurança devem estar aptas a serem configuradas pela interface, de preferência por um painel de configurações, com as configurações separadas em grupos pela interface.
- Observabilidade. A interface deve representar como o estado interno do sistema procede com a segurança através da política de segurança do sistema.
- Conformidade de Tarefas. A expressividade da segurança deve estar de acordo com a aplicação e com a necessidade do usuário. O usuário pode realizar inconscientemente ações inseguras caso não entenda claramente como o sistema trata a segurança.

3.2.9.1 Políticas de Segurança em Navegadores Web

Navegadores Web utilizam regras de segurança contra para proteger os dados e o computador do usuário; bloqueando páginas identificadas como perigosas, avisando o usuário sobre questões de segurança a cada página sendo aberta.

A seguir serão apresentados e analisados alguns navegadores Web e mostrados como a interface deles expressa as políticas de segurança de navegação. O navegador Mozilla Firefox oferece as opções em um painel apenas, como visto na imagem a seguir.

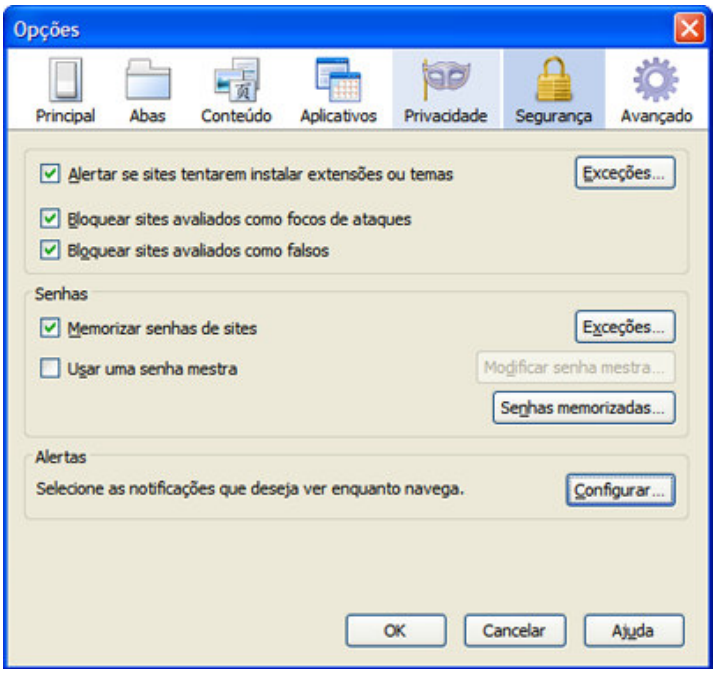

Figura 3.45: Opções de segurança no Firefox. Todas as opções de segurança estão centralizadas em um painel de opções. Tais opções são de bloqueios, alertas e senhas.

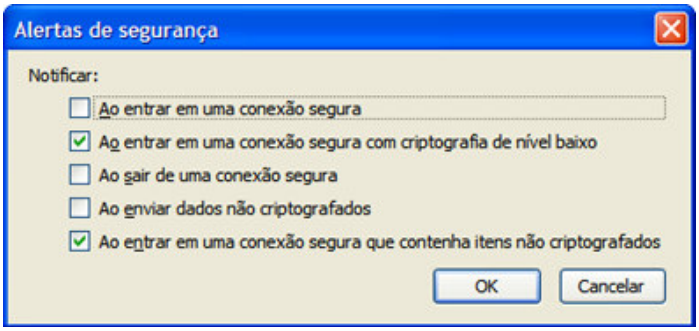

Figura 3.46: Configuração de alertas de segurança no Firefox. O usuário configura a política de alertas de segurança de acordo com sua necessidade.

Uma característica que facilita a interação com o usuário e também facilita expressar as políticas de segurança é centralizar todas as opções pertinentes à segurança em um só local, assim como a interface do Firefox implementa.

O navegador Google Chrome oferece uma maneira de deixar mais pessoais as configurações, como pode ser visto nas imagens a seguir.

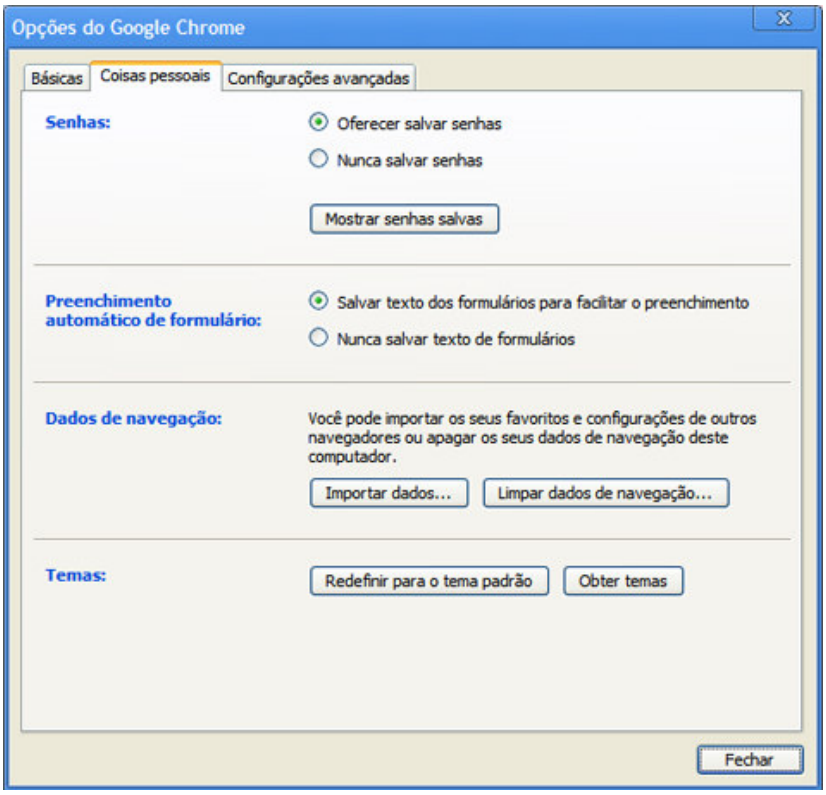

Figura 3.47: Configurações de opções pessoais no Google Chrome. As opções de segurança estão dispersas nas aba Coisas pessoais e na aba Configurações Avançadas. A aba Coisas pessoais contém a configuração de senhas, no qual facilita a interação com o usuário salvando senhas, mas deixa o usuário decidir se isso será seguro.

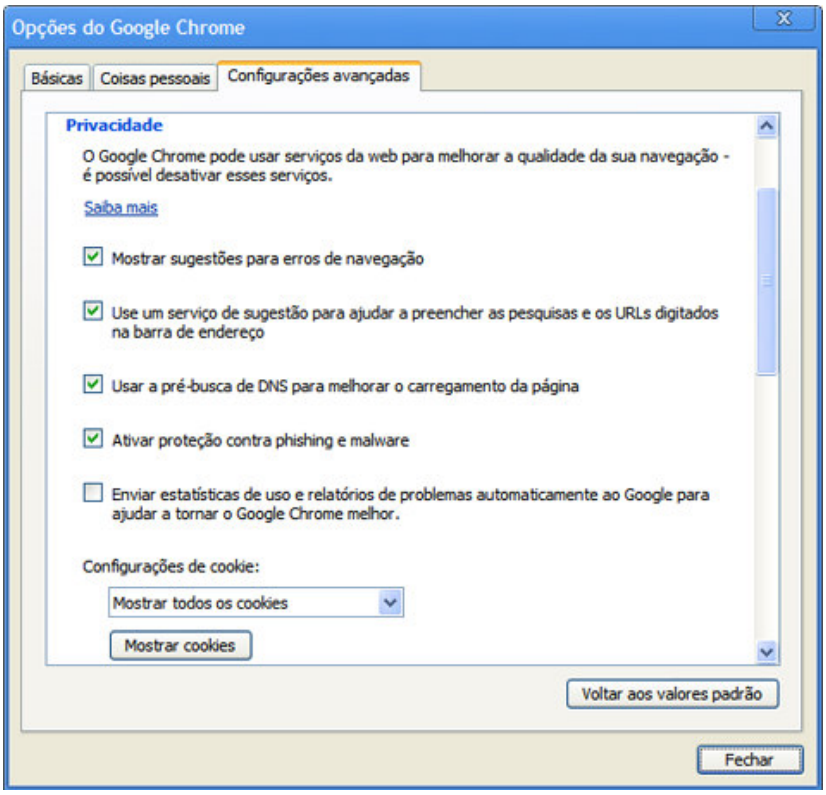

Figura 3.48: Configurações de privacidade do usuário no Google Chrome. A aba de Configurações avançadas contém no item de Privacidade a configuração de *phishing* e *malware*, no qual é um item de segurança relacionada à interação do usuário.

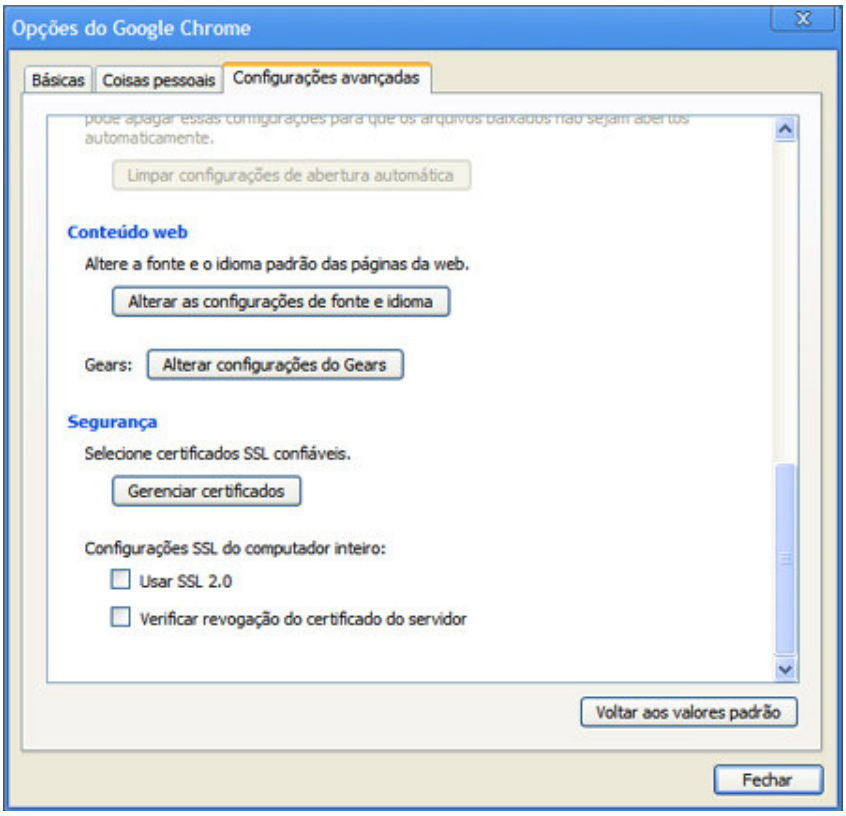

Figura 3.49: Configurações de segurança no Google Chrome. O item Segurança na aba Configurações avançadas contém apenas configurações relacionadas ao sistema, não à interação do usuário.

O Google Chrome separa as configurações políticas de segurança em distintas partes da interface, o que fica obscuro ao usuário, pois o usuário terá que ler todas as opções de configuração para decidir quais são de segurança, e só então poderá configurar a partir do entendimento que teve das políticas de segurança. Caso centralizasse como o Firefox faz, facilitaria para o usuário decidir sobre questões de segurança.

O navegador Internet Explorer oferece muitas opções relativas à segurança, como pode ser vistas nas imagens a seguir.

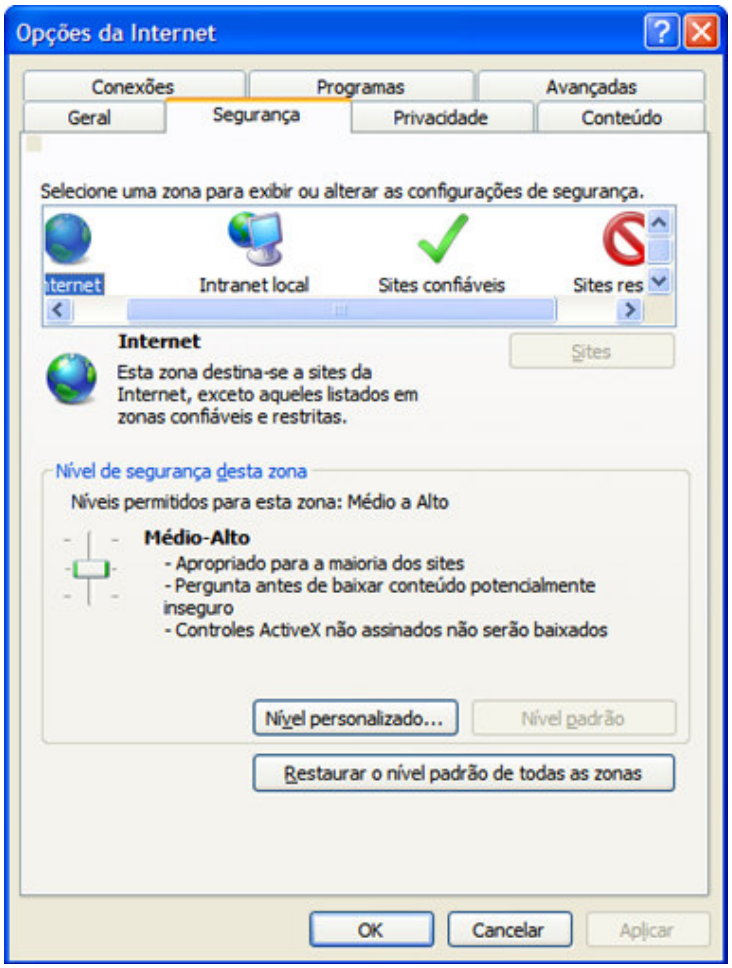

Figura 3.50: Configurações de segurança no Internet Explorer. Uma boa tentativa de políticas de segurança é apresentar níveis de segurança para a navegação na Internet. Um problema é o Internet Explorer permitir que o usuário possa selecionar apenas três níveis: Médio, Médio-Alto e Alto.

Essa baixa expressividade de níveis de segurança faz com que o usuário personalize o seu nível. Personalizando o nível faz com que o usuário decida diversas opções técnicas avançadas, ilustrado na figura a seguir.

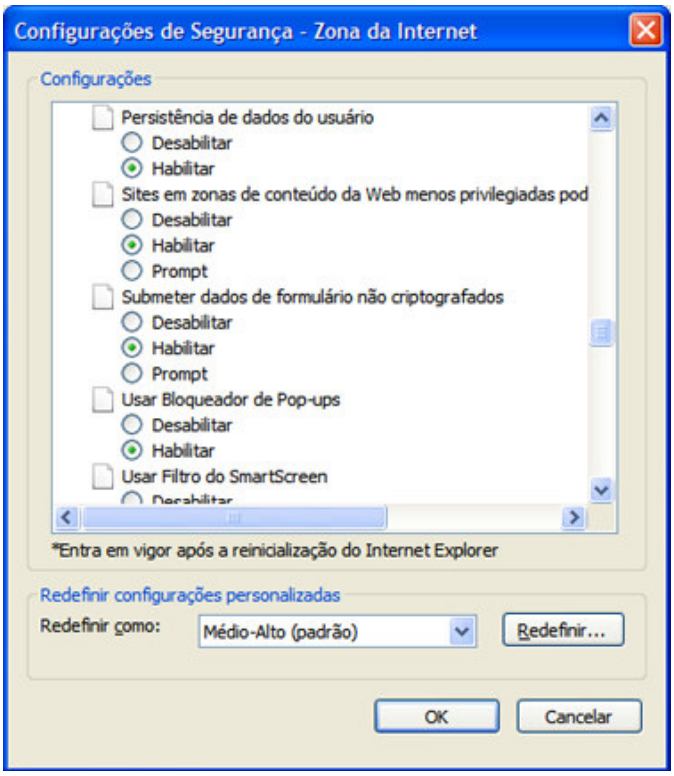

Figura 3.51: Configurando o nível de segurança da interface do Internet Explorer. Essas opções para configurar o nível de segurança podem ser muito avançadas e incompreensíveis para a maioria dos usuários, fazendo com que o usuário não configure a política de segurança, ou que configure de um modo que prejudique a sua segurança.

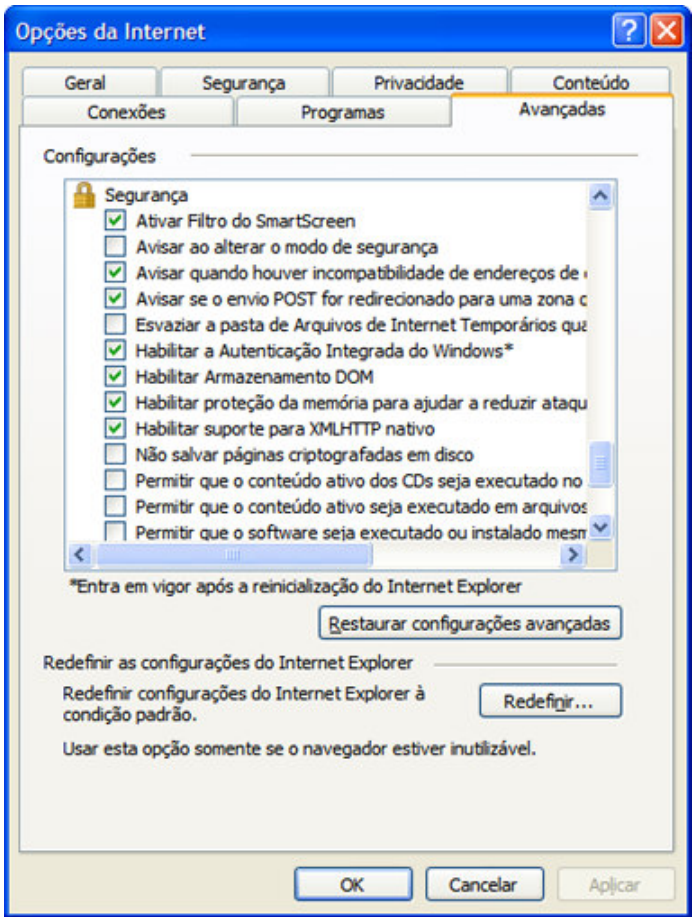

Figura 3.52: Configurações Avançadas no Internet Explorer. Gerando muita desordem na configuração do usuário, o Internet Explorer oferece mais opções de configuração na aba Avançadas, configurações nas quais são de difícil compreensão para um usuário comum.

O Internet Explorer oferece muitas opções avançadas, ao contrário do Google Chrome que oferece poucas opções avançadas, e do Firefox que centraliza todas opções num painel apenas. Visto isso, o browser com melhor expressividade é o Firefox. O Internet Explorer lista muitas opções avançadas e torna isso incompreensível ao usuário. Já o Google Chrome oculta a expressividade das políticas de segurança, fragmentando as configurações de segurança, na tentativa de tornar as configurações mais pessoais do usuário.

3.2.10. Clareza (*Clarity*)

A interação do usuário através de uma interface mal projetada, que oculta informações, que apresenta informações de modo errado, pode conduzir o usuário a tomar decisões inseguras, podendo corromper o sistema ou seus dados como um todo, desmerecendo toda a preocupação com a corretude do sistema que há por trás dessa interface.

A interface deve ser clara em toda ação relacionada à segurança, garantindo assim que o usuário não vá realizar operações perigosas. Toda informação relevante à segurança deve ser claramente apresentada antes ou durante a ação poder ser realizada pelo usuário. O princípio da Clareza relaciona-se diretamente com o critério de usabilidade de Observabilidade.

A característica de **Clareza** relaciona-se com os critérios de usabilidade:

- Previsibilidade. As possíveis ações a serem realizadas devem ser claramente previsíveis pelo usuário. Ele baseia-se na interação que teve até o momento para predizer o que poderá ser realizado, então ele deve poder confiar na interface para realizar as ações que lhe são esperadas. Algumas dessas ações podem ser relacionadas à segurança, e são nessas ações que o usuário deverá ter um cuidado especial através da sua interação.
- Síntese. O usuário deve estimar o que foi realizado nas ações passadas através da transparência das informações que a interface lhe dá no momento.
- Iniciativa em Diálogo. A clareza das mensagens de autorizações explícitas tem grande impacto nas decisões de segurança do usuário. Tamanhos variados de fontes, cores chamativas para casos críticos, sons, títulos adequados e mensagens sucintas participam diretamente nessas decisões. Qualquer autorização explícita realizada de um modo que não seja claro pode ser uma ameaça à segurança do sistema e do usuário.
- Substituitividade. A interface deve declarar explicitamente de um modo simples os possíveis valores de entrada e de saída que o sistema aceita.

- Personalização. A disposição e os modos permitidos da personalização devem ser facilmente encontrados e configurados na interface. Quanto mais próximo o usuário sentir-se da interface, mais tenderá a realizar ações de um modo seguro.
- Observabilidade. A representação interna do sistema deve ser clara através da interação com o usuário. O usuário deve observar de algum modo o sistema realizando as operações que lhe interessa, especialmente operações de segurança.
- Recuperabilidade. Recuperar-se de erros pode ser necessário a qualquer momento, ainda mais com erros relacionados à segurança. A interface deve prover uma interação clara com o usuário para que ele possa tornar o estado do sistema seguro novamente.

3.2.10.1 Mensagens seguras

Mensagens representam o estado interno do sistema ao se deparar com situações de aviso ou de decisões do usuário. Mensagens são apresentadas ao usuário através da interface, e pelo fato de representarem o estado interno do sistema, muitas vezes são mensagens voltadas ao sistema, e não mensagens voltadas ao usuário. O usuário deve facilmente entender o estado interno do sistema ao se deparar com uma mensagem. Mensagens de difícil compreensão pelo usuário podem fazer com que o usuário faça decisões erradas, caso essas mensagens sejam relacionadas à segurança do sistema, o usuário pode fazer decisões que comprometam a segurança como um todo.

A seguir são apresentados alguns exemplos de mensagens de segurança.

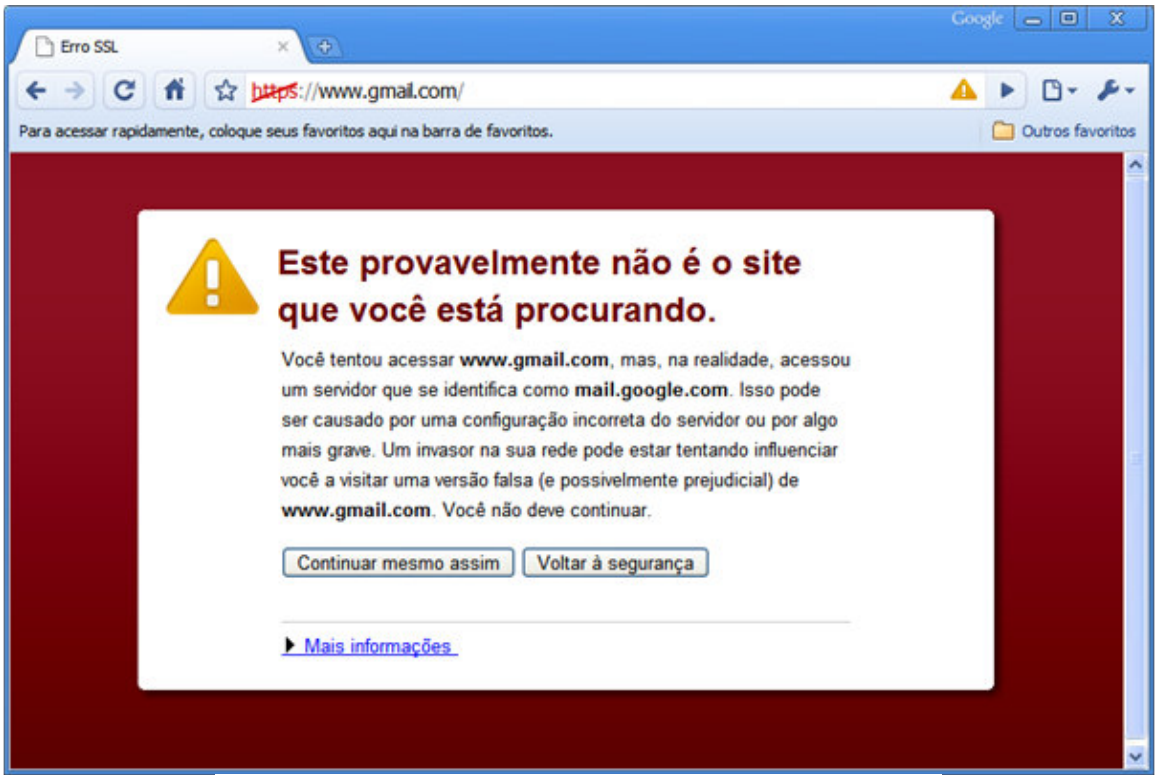

Figura 3.53: Erro de conexão segura no Google Chrome.

Na figura 3.53, a interface do navegador apresenta o aviso na própria janela de navegação de páginas web, aproveitando espaço da interface, e chamando a atenção do usuário utilizando cores chamativas, símbolos e tamanhos variados de fontes.

São utilizados três níveis de informação textual: o primeiro nível é o mais sucinto e com maior fonte, identificando claramente do que se trata o problema; o segundo nível detalha o problema, tendo fonte de tamanho médio e as principais informações em negrito; o terceiro nível é um hiperlink para maiores informações sobre esse problema. Os botões são concisos e fáceis de entender que ação será tomada ao selecioná-los.

A seguir segue uma mensagem de segurança que não é clara.

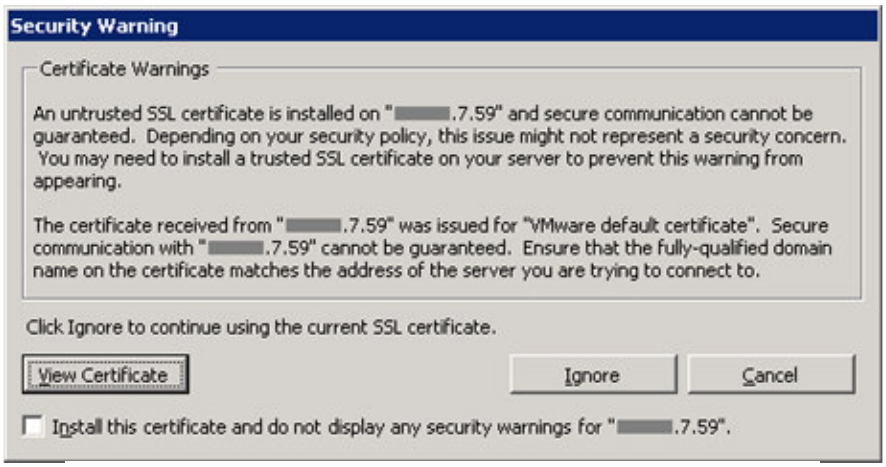

Figura 3.54: Erro de conexão segura no VMWare vSphere.

Na figura 3.54, a interface do sistema apresenta um aviso totalmente textual, sem identificações não-textuais de que é um aviso de segurança. A mensagem apresenta apenas um nível de informação textual, no qual é bastante técnico e não é sucinto. Não há uma ação definida a ser tomada, e as ações dos botões se confundem, pois ignorar e cancelar têm sentidos semelhantes. Essa mensagem possivelmente leve o usuário a tomar uma decisão duvidosa, na qual poderá levar o sistema a um estado inseguro.

4. CONCLUSÃO

Analisando os problemas relacionados à segurança da interação do usuário em interfaces inseguras, reafirma-se que é necessário dar importância às características do design de interação que visam à segurança do sistema e do usuário. A segurança dada em interfaces e interações com o usuário é requisito básico para desenvolver um software de qualidade. Os fatores e critérios de usabilidade utilizados auxiliaram suficientemente para explanar melhor os princípios de design de interação segura citados. Alguns princípios podem parecer antagônicos caso forem abordados isoladamente, por isso deve-se considerar o framework como um todo, pois os princípios são inter-relacionados.

A continuação desse trabalho pode ser feita a partir de uma análise para modificar ou encontrar novos princípios de segurança, critérios de usabilidade e relacionamentos dentre esses princípios e critérios. Tal análise poderá ser feita fazendo testes de usabilidade em protótipos de interfaces, ou através de outras características de interação e de usabilidade relacionadas, tal como em (1993, BASTIEN & SCAPIN).

Diversos tipos de programas desenvolvidos podem ser analisados com respeito à interação segura a partir desse trabalho realizado, identificando os pontos positivos e negativos de segurança relacionados à interação. Outra linha de continuação desse trabalho é adaptar um framework para engenharia de requisitos de sistemas interativos, tal como em (PIMENTA, 2000), dando-se enfoque na segurança da interação com o usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

2009. U.S.NRC - United States Nuclear Regulatory Commission. Backgrounder on the Three Mile Island Accident. http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html . Agosto de 2009.

2005. CRANOR, Lorrie ; GARFINKEL, Simson. Security and Usability, O'Reilly Media, Inc., 2005.

2005. PREECE, Jennifer ; ROGERS, Yvonn ; SHARP, Helen. Design de Interação: Além da Interação Homem-Computador. Porto Alegre. Bookman, 2005.

2004. NASA Software Safety Guidebook - National Aeronautics and NASA-GB-8719.13 - Space Administration, 31 de março de 2004.

2003. YEE, KA-PING. Secure Interaction Design and the Principle of Least Authority. Position paper accepted to HCI and Security Workshop at the ACM Conference on Computer-Human Interaction, 2003.

2002. YEE, KA-PING. User Interaction Design for Secure Systems (ACM). In Proceedings of the 4th International Conference on Information and Communications Security (Lecture Notes in Computer Science 2513), 278–290, Springer-Verlag, 2002.

2000. PIMENTA, M. S.. TAREFA: Uma abordagem para Engenharia de Requisitos de Sistemas Interativos. Terceras Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software (IDEAS00), 2000, Cancún. Anales de Terceras Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software (IDEAS00), 2000.

1999. WHITTEN, Alma ; TYGAR, J.D.. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In Proceedings of the 9th USENIX Security Symposium, Agosto de 1999.

1999. HOLMSTRÖM, Ursula. User-centered design of secure software. 17th Symposium on Human Factors in Telecommunications. Dinamarca, 1999.

1994. NIELSEN, Jakob; MACK, Robert L.. Published by John Wiley & Sons, New York, NY, 1994.

1993. BASTIEN, J.M.C.; SCAPIN, D.. Ergonomic Criteria for the Evaluation of Human-Computer interfaces. Institut National de recherche en informatique et en automatique, France, 1993.

1992. ABOWD, Gregory D. ; COUTAZ, Joëlle ; NIGAY, Laurence. Structuring the Space of Interactive System Properties. Engineering for Human-Computer Interaction 1992: 113-129.

1975. SALTZER, J. H. ; SCHROEDER, M. D. The protection of information in computer systems. Proc. IEEE 63, 9 (Sept. 1975), 1278-1308.

GLOSSÁRIO

Framework: Estrututura de suporte na qual serve como soluções para diversos problemas gerais, dentro de um determinado assunto.

Malware: Abreviação vinda do inglês “Malicious Software”, que significa software malicioso, que causa algum tipo de dano ao usuário.

Phishing: Palavra originada do verbo pescar do inglês, na qual designa a tentativa de se adquirir informações sigilosas de usuários, tentando se passar como uma pessoa ou empresa confiável.

Spam: Lixo eletrônico; mensagens de correio eletrônico que não tem utilidade, nas quais diversas contém vírus e tentativas de invasão.

APÊNDICE A - PESQUISA DE INTERAÇÃO DO USUÁRIO EM SEGURANÇA DE SISTEMAS DE COMPUTAÇÃO

Essa pesquisa foi realizada com 258 participantes, nos quais todos são usuários de sistemas de computação. Cada participante respondeu cada pergunta a seguir, gerando-se uma base para se obter conclusões sobre alguns problemas de segurança em interação com o usuário citados no trabalho no qual essa pesquisa está anexada.

Pergunta 1 – "Quanto tempo você tem de experiência no uso de computadores?"

Resposta	Usuários	Porcentagem
Menos de 1 ano.	1	0%
1 a 2 anos.	0	0%
2 a 5 anos.	7	3%
5 a 10 anos.	50	19%
10 a 15 anos.	117	45%
Mais de 15 anos.	83	32%

Tabela 01. Experiência dos usuários.

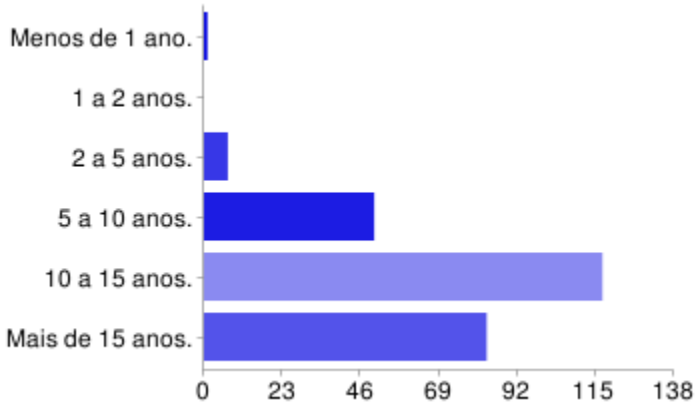

Imagem 01. Gráfico da experiência dos usuários.

Pergunta 2 – "Em relação à atualização do sistema operacional, você:"

Legenda	Resposta	Usuários	Porcentagem
A	Habilita as atualizações automáticas.	68	26%
B	Habilita apenas o aviso de que existem atualizações a serem baixadas, atualizando logo que possível.	161	62%
C	Habilita apenas o aviso de que existem atualizações a serem baixadas, deixando as atualizações de lado.	13	5%
D	Não atualiza, desabilitando as atualizações.	16	6%

Tabela 02. Atualização do sistema operacional.

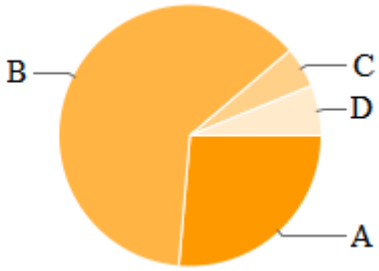

Imagem 02. Gráfico de respostas da atualização de sistema operacional.

Pergunta 03 – "Ao excluir arquivos, você utiliza a lixeira do sistema operacional?"

Legenda	Resposta	Usuários	Porcentagem
A	Sempre.	49	19%
B	Frequentemente.	59	23%
C	Raramente, na maioria das vezes excluo o arquivo diretamente.	138	53%
D	Desabilito a lixeira.	12	5%

Tabela 03. Utilização da lixeira do sistema operacional.

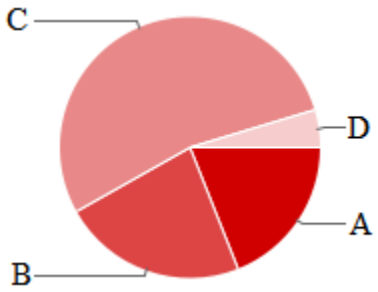

Imagem 03. Gráfico da utilização da lixeira do sistema operacional.

Pergunta 04 – "Com que frequência você costuma analisar processos que estão rodando no sistema operacional através do gerenciador de processos?"

Resposta	Usuários	Porcentagem
Sempre.	38	15%
Frequentemente.	117	45%
Moderadamente.	77	30%
Raramente.	24	9%
Nunca.	2	1%

Tabela 04 – Frequência do uso do gerenciador de processos do sistema operacional pelos usuários.

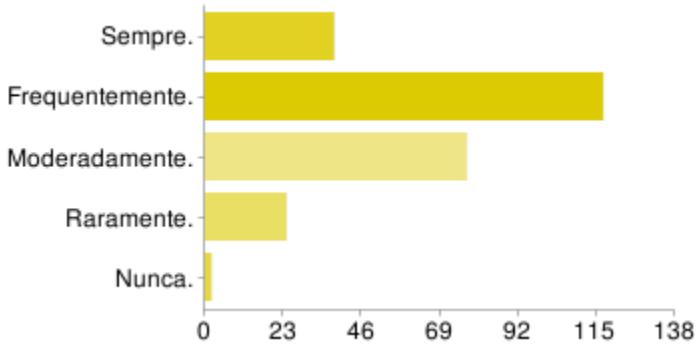

Imagem 04 – Gráfico da frequência do uso do gerenciador de processos do sistema operacional pelos usuários.

Pergunta 05 – "Você costuma salvar senhas no navegador Web de seu computador pessoal?"

Legenda	Resposta	Usuários	Porcentagem
A	Sim, sempre que possível.	39	15%
B	Sim, moderadamente.	82	32%
C	Não.	137	53%
D	Não se aplica.	0	0%

Tabela 05 – Modo que o usuário salva ou não as senhas no navegador Web do seu computador.

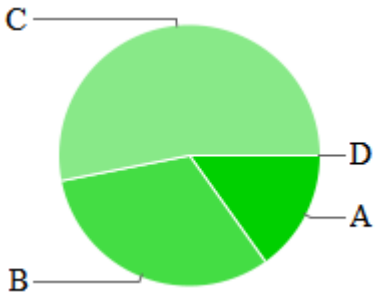

Imagem 05 - Gráfico do modo em que o usuário salva ou não as senhas no navegador Web do seu computador.

Pergunta 06 – "Ao entrar em uma conexão segura (HTTPS) em uma página Web, como você analisa se a conexão é confiável?"

Legenda	Resposta	Usuários	Porcentagem
A	Analiso a informação do certificado.	58	22%
B	Vejo se o cadeado de conexão segura está fechado.	95	37%
C	Espero avisos do navegador de que o certificado é não confiável para avisar.	48	19%
D	Não analiso.	57	22%

Tabela 06 – Modo que o usuário analisa a segurança da conexão em uma página Web.

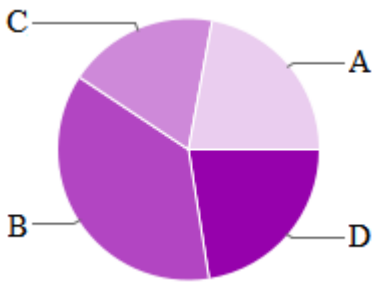

Imagem 06 – Gráfico do modo que o usuário analisa a segurança da conexão em uma página Web.

Pergunta 07 – "No Windows Vista, Windows 2008 ou Windows 7, quando é requisitada a permissão de realizar alguma tarefa, com que frequência você autoriza?"

Resposta	Usuários	Porcentagem
Sempre autorizo.	32	12%
Muito frequentemente.	69	27%
Frequentemente.	40	16%
Pouco frequentemente.	8	3%
Desabilito o UAC.	33	13%
Não utilizo tais sistemas operacionais.	76	29%

Tabela 07 – Frequência de autorização do usuário no Windows UAC.

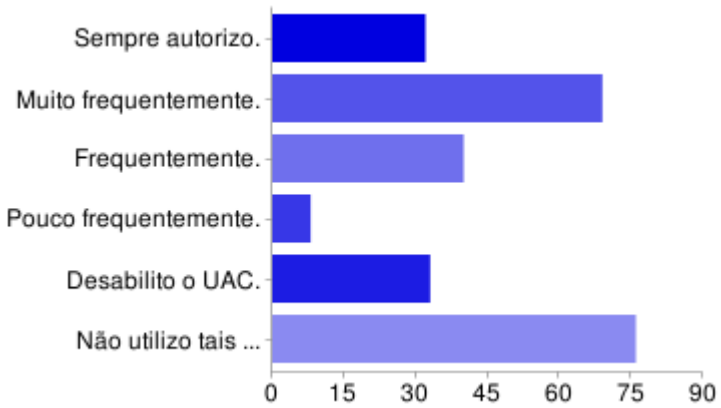

Imagem 07 – Gráfico da frequência de autorização do usuário no Windows UAC.

Resultados da Pesquisa

A partir da análise dessa pesquisa de 258 usuários de sistemas de computadores, notou-se que os usuários de sistemas de computadores eram de intermediários a experientes, pois o 97% dos participantes da pesquisa tinham pelo menos 5 anos de experiência no uso desses sistemas. Tudo indica que esse grande grupo de pessoas experientes participou da pesquisa, porque a pesquisa era baseada em formulários da Internet, e normalmente quem acessa a Internet já tem algum tempo de experiência no uso de computadores.

Em relação à atualização de sistemas operacionais, 26% dos usuários habilitam as atualizações automáticas, e o restante não as deixam automáticas por, provavelmente por uma questão de desempenho. 11% dos usuários não atualizam o sistema operacional de alguma forma, o que deixaria vulnerável a segurança dos dados e informações contidos no computador.

Ao excluir arquivos, apenas 42% dos usuários utilizam a lixeira de um modo seguro. Isso significa que mais da metade dos usuários não utilizam a lixeira, excluindo os arquivos diretamente, sem possibilidade de recuperá-los facilmente e de um modo garantido.

90% dos usuários analisam com pelo menos uma frequência moderada os processos do sistema operacionais pelo gerenciador de processos, o que significa que a visibilidade dos processos nos sistemas operacionais é muito precária e insegura.

Quase metade (47%) dos usuários salvam senhas de autenticação de páginas Web nos seus computadores pessoais, o que não significa um perigo muito grande por ser seu computador pessoal, porém está suscetível a vulnerabilidades dos navegadores Web no armazenamento dessas senhas.

Ao navegar em páginas Web, 41% dos usuários não analisam se a conexão é segura ou não. Já que os navegadores Web não identificam perfeitamente se a página é prejudicial, os usuários ficam suscetíveis a problemas de segurança.

No Windows Vista, Windows 2008 ou Windows 7, quando é requisitada a permissão de realizar alguma tarefa através do sistema UAC, 12% do usuários que autorizam sempre, podem estar executando programas maliciosos, bem como os 13% dos usuários que desativam tal sistema.

Concluindo, essa pesquisa apresenta o comportamento de usuários experientes em sistemas de computação, a quantidade de usuários que ficam vulneráveis a problemas de segurança é considerável. Tais vulnerabilidades poderiam muito bem ser evitadas caso a interface provesse uma interação facilitada em aspectos de segurança. Utilizar propriedades de design de interação em segurança aprimoraria a segurança do usuário e do sistema como um todo.