UNIVERSIDADE DO OESTE DE SANTA CATARINA
UNOESC – UNIDADE CHAPECÓ
ÁREA DAS CIÊNCIAS EXATAS E DA TERRA

FÁBIO ADRIÉL BECKERT

DESENVOLVIMENTO DE UM SOFTWARE OTP (ON-TIME PASSWORD, SENHAS DESCARTÁVEIS) PARA AUMENTO NA SEGURANÇA DE APLICAÇÕES WEB.

CHAPECÓ, SC
2012

FÁBIO ADRIÉL BECKERT

DESENVOLVIMENTO DE UM SOFTWARE OTP (ON-TIME PASSWORD, SENHAS DESCARTÁVEIS) PARA AUMENTO NA SEGURANÇA DE APLICAÇÕES WEB.

Relatório de pesquisa
Orientadora: Carla de Almeida Martins Basso
Pesquisador: Fábio Adriél Beckert
Curso: Sistemas de Informação
Área de conhecimento: Ciências Exatas e da Terra
Fonte financiadora: PIBIC

CHAPECÓ, SC
2012

RESUMO

Muitas aplicações web atualmente ainda não utilizam criptografia para serem acessadas, deixando a aplicação vulnerável a ataques feitos por captura de tráfego, e muitas máquinas clientes não possuem software de antivírus, comprometendo desta forma a segurança dos dados digitados e por consequência possibilitando a captura de senhas.

Por esse motivo verificou-se que com a utilização de senhas descartáveis, estas vulnerabilidades apesar de continuarem existindo, acabam sendo minimizadas pelo fato da senha ser válida por apenas um período de tempo.

Considerando os fatos acima mencionados, este trabalho foi elaborado com o objetivo de desenvolver dois softwares, um software cliente para rodar em smartphones e celulares com suporte a JME – Java Micro Edition, e um protótipo de software servidor com o objetivo de simular uma aplicação WEB.

O software cliente, é responsável por gerar as senhas descartáveis a cada intervalo de tempo, e o software servidor para demonstrar a forma de se efetuar a implementação do uso do sistema.

Considerando as atuais formas de furto de senhas e com a preocupação em se evitar os acessos indevidos a aplicações web, principalmente as aplicações onde o acesso indevido causa prejuízos muito reais como o caso das aplicações de internet banking é que surgiu a necessidade dos sistemas de senhas descartáveis do inglês On Time Password, várias são as formas de implementação.

Para este trabalho, utilizou-se as senhas baseadas em tempo, que usa o seguinte principio, através de uma determinada combinação de caracteres alfanuméricos conhecidos previamente pela aplicação servidora e pela aplicação cliente combinada com a data, hora e minuto atual, é feita a geração de um hash, do resultado deste hash é extraída uma parte também previamente definida nas duas aplicações, esta parte geralmente é composta de 6 caracteres, que são exibidos para o usuário na tela do dispositivo, onde o mesmo fornece para a aplicação e o mesmo calculo é feito no servidor, coincidindo os valores o servidor autentica o usuário, como a semente utilizada para gerar o hash é baseada no tempo, a cada fração de tempo a senha muda.

Este artigo apresenta a solução desenvolvida, envolvendo tanto a parte do cliente como do servidor, os modelos de smatphones utilizados e toda a implementação.

Palavras-chaves: Token, One-Time Password (OTP), Senhas descartáveis, Segurança site internet.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Com o avanço no uso dos sistemas de informações nas organizações, e a disponibilização dos mesmo sob a internet, muitas vezes um quesito importantíssimo é deixado de lado, que é a segurança no acesso as informações, quando o acesso ao sistema é liberado pela internet, foge dos controles da empresa os meios utilizados para acessar tais sistemas, e muitas vulnerabilidades podem ser exploradas pelo caminho, e é exatamente por este motivo que o uso de um software otp na autenticação do usuário reduz drasticamente os problemas relacionados ao acesso não autorizado, e a falsificação da identidade.

A criação do OTP será feita da seguinte forma, através da concatenação de uma determinada chave alfanumérica e em uma ordem pré-determinada a data,hora e minuto da execução da requisição dividido pelo fator de tempo definido, conhecida pelo software cliente e servidor, após, é aplicado o algorítimo de hash md5 e extraído do mesmo uma determinada parte que contenha 6 caracteres, destes 6 caracteres a parte hexa-decimal é convertida em números, com uma determinada relação pre-definida, e este número resultante será a senha.

O Software cliente foi desenvolvido em j2me objetivando o uso do mesmo em aparelhos celulares/smathphones com suporte a tecnologia de MIDlets.

E o servidor pode ser uma aplicação em php, jsp ou outra que possua suporte a algorítimos de hash md5, e funções básicas de manipulação de strings e tempos.

Como a eficiência do otp depende da sincronização entre os relógios do aparelho e do servidor, foi criado um script no servidor linux para manter sincronizado o relógio com algum servidor ntp.

No cliente, os aparelhos sincronizam seus relógios com as operadoras, o que nos garante uma precisão mínima para a operação do software.

Este artigo está organizado em cinco seções. Após esta introdução, a seção 2 apresenta o referencial teórico. Posteriormente, na seção 3, é descrito os procedimentos metodológicos.Na seção 4 são apresentados os resultados e discussões sobre o resultado da pesquisa. Por fim, a seção 5 traz as conclusões do estudo.

2 REFERENCIAL TEÓRICO

Para que se possa entender a real necessidade do uso de senhas descartáveis em aplicações web, o funcionamento de vários protocolos e padrões utilizados na internet para que uma aplicação web possa rodar serão elencados, demonstrando onde se encontram os pontos vulneráveis.

2.1 A ARPANET

Segundo BELLIS (2011) o precursor da internet foi a ARPANET (Advanced Research Projects Agency Networks), que foi um projeto financiado pelo exército dos Estados Unidos da América após a gerra fria com o objetivo de ter um centro de comando e controle militar que pudesse resistir a um ataque nuclear. O objetivo era distribuir informações entre computadores geograficamente dispersos.

Roberts (1995) e Waldrop (2009) descrevem que em 23 de abril de 1963 Licklider tinha um ideia muito interessante, a de interligar pessoas em todos os lugares através de computadores de uma forma que o conhecimento pudesse ser compartilhado por todos, e que a DARPA (Defense Advanced Research Projects Agency) pretendia investir em um projeto de interconexão entre os seus sistemas de tal forma que não existisse apenas uma única rota para a comunicação entre os computadores então foi criado o projeto Lincoln. Porém foi só em 1965 que Taylor Brought e Larry Roberts no laboratório Lincoln continuaram desenvolvendo o projeto: Em primeiro lugar ficou definido que o sistema usariam linhas comuns de telefones que ficariam permanentemente conectadas. Em segundo lugar Roberts decidiu que as mensagens digitais não seriam enviados através da rede como um fluxo contínuo de bits. Em vez disso, como uma longa carta escrita em uma série de cartões postais, cada mensagem seria quebrada em segmentos de comprimento fixo o que hoje é conhecido como pacotes. A idéia era proteger contra energia estática e distorção na linha, o que poderia facilmente inutilizar todo o conteúdo enviado, os pacotes não iria eliminar o ruído, mas eles iriam isolar os erros e dar ao sistema a chance de corrigi-los, seja através de sofisticados mecanismos de correção de erro, ou solicitando ao remetente original uma nova cópia. Por

fim Roberts decidiu tornar a rede completamente descentralizada, sem um computador principal responsável pela classificação e roteamento dos pacotes até seu destino, ele decretou que os locais da DARPA seriam ligados em um padrão complexo. Isto é, o computador iria ler somente o endereço digital de cada pacote e então aceitar os que são destinados para ele, ou então enviá-lo novamente para a próxima etapa da sua jornada. Este abordagem significaria mais complexidade porém o sistema final seria muito mais robusto e tolerante a falhas. Em 1972, Roberts contratou Robert Kahn,para supervisionar a implementação a comutação de pacotes com comunicações via rádio e satélite. Kahn logo começou a tropeçar em mais uma daquelas questões triviais, que parece ser no final o mais profundo de todos. Como ele faria para que todas essas redes de satélites e redes de rádio conversassem entre si. As diversas redes foram otimizadas para ambientes muito diferentes. A ARPANET, por exemplo, vivia num mundo de estabilidade comparativa, com pacotes fluindo sobre uma rede fixa, de confiança, em linhas terrestres de telefone. Mas os pacotes de rádio vivia num mundo de caos, com transceptores sempre em movimento, sendo cortados por morros e túneis, e geralmente, perdendo pacotes constantemente. Os dois sistemas tiveram grandes incompatibilidades no tamanho dos pacotes transmitidos e na velocidade, praticamente qualquer parâmetro poderia ser variável. Kahn percebeu que a melhor maneira de integrar essas redes foi começar por separalas. Em seguida, conectar à ARPANET através de uma espécie de gateway conhecido hoje como roteador, que nada mais é do que um computador especializado, cujo único trabalho seria o de traduzir pacotes ARPANET em pacotes por satélite e vice-versa. Enquanto ambos os lados cumprissem as normas (que ainda não tinham sido definidas), nenhum dos lados teria que saber os detalhes internos do outro. Como as coisas estavam, tentando enviar um pacote de uma rede para outra era como tentar enviar um cartão postal escrito em caracteres japoneses através de uma estação de correios nos Estados Unidos. Então Kahn e Cerf, usando a analogia como um cartão postal poderia ser enviado dentro de um envelope endereçado na língua local e em seguida, quando cruzar a fronteira seria colocado dentro de outro envelope endereçado na nova língua. Dessa forma, cada pacote poderia navegar através de sua rede local até chegar à entrada da rede seguinte quando então o computador gateway iria colocar o pacote em um novo envelope compatível com a nova rede. O protocolo universal em si teria que lidar com um número de aspectos práticos, incluindo um protocolo inter-redes que iria codificar as coisas tais como o endereço final de um pacote, e um esquema de controle de transmissão que permite ao computador de destino solicitar substituições para os pacotes que haviam sido perdidos em trânsito. Assim, em 1974, surgiu o Transmission Control Protocol/Internet Protocol, ou TCP/IP. Em março de 1982 o TCP/IP

versão 4 foi considerado bastante confiável para o Departamento de Defesa para torná-lo o padrão para todas as redes de computadores militares. E em 1 de janeiro de 1983 uma data que sem dúvida marca o nascimento do mundo real da Internet a ARPANET se mudou para TCP/IP. Em 1989, a ARPANET foi formalmente encerrada, porém tinha dado à luz a Internet que estava apenas começando.

2.2 A CSNET

Mandel (1997) descreve que o sucesso da Arpanet se propagou rapidamente a outras comunidades, que não possuíam contratos com o DoD (Department of Defense), mas desejavam participar da nova era das comunicações. Assim nasceu a Csnet em 1979 e tornou-se operacional em 1982, uma rede computacional relativamente barata que interligou todos os departamentos de Ciência da Computação dos EUA. O conjunto da Arpanet e da CSnet constituiu-se na primeira rede heterogênea e pode ser considerado o precursor da Internet.

2.3 A USENET

Mandel (1997) A terceira rede que merece menção nesta retrospectiva histórica é a chamada Usenet, por suas características únicas. Trata-se de uma rede que obteve o maior número de participantes nos anos pioneiros, sem contar com nenhum tipo de subsídio centralizado. Era baseada numa arquitetura muito simples, sem precisar da comutação de pacotes, mas muito fácil de ser realizada. Oferecia apenas os serviços mais simples como correio eletrônico e transferência de arquivos. A rede levou ao estabelecimento de um novo serviço, o News ou a conferência mundial, que se tornou muito popular e que existe até hoje. A Usenet era baseada no programa uucp (Unix to Unix copy) que vinha incluído em todos os sistemas Unix, cuja popularidade crescia exponencialmente nessa época. Esse programa nada mais é do que a cópia de arquivos entre sistemas remotos. As ligações eram feitas por linha discada, usando modems de 300 a 2400 bps, em horários predeterminados, quando dois computadores trocavam as suas informações. Para se conectar à rede tudo que era necessário era um computador rodando Unix com acesso a um modem, uma linha discada e de um amigo

ou organização dispostos a servir de ponto intermediário de comunicação. Assim, a rede cresceu de forma absolutamente descentralizada, anárquica mesmo, chegando a centenas de milhares de usuários em 1986 . Essa rede iniciou-se por volta de 1978 e posteriormente a arquitetura foi aproveitada para formar uma das primeiras redes européias, a Eunet.

2.4 RFC (REQUEST FOR COMMENTS)

Mandel (1997) A partir daí o protocolo TCP/IP começa a ser adotado em massa e a internet ganha vida e se torna como é conhecida atualmente, e como ela se dá de forma anárquica, o que mantém a Internet como uma unidade é o protocolo TCP/IP e o mecanismo público de fixação de suas normas através dos chamados RFC (Request for Comments), cuja aderência rigorosa é forçada pelo desconforto, que se manifestaria em dificuldade de comunicação com o resto do mundo, de quem queira, porventura, se afastar. Em outras palavras, o volume da Internet no momento é tal que se torna impossível qualquer tentativa de concorrência com ela.

IETF (2011) A missão do IETF (Internet Engineering Task Force) é “fazer a Internet funcionar melhor”, do ponto de vista da engenharia. A maioria dos participantes do IETF são engenheiros com conhecimento de protocolos de rede e software. Os produtos oficiais do IETF são documentos, publicados gratuitamente como RFCs, e esse nome (usado desde 1969, antes do IETF existir) expressa algo importante: a Internet é um sistema em constante evolução técnica, e qualquer documento que é escrito hoje pode precisar ser atualizado amanhã. Uma maneira de olhar para o IETF é como o grupo de pessoas que trabalham em conjunto para melhorar a tecnologia da Internet em uma base diária. Bem como produzir RFCs, o IETF é um fórum onde os operadores de rede, implementadores de hardware e software, e os pesquisadores falam uns com os outros para garantir que o futuro de protocolos, padrões e produtos será ainda melhor. Este não é o único fórum técnico para a Internet, é claro. Mas é o fórum onde as normas de base técnica para os protocolos da Internet são definidos e mantidos. O IETF não padronizar hardware de transmissão isto fica a encargo de organizações como o IEEE (Institute of Electrical and Electronics Engineers) e ITU (International Telecommunication Union) e não padronizar protocolos de camada de

aplicativo especializado. Por exemplo, deixamos os padrões HTML e XML para o W3C (Consórcio World Wide Web). Nem todos os RFCs são padrões, eles são classificados em:

- Proposed Standard (PS). A primeira etapa oficial, mas muitos padrões nunca progridem além deste nível, provavelmente porque os membros não gostam de burocracia.
- Draft Standard (DS).O segundo nível, quando a interoperabilidade das implementações têm sido convincentemente demonstrada.
- Standard (STD). A etapa final, quando o padrão é mostrado para ser amplamente implantado. No entanto, uma nova PS ou DS pode tornar um padrão obsoleto.
- Best Current Practice (BCP). São recomendações das melhores práticas correntes.

O texto de um RFC nunca muda. Quando um RFC é atualizado, ele recebe um novo número.

2.5 HTTP (HYPERTEXT TRANSFER PROTOCOL), WWW (WORLD WIDE WEB) E HTML(HYPERTEXT MARKUP LANGUAGE)

Berners-Lee et al (1996) O HTTP (HyperText Transfer Protocol) é um protocolo leve desenvolvido devido a necessidade de distribuição de conteúdo em sistemas de informação. É um protocolo genério orientado a objeto que pode ser usado para muitas tarefas, tais como servidores de nomes e sistemas de gerenciamento de objetos distribuidos através da extensão dos seus métodos de pedido (comandos). Uma das características do HTTP é a tipagem de representação de dados, permitindo que os sistemas sejam construído independentemente dos dados serem transferidos. HTTP tem sido usado pela World Wide Web desde 1990.

Berners-Lee e Cailliau (1990) A essência por trás da WWW (World Wide Web) foi iniciada em 1989, no CERN (Organização Europeia para a Investigação Nuclear), baseado no ENQUIRE de Berners-Lee rodando sob o protocolo HTTP.

Berners-lee (1955) O projeto da WWW continha algumas das mesmas ideias primordiais do ENQUIRE, e também algumas ideias da web semântica. Seu intento original

do sistema foi tornar mais fácil o compartilhamento de documentos de pesquisas entre os colegas.

Berners-Lee e Cailliau (1990) Em março de 1989, Tim Berners-Lee escreveu uma proposta de gerenciamento de informação, referenciando o ENQUIRE e descrevia um sistema de informação mais elaborado. Com a ajuda de Robert Cailliau, ele publicou uma proposta mais formal para a World Wide Web no final de 1990. Berners-Lee escreveu o primeiro servidor web e também escreveu o primeiro navegador, o WorldWideWeb, em 1990. No final do mesmo ano, Berners-Lee já havia construído todas as ferramentas necessárias para o sistema: o navegador, o servidor e as primeiras páginas web, que descreviam o próprio projeto. Em 6 de agosto de 1991, ele postou um resumo no grupo de notícias. Essa data marca a estreia da Web como um serviço publicado na Internet.

A ideia revolucionária de Tim foi unir o hipertexto e a Internet. Berners-lee implementou o casamento das duas tecnologias e no processo, ele desenvolveu um sistema de identificação global e único de recursos, o Uniform Resource Identifier (URI). (BERNERS-LEE,1999)

LONGMAN(1998) fala que em 1989 Tim Berners-Lee inventou a Web com o HTML como linguagem de publicação. Em 1989, Tim estava trabalhando em uma seção de serviços de computação do CERN, quando lhe veio o conceito, no momento ele não tinha idéia que seria implementada em uma escala enorme. Pesquisa em física de partículas, muitas vezes envolve a colaboração entre os institutos de todo o mundo. Tim teve a idéia de permitir que pesquisadores de locais remotos do mundo pudessem se organizar e reúnirem informações. Mas, longe de simplesmente fazer um grande número de documentos disponíveis como arquivos de pesquisa que poderiam ser baixados para computadores individuais, ele sugeriu que os documentos pudessem ser ligados uns aos outros, em outras palavras, poderia haver referências cruzadas a partir de um trabalho de pesquisa para outro. Isto significaria que, enquanto se fazia a leitura de um trabalho de pesquisa, poderia exibir rapidamente parte de outro documento que contém textos ou diagramas diretamente relevantes. Documentação de natureza científica e matemática, assim, ser representada como uma 'teia de informações' mantidas em formato eletrónico em computadores em todo o mundo. Este pensamento de Tim, poderia ser feito usando alguma forma de hipertexto, através de alguma forma de ligar documentos juntos usando os botões na tela, que simplesmente pudesse ser clicado para saltar de um documento para outro. Antes de vir para CERN, Tim já havia trabalhado na produção de documentos e processamento de texto, e havia desenvolvido seu primeiro sistema de

hipertexto para seu próprio uso pessoal. Foi com o advento do computador pessoal que o hipertexto saiu do papél. No final de 1980, Bill Atkinson desenvolveu um aplicativo chamado Hypercard para o Macintosh. Hypercard lhe permitiu construir uma série de cartões de apresentação na tela que continha informações textuais e gráficas. Os usuários podiam navegar nestes pressionando os botões na tela, levando-os em uma excursão da informação, no entanto, ainda haviam grandes limitações,pois ao clicar em um hipertexto só poderiam ser feitas chamadas em arquivos no mesmo computador. Por meio da década de 1980, a Internet teve um novo sistema easy-to-use para a nomeação de computadores. Isto envolveu usando a idéia do nome de domínio. Um nome de domínio compreende uma série de letras separados por pontos, por exemplo: `www.bo.com 'ou` www.erb.org.uk'. Estes nomes são de uso fácil e muito menos complicados de decorar como os números de endereços IP, desta forma Tim escolhe a abordagem certa para criar um sistema global de hipertexto, para Tim links mundiais de hipertexto pareciam viável, porém havia uma questão que deveria ser considera, usando os pacotes de hipertexto existentes poderiam parecer uma proposta atraente, mas isso era impraticável por várias razões, para começar, qualquer ferramenta de hipertexto que fosse usada em todo o mundo teria que levar em conta que foram ligados a internet muitos tipos de computadores diferentes: computadores pessoais, Macintoshes, máquinas UNIX e terminais simples. Além disso, muitos métodos de publicação de desktop estavam em alta: SGML, Interleaf, LaTex, Microsoft Word, e Troff entre muitos outros. Era necessário uma coisa muito simples, pelo menos no início. Tim demonstrou uma maneira básica, mas atraente de publicação de texto através do desenvolvimento de um software próprio, e também o seu próprio protocolo simples, o HTTP, para recuperar documentos de texto de outros através de links de hipertexto. O formato de texto para HTTP foi nomeado como HTML(HyperText Mark-up Language) e com a implementação de Tim o hipertexto foi demonstrado em uma estação de trabalho NeXT, que forneceu muitas das ferramentas que ele precisava para desenvolver o seu primeiro protótipo. Por manter as coisas muito simples, Tim encorajou outros a construir seus próprios software software para exibir o HTML, e para a criação de seus próprios documentos HTML.

2.6 OTP (ON TIME PASSWORD)

Wyld (2011) explica sobre as formas de implementação de senhas descartáveis, apresenta em sua publicação os modos de geração de sementes para base das senhas descartáveis utilizando algorítimos específicos de arvore.

Johnston (2011) demonstra através de seus artigos publicados em seu próprio site de internet que muitas aplicações que por vezes não podem utilizar criptografia devido ao fato do hardware não suportar tal técnica, pode-se mesmo assim, utilizar *JavaScript* para gerar hashs *MD5* na página do cliente, enviando desta forma ao servidor a senha em *hash* e não em texto puro, garantindo desta forma mais segurança, também demonstra uma forma interessante de resolver o problema de interceptações de trafego, trabalhando da seguinte forma, o servidor gera uma *string* aleatória e envia para o cliente juntamente com o formulário de autenticação, ao enviar o formulário de resposta o *browser* do cliente concatena a *string* com a senha do usuário e envia para o servidor, o servidor faz o mesmo processo, caso o resultado seja o mesmo o cliente é autenticado, garantindo desta forma que qualquer intercepção do trafego não servirá para nova autenticação, levando em conta que para nova autenticação um novo cálculo deverá ser feito.

3 PROCEDIMENTOS METODOLÓGICOS

Este trabalho foi desenvolvido no período de 2011 a 2012, onde inicialmente foi realizada um estudo sobre as tecnologias utilizadas em aplicações web, envolvendo as linguagens e protocolos de comunicação na web e suas vulnerabilidades.

O método utilizado para abordagem do assunto foi o dedutivo utilizando procedimento monográfico. A técnica utilizada foi a documentação indireta, com pesquisa documental e bibliográfica.

A caracterização do estudo foi bibliográfica, experimental e de campo, a análise utilizada foi a qualitativa, e os resultados são apresentados através das figuras da interface dos aplicativos.

Os seguintes procedimentos foram adotados para a elaboração deste trabalho, inicialmente foi efetuada pesquisa bibliográfica, após foi efetuado estudo nos algorítimos disponíveis para a geração de senhas descartáveis, e por fim elaborado um protótipo de software cliente (para o dispositivo móvel) e um servidor (simulando a aplicação empresarial).

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

No decorrer da história da internet, pode-se constatar, que o intuito inicial da rede, era o de constituir redundância de rotas para os computadores do departamento de defesa dos EUA, posteriormente foi crescendo e serviu como rede de troca de informações para as universidades, e por fim tornou-se comercial e hoje conta com milhares de computadores conectados a rede, porém, a estrutura inicial não visava nenhum tipo de segurança no trafego das informações, focando principalmente uma forma de interconectar redes distintas, trazendo com isso toda a vulnerabilidade encontrada hoje, pois todo tráfego http é feito em texto plano, sem nenhum tipo de criptografia.

Devido ao fato de todo trafego ser feito em texto plano, e este tráfego ter de passar por vários roteadores até chegar no servidor de destino, várias são as formas de se interceptar o tráfego, podendo ser através da instalação de software de *sniffer* (um software que captura e armazena todo tráfego de pacotes feito em uma determinada placa de rede) em um dos equipamentos que fará o repasse de pacotes, sendo através da interceptação do tráfego wifi feito através do uso de redes abertas ou com criptografia fraca.

Para resolver os problemas acima, a implementação de serviço de https (protocolo http usando criptografia) feito no servidor que hospeda a aplicação web já seria suficiente, porém, muitas vezes existe impossibilidade de se implementar https no servidor, pelo custo de se obter uma chave de uma entidade certificadora registrada nos navegadores mais populares ou por limitações no hardware como interfaces administrativas de equipamentos diversos, com o uso do https resolve-se os problemas relacionados a captura de tráfego.

Considerando que a aplicação web esta utilizando tráfego criptografado, surge outro problema de segurança que são os softwares mal intencionados muitas vezes instalados nos equipamentos do usuário final da aplicação web, conhecidos como *Keylogger* (software que capturam tudo que é digitado no teclado do computador) sendo que desta forma, é possível capturar a senha do usuário quando o mesmo está se logando na aplicação.

Para resolver os problemas de *Keylogger* muitas vezes o usuário é obrigado a clicar em um teclado virtual para digitar a senha, porém alguns aplicativos de *Keylogger* já evoluíram e capturam fotos da tela do computador e a posição do mouse na hora que os clics são feitos.

Considerando todas as atuais formas de furto de senhas e com a preocupação em se evitar os acessos indevidos a aplicações web, principalmente as aplicações onde o acesso indevido causa prejuízos muito reais como o caso das aplicações de *internet banking* é que surgiu a necessidade dos sistemas de senhas descartáveis do inglês *On Time Password*, várias são as formas de implementação.

Uma das formas mais simples é através do uso do cartão de segurança, que foi muito utilizado pelos bancos brasileiros (até o surgimento de praticas de roubo do cartão de senhas através de engenharia social, onde um e-mail solicitava que o usuário fornecesse a sequencia com todas as senhas contidas no cartão para o recadastramento do mesmo), este cartão é composto de uma sequencia de números seguido de um número aleatório como mostrado no quadro 1.

```
01 = 0154   02 = 1548   03 = 4578   04 = 9684
05 = 5489   06 = 4787   07 = 3254   08 = 5794
09 = 1456   …
```

Quadro 1: Exemplo de cartão de senhas

Após o usuário se logar da forma convencional usando usuário e senha, o sistema exibe uma tela solicitando a senha de número X, o usuário localiza a senha no seu cartão de senhas e caso a senha seja digitada corretamente, na próxima tentativa de login o sistema irá solicitar a senha de número X+1.

Outra forma utilizada, a qual será utilizada na implementação deste trabalho, é a de senhas baseadas em tempo. Esta senha segue o seguinte principio, através de uma determinada combinação de caracteres alfanuméricos conhecidos previamente pela aplicação servidora e pela aplicação cliente combinada com a data, hora e minuto atual, é feita a geração de um hash, do resultado deste hash é extraída uma parte também previamente definida nas duas aplicações, esta parte geralmente é composta de 6 caracteres, que são exibidos para o usuário na tela do dispositivo, onde o mesmo fornece para a aplicação e o mesmo calculo é feito no servidor, coincidindo os valores o servidor autentica o usuário, como a semente utilizada para gerar o hash é baseada no tempo, a cada fração de tempo a senha muda.

O Tempo para se gerar uma nova senha é definido através da fração do minuto, geralmente a cada 30 ou 60 segundos, no caso em questão iremos utilizar 60 segundos

Quanto maior o tempo para se gerar uma nova senha, maior será a tolerância a diferença de horário entre o aplicativo cliente e o aplicativo servidor, e maior será a janela de tempo para possível reutilização da senha gerada caso capturada durante o trafego, os tokens por hardware geralmente mudam a cada 36 segundos.

Durante os experimentos, foram testados cinco aparelhos quatro da Motorola sendo o v3, A1200, MB300 (backflip), MB860 (atrix) e um BlackBerry 9800. Percebeu-se que o aparelho V3 ao solicitar o horário do Meridiano de Greenwich (GMT+0) o resultado vem com 3 horas de atraso, também dependendo a operadora de telefonia usada, algumas traziam diferença de até 2 minutos em relação aos servidores de tempo do http://ntp.br

Como forma de resolver o problema, foi inclusa uma opção no software cliente de se introduzir um ajuste de tempo (em segundos), desta forma para se obter a hora correta o software utiliza a hora do sistema + o ajuste em segundos, e desta forma obtém o horário correto, no caso do V3 o ajuste foi de +3600 segundos.

O Algorítimo usado para a geração do token pode ser desenvolvido em qualquer plataforma de programação que possua suporte aos seguintes itens:

- método para capturar a data e hora atual do sistema;
- função de string para substituição de conteúdo;
- função de string para extração de partes do conteúdo;
- geração de hash md5;

O método usado para gerar o token é o seguinte:

Gerar uma string com o seguinte conteúdo: aaaa.m.d.h.m.token sendo aaaa = ano atual,m = mês, d = dia, h = hora (0 a 24), m = minuto (0 a 60) e o token do usuário (sendo esta uma string única e aleatória gerada para cada usuário)

Levando em conta que a data e hora deve ser sempre GMT0 ou seja, a hora atual do Meridiano de Greenwich para evitar conflitos de fuso horário.

passo 1: 2012.4.10.17.50.abc

Após gerada a string acima aplicar o algorítimo md5

passo 2: 613f1c2a3ad8b5de000845ab9d7d941b

Depois extrair os 6 primeiros caracteres

passo 3: 613f1c

depois substituir os caracteres a,b,c,d,e,f por 0,1,2,3,4,5 respectivamente

passo final: 613512

No quadro 2 pode-se ver o exemplo de implementação da geração do token em MYSQL.

```
SELECT
CAST(
 REPLACE(
  REPLACE(
   REPLACE(
    REPLACE(
     REPLACE(
      REPLACE(
       REPLACE(
        SUBSTR(
         MD5(
          CONCAT(
           DATE_FORMAT(UTC_TIMESTAMP(),'%Y.%c.%e.%k.'),
           CAST(DATE_FORMAT(UTC_TIMESTAMP(),'%i') AS SIGNED),
           '.',
           'abc'
          )
         )
        ,1,6)
       ,'a','0')
      ,'b','1')
     ,'c','2')
    ,'d','3')
   ,'e','4')
  ,'f','5')
AS CHAR);
```

Quadro 2: Exemplo de implementação do token em MYSQL

O Software cliente é uma aplicação jme (.jar), na ilustração 1 pode-se ver o fluxo das telas do sistemas.

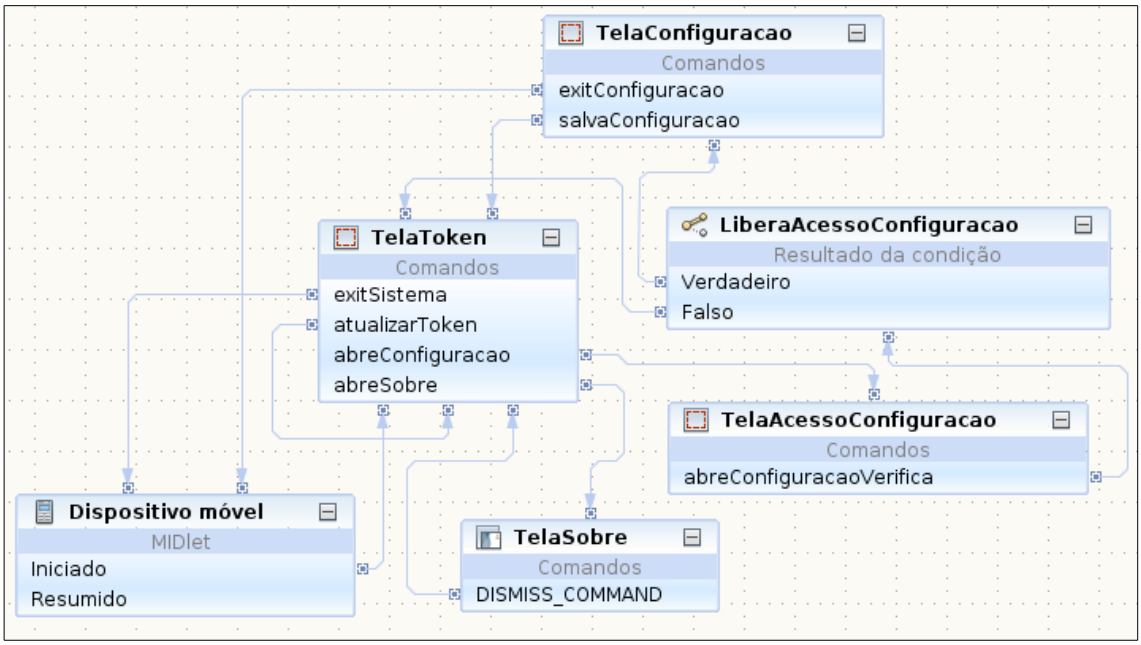

Ilustração 1: Fluxo das telas do sistemas cliente

Nas ilustrações 2, 3 e 4 pode-se ver as telas do sistema.

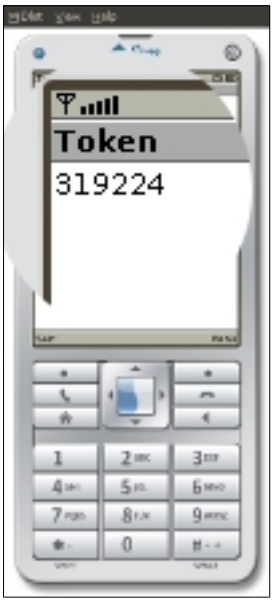

Ilustração 2: Tela inicial

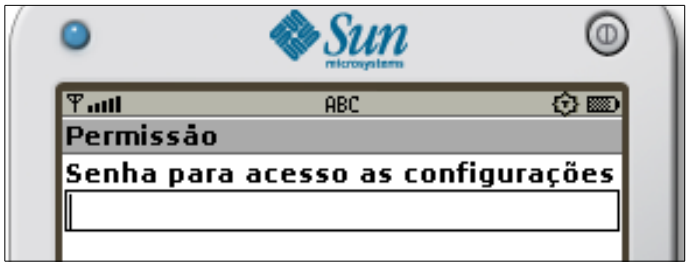

Ilustração 3: Acesso a configurações

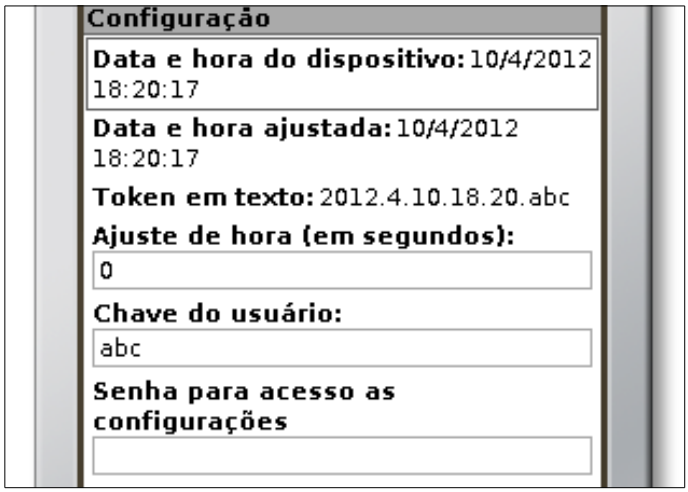

Ilustração 4: Tela de configurações

Para que o aplicativo pudesse rodar no aparelhos mb300 e mb860 foi necessário converter o aplicativo jme (.jar) para um aplicativo android (.apk) para isso foi utilizada a ferramente MicroEmulator disponível em http://microemu.org .

Nas ilustração 5 pode-se ver a tela do aplicativo sendo executado em um smartphone Motorola MB860 e na ilustração 6 o aplicativo rodando em um BlackBerry 9800.

Ilustração 5: Token rodando em Motorola MB860.

Ilustração 6: Token rodando em BlackBerry 9800.

Para fazer download do aplicativo acesse: http://www.fbasistemas.com.br/fba/2012/04/10/token/ ou faça a leitura do QRCode através da ilustração 7.

Ilustração 7: QRCode para download deste trabalho e dos aplicativos

5 CONCLUSÕES

Os objetivos foram alcançados e a aplicação esta disponível na internet através do site http://www.fbasistemas.com.br/fba/2012/04/10/token/ e pode ser utilizada e implantada por qualquer interessado.

Conforme planejado, os dois softwares foram implementados e com os testes feitos alguns ajustes foram efetuados para garantir um funcionamento minimo aceitável para utilização efetiva no mercado.

No decorrer do desenvolvimento e com a busca de informações sobre os protocolos de funcionamento da internet, pôde-se identificar que a internet possui muito mais falhas do que se imaginava no inicio.

Durante o desenvolvimento do trabalho, e buscando alternativas para o uso de https nas aplicações web em detrimento a http, encontrou-se um servidor de certificação digital que fornece certificados SSL, reconhecido pelos principais navegadores do mercado, sem custo. A empresa é StartSSL (www.startssl.com).

REFERÊNCIAS

BELLIS, Mary. The History of the Internet Disponível em: <http://inventors.about.com/od/istartinventions/a/internet.htm>. Acesso em: 29 setembro 2011.

BERNERS-LEE, Tim. Software Engineer, 1955 Disponível em:<http://designmuseum.org/design/tim-berners-lee> Acesso em: 10 novembro 2011.

BERNERS-LEE, Tim., CAILLIAU, Robert., WorldWideWeb: Proposal for a HyperText Project Novembro de 1990 Disponível em:<http://www.w3.org/Proposal> Acesso em: 10 novembro 2011.

BERNERS-LEE, Tim. ,R. Fielding, H. Frystyk Hypertext Transfer Protocol – HTTP/1.0 Maio de 1996 Disponível em:<ftp://ftp.rfc-editor.org/in-notes/rfc1945.txt> Acesso em: 10 novembro 2011.

BERNERS-LEE, Tim. Weaving the Web Setembro de 1999.

COLEMAN, David D.,David A. Westcott, Bryan E. Harkins, Shawn M. Jackman. CWSP: Certified Wireless Security Professional Official Study Guide Disponível em: <http://books.google.com.br/books?id=0ZWLn57EdpsC&pg=PA126&dq=implementation+on+time+password&hl=pt-BR&sa=X&ei=-ABdT6jSH4jdggfY2YijCw&ved=0CFcQ6AEwBQ#v=onepage&q=implementation%20on%20time%20password&f=false> Acesso em: 10 novembro 2011.

IEEE, Institute of Electrical and Electronics Engineers Disponível em: <http://www.ieee.org>. Acesso em: 10 novembro 2011.

IETF, The Internet Engineering Task Force Disponível em: <http://www.ietf.org/>. Acesso em: 10 novembro 2011.

ITU, International Telecommunication Union Disponível em: <www.itu.int>. Acesso em: 10 novembro 2011.

JOHNSTON,Paul. Paj's Home: Cryptography: JavaScript MD5: Protecting Passwords Disponível em: <http://pajhome.org.uk/crypt/md5/auth.html> Acesso em: 10 novembro 2011.

LONGMAN, Addison Wesley. A history of HTML 1998 Disponível em: <http://www.w3.org/People/Raggett/book4/ch02.html> Acesso em: 10 novembro 2011.

MANDEL, Arnaldo; SIMON Imre; DELYRA Jorge. Artigo Imformação: Computação e comunicação Publicação revista USP número 35 Informática/Internet set/out/nov 1997

ROBERTS, Lawrence G. The ARPANET & Computer Networks Maio 1995 Disponível em: <http://www.packet.cc/files/arpanet-computernet.html >. Acesso em: 30 setembro 2011.

WALDROP, Mitch. Artigo DARPA and the Internet Revolution Pag 78-85 Publicação 50 Years of Bridging the Gap 2009 Disponível em:

<http://www.darpa.mil/WorkArea/DownloadAsset.aspx?id=2554> Acessado em: 30 setembro 2011.

WYLD, David C. Advances in Network Security and Applications: 4th International Conference Disponível em: <http://books.google.com.br/books?id=fiAhDIJLayUC&pg=PA374&lpg=PA374&dq=Passblot:+A+Usable+Way+of+Authentication+Scheme+to+Generate+One+Time+Passwords&source=bl&ots=FM4sR4NXw5&sig=AwdHaul7WYxxv62X36Lou4kNojk&hl=pt-BR&ei=teGATv2tBILKgQezmplR&sa=X&oi=book_result&ct=result&resnum=2&ved=0CCkQ6AEwAQ#v=onepage&q=Passblot%3A%20A%20Usable%20Way%20of%20Authentication%20Scheme%20to%20Generate%20One%20Time%20Passwords&f=false
> Acesso em: 10 novembro 2011.