Análise de segurança em aplicativos bancários na plataforma Android

RAFAEL J. CRUZ ,

DIEGO F. ARANHA

Introdução

As inovações tecnológicas afetam diretamente a forma como as organizações atuam sendo que, segundo McCole *et al* (2010), a internet é considerada uma das principais fontes de mudança organizacional já que esta tecnologia tornou-se essencial para as atividades de negócios e na vida dos consumidores.

Um dos fenômenos organizacionais gerados pelo uso da inovação tecnológica nos negócios foi o surgimento dos serviços de mobile banking, um serviço financeiro prestado por meio de tecnologias móveis, tal como os smartphones (SHAIKH E KARJALUOTO, 2014).

Introdução

Mobile Banking refere-se a disposição e vantagem dos serviços da operação bancária e financeiros com a ajuda dos dispositivos móveis da telecomunicação. A variedade de serviços oferecidos pode incluir facilidades para realizar operações bancárias e transações do mercado acionário, para administrar clientes e para ter acesso a informações personalizadas.

Vivemos uma nova realidade. Hoje temos disponíveis dezenas de aplicativos bancários para dispositivos móveis. O acesso ao saldo, extrato, pagamento de contas, transferência de valores, investimentos etc, está ao alcance de todos com um smartphone em mãos, incluindo estelionatários e/ou Crackers (indivíduos que invadem sistemas e quebram a segurança de forma ilegal ou sem ética).

Quanto mais utilizarmos o mobile banking, mais estelionatários tentarão aplicar seus golpes. Ou seja, quanto maior a escala das transações financeiras móveis, novas ameaças surgirão e maior serão os número de vítimas de crime cibernético. Por exemplo, um dos maiores banco no Brasil registra aproximadamente 8 milhões de transações móveis por dia.

Problemas associados

Segundo a FEBRABAN,

- 24% dos clientes utilizam Mobile Banking
- Cerca de 1.4 Bilhão foi perdido devido a fraudes em 2012 (Cerca de 0.007% do total envolvido)

Segundo a Symantec,

- 38% dos usuários de dispositivos móveis foram vítimas de crime cibernético nos últimos 12 meses;
- 97% dos códigos maliciosos identificados foram desenvolvidos para atacar dispositivos com o sistema operacional Android;
- 82% das vulnerabilidades foram localizadas no iOS da Apple;
- 28% dos códigos maliciosos foram programados para espionar as informações armazenadas no dispositivo móvel;
- 17% dos códigos maliciosos roubam dados no dispositivo móvel.

Pontos de ataque

Dados sensíveis armazenados no dispositivo móvel;

Falta de criptografia ou criptografia fraca durante a transmissão dos dados, tais como:

- agência,
- conta corrente
- senha do banco;

Falhas no processo de validação do Secure Socket Layer (SSL);

Acesso privilegiado de forma indevida ao aplicativo móvel para roubar dados armazenados em memória ou fazer engenharia reversa.

Técnicas de ataque ao mobile banking

No Brasil, o acesso a conta corrente através do mobile banking possui duas camadas de segurança importantes: uma é a senha eletrônica e a outra é o Token. Sendo assim, temos alguns ataques direcionados a autenticação:

1 – SSL Proxy

Um dos ataques preferidos pelos estelionatários para espionagem de informações confidenciais. Consiste em interceptar toda a comunicação entre o cliente e o banco, incluindo a utilização de um certificado digital falso, criando uma “conexão segura” entre o cliente do banco e o atacante. Dessa forma, o atacante consegue capturar a agência, conta corrente, senha eletrônica, senha do cartão de débito e o token.

SSL Strip e Man-in-the-middle

Este tipo de ataque é executado para roubo de senhas, tokens, nomes de usuário, agencia, conta corrente, palavras-chave e outros dados sensíveis;

Mesmo quando as informações estão criptografadas ou codificadas. Na prática, são utilizadas algumas ferramentas para sequestrar o tráfego de informações entre o mobile banking instalado no smartphone/tablet e o banco, monitorando sempre as solicitações via HTTPS;

Depois eliminamos o SSL deixando o cliente do banco em uma conexão não segura. Dessa forma, o número da agência, conta corrente e senha eletrônica são capturadas pelo atacante através de uma transmissão não segura das credenciais do banco.

Certificados digitais falsos

Outro tipo de ataque baseado em Man-in-the-middle onde o atacante emite um certificado digital falso tendo como objetivo interceptar o tráfego em uma conexão HTTPS supostamente segura.

Evitar certificados falsos : Para evitar este problema, as boas praticas de segurança determinam então, que o aplicativo seja capaz de detectar a validação indevida do certificado falso, a partir da implementação de uma técnica de pinagem de chave pública, ou seja, qualquer informação relevante sobre os certificados do servidor armazenada no cliente, por exemplo: a autoridade certificadora ou hash da cadeia de certificados.

Engenharia reversa e análise de código

A engenharia reversa consiste em ter acesso ao código-fonte de um aplicativo móvel. Através de uma combinação de ferramentas e técnicas é possível identificar senhas pessoais armazenadas no aplicativo ou dispositivo móvel, permissões de acesso, certificados utilizados pelo aplicativo móvel e se o código está ofuscado.

Serviços protegidos

Desta forma é necessário a produção de aplicativos eficientes e seguros, sempre atualizados de acordo com ataques recentes.

Protocolo SSL/TLS

Evitar um ataque MITM : Implementar de maneira correta o protocolo SSL/TLS

Boas práticas

Suporte a segredo futuro,

Atualização de algoritmos,

Validação de certificados,

Atualização de protocolos,

Ataques conhecidos

BEAST

- O BEAST explora uma vulnerabilidade no cipher block chaining (CBC) no protocolo TLS v1.0 e foi descoberto em 2002, porém ele só foi aplicado na prática em 2011, onde se conseguiu explorar com sucesso a vulnerabilidade. Como é usado em determinadas configurações no Microsoft Windows e Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera e outros produtos, o CBC é utilizado para criptografar os dados. Quando esta vulnerabilidade é explorada, permite a realização de ataques do tipo man-in-the-middle (MiTM).

Ataques conhecidos

CRIME

- Os autores do BEAST também são os criadores do CRIME, este ataque permite que um invasor recupere o conteúdo de cookies da Internet quando a compressão de dados é utilizada junto com TLS. Por exemplo, este ataque pode ser usado para recuperar o conteúdo de um cookie de autenticação, permitindo a um invasor realizar o sequestro (session hijacking) de uma sessão web. Estes ataques são os mais conhecidos, mas ainda podemos destacar os ataques de Padding, RC4, Truncation e Forward secrecy, todos estes também ligados ao protocolo SSL ou seu sucessor, o TLS. Esteja sempre atento às melhores práticas para configurar de forma segura o seu servidor HTTPS e mantenha sempre seu navegador web atualizado.

Ataques conhecidos

FREAK

- Também chamada de “Factoring Attack on RSA-EXPORT Keys”, a vulnerabilidade permite que invasores consigam interceptar conexões HTTPS entre clientes e servidores vulneráveis e forçá-los a utilizar uma criptografia fraca, conhecida como “export-grade key” ou “512-bit RSA keys”, que pode, então, ser decifrada. A conexão é vulnerável se o servidor aceitar a suite de cifras “EXPORT” ou estiver utilizando uma das versões vulneráveis do OpenSSL.

Ataques conhecidos

POODLE

- O POODLE se aplica somente no protocolo SSL 3.0, que foi lançado em 1996. O SSL 3.0 já se tornou obsoleto quando o TLS 1.0 foi lançado em 1999, e o protocolo mais novo é o TLS 1.2 (2008). Existe também um draft para o TLS 1.3 (possivelmente será lançado em 2015). O SSL 3.0 utiliza criptografia RC4 (cifrador de fluxo) ou criptografia com cifrador de bloco no modo CBC. O RC4 possui fragilidades conhecidas que resultaram em ataques práticos no ano passado. O modo CBC, como usado no SSL 3.0, já era problemático e resultou em ataques em 2013. O POODLE explora outra falha no CBC como usado no SSL 3.0, porém é um ataque mais fácil de se executar do que os anteriores. Além disso, ao contrário dos ataques anteriores, o POODLE é uma falha no protocolo em si e não na sua implementação, portanto o SSL 3.0 pode ser considerado quebrado. O problema é que vários sistemas legados utilizam esse protocolo, então simplesmente desabilitá-lo é complicado.

Metodologia

Para o desenvolvimento do estudo sobre a vulnerabilidade de aplicações Mobile Banking no ambiente Android, foi criada uma metodologia de ataque com base nas principais falhas dos protocolos SSL/TLS além das principais técnicas de ataque, com foco de verificar quais aplicativos estão susceptíveis a estes ataques.

Parte 1

Consiste na análise dos códigos e identificação dos servidores de cada aplicação bancária

Para isto foram utilizadas algumas ferramentas como:

- Wireshark
- APK downloader
- Dex2jar
- JD-GUI
- Monitor-SDK

Esta parte tem como função explorar as aplicações Android, transformando-os em arquivos java, e desta forma, com a utilização de ferramentas *sniffer,* obter *hostnames* dos servidores e *logs* dos aplicativos.

Parte 2

Consiste no ataque por meio da técnica (Man in the Middle), para esta parte são utilizadas:

- Iptables
- Arpspoof
- OpenSSL
- Sslsplit

Esta parte tem como função redirecionar o tráfego da rede para um computador malicioso, com a utilização do iptables, após isso, na rede local, pode-se controlar a rota do cliente através de um proxy ou utilizando arpspoof.

Os ataques de arp spoofing consistem em adicionar/substituir na tabela arp da maquina alvo uma entrada que diz IP_QUE_A_MAQUINA_ALVO_ESTA_SE_COMUNICANDO <===> MAC_DO_ATACANTE. Com isso quando a maquina alvo for montar o pacote para envio ela montara com o IP real do servidor de destino que ela quer acessar, porem utilizará o endereço MAC do atacante, ou seja, quando este pacote passar pelo switch o mesmo encaminhará o pacote para o atacante, no caso você.

Após isso, pode-se gerar um certificado com o auxílio do OpenSSL, e combinado com o sslsplit, pode-se trocar informações com na conexão SSL/TLS, instalando o certificado forjado.

Parte 3

Exame dos servidores identificados

- SSLlabs

Após ter todo o processo de identificação e instalação completo (Parte 1 e 2), pode-se por meio do ssllabs, fazer exames semanais das configurações do servidor, e identificar o que está desatualizado no servidor, e assim criar um ataque específico.

Resultados

Tabela 1. Resultado do SSLlabs sobre os servidores examinados. As notas variam de F a A+.

	Banco do Brasil	Bradesco	Caixa Econômica Federal	Citibank	HSBC	Itaú	Santander
😊 Emprega TLS 1.2	✘	✘	✔	✘	✔	✔	✘
😊 Emprega TLS 1.1	✘	✘	✘	✘	✔	✔	✘
😐 Emprega TLS 1.0	✔	✔	✔	✔	✔	✔	✔
☹ Suporta SSL 3.0	✔	✔	✔	✘	✔	✔	✔
☹ Suporta SSL 2.0	✘	✔	✘	✘	✘	✘	✘
☹ Suporta RC4	✔	✔	✔	✘	✘	✔	✔
☹ Suporta MD5	✔	✔	✘	✘	✘	✘	✘
😐 Suporta SHA-1	✔	✔	✔	✔	✔	✔	✔
😊 Suporta Segredo Futuro	✘	✘	✘	✘	✘	✘	✘
😊 Suporta OCSP	✘	✘	✘	✘	✔	✘	✘
😊 Suporta HPKP	✘	✘	✘	✘	✘	✘	✘
☹ Diffie-Hellman Inseguro	✔	✘	✘	✘	✘	✘	✘
☹ Vulnerável a Deterioração	✔	✔	✘	✘	✔	✔	✔
☹ Vulnerável ao POODLE	✔	✘	✔	✔	✔	✔	✘
☹ Vulnerável ao DoS	✘	✘	✔	✔	✘	✘	✘
☹ Vulnerável ao FREAK	✔	✔	✘	✘	✘	✘	✘
Nota	F	F, F	C	C	C, A-, A-	F	C

Propriedades: 😊 (Boa) | 😐 (Neutra) | ☹ (Ruim) || ✔ Sim/Aplica-se | ✘ Não/Não se Aplica.

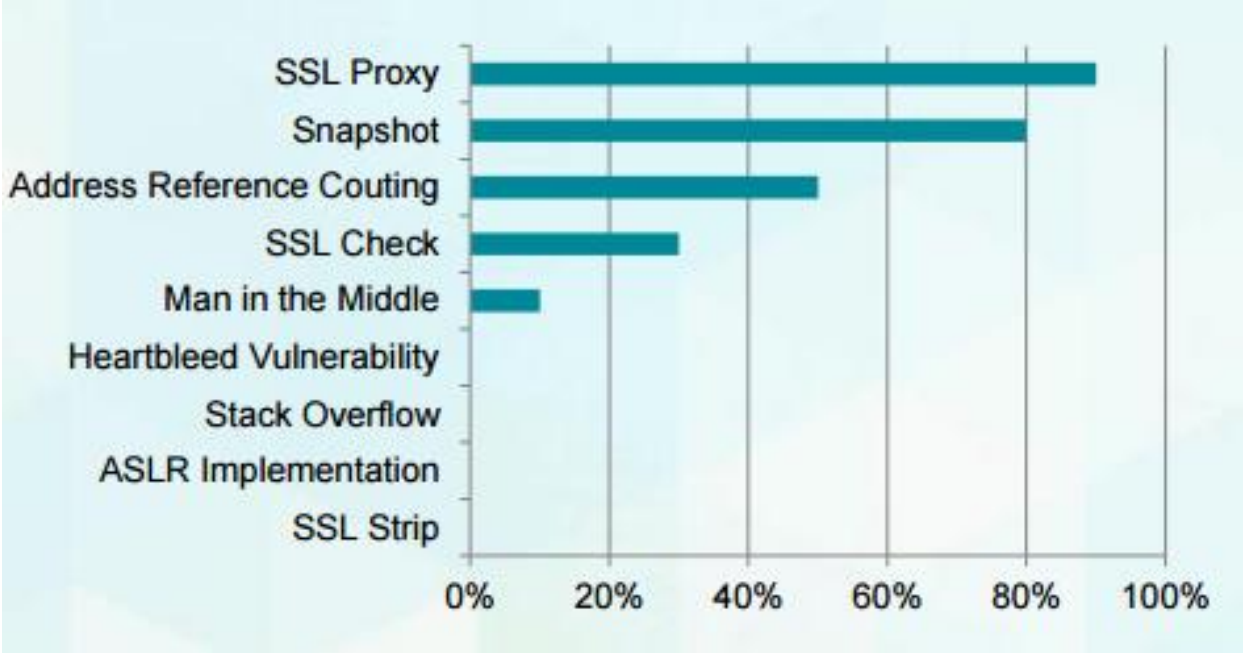
Resultados

Tabela 2. Resultado dos aplicativos Android analisados quanto à vulnerabilidade a ataques de personificação e outras decisões de projeto. As notas variam de 0 a 5 estrelas.

	Banco do Brasil	Bradesco	Caixa Econômica Federal	Citibank	HSBC	Itaú	Santander
Versão analisada	6.5.0.7	2.9.6	1.3.3	9.0	1.5.10.0	4.1.5	4.4.0
Não sofre MITM comum	✔	✔	✔	✔	✔	✔	✘
Não sofre MITM com certificado raiz	✔	✘	✘	✘	✘	✘	✘
Não Vaza credenciais	✔	✘	✘	✘	✔	✘	✘
Não Vaza informações financeiras	✔	✘	✘	✘	✘	✘	✘
Não contém redes sociais externas	✔	✘	✔	✘	✔	✘	✔
Utiliza pinagem de chave pública	✔	✘	✘	✘	✘	✘	✘
Nota	★★★★★	★½	★★	★½	★★★	★½	½

✔ Sim/Aplica-se | ✘ Não/Não se Aplica.

Outra análise



RESULTADOS

Como forma de conseguir resultados para validar o artigo, foram feitas análise nos aplicativos reais de bancos. Desta forma chegamos aos seguintes resultados

Banco do Brasil

- Apesar de obter a pontuação mais alta dentre os aplicativos, ainda possuem falhas de atualização dos serviços, pois utiliza o protocolo SSL 3.0 e algoritmos RC4, MD5 e SHA-1, estas configurações permitem ataques.

Bradesco

- O conjunto aplicativo-servidores mostrou-se muito vulnerável, para deixar de ser vulnerável aos ataques deste trabalho, o Bradesco deve realizar algum tipo de pinagem no cliente ou servidor (ou ambos). O servidor tem compatibilidade com protocolos muito antigos, como SSL 2.0 e SSL 3.0, além de utilizar algoritmos inseguros, como RC4, MD5 e SHA-1. Um outro problema de segurança e a integração do aplicativo com redes sociais externas (como Facebook), o que torna um alvo fácil para ataques.

Resultados

Caixa Federal

- O aplicativo ainda precisa realizar a pinagem de chave publica, assim como o Bradesco. O servidor suporta SSL 3.0 e a falta de uma renegociação segura de chaves permite o ataque DoS. Um outro problema de segurança é o excesso de utilização do JavaScript, que fica armazenado no cliente de maneira transparente, permitindo uma possível injeção de código.

Citibank

- A situação do aplicativo é análoga ao Bradesco. A falta de renegociação segura de chaves permite o ataque DoS e a não verificação do preenchimento restante da mensagem permite a utilização do POODLE em TLS.

Itaú

- Situação análoga ao Bradesco. O servidor utiliza SSL 3.0 e algoritmos inseguros, como RC4 e SHA-1, além de permitir ataques de deterioração e POODLE.

Resultados

HSBC

- Apesar de sofrer o ataque MITM com certificado autoassinado, não transmitiu de maneira clara – sem cifrar – as credenciais do cliente, pois a habilitação de um dispositivo móvel permite negociação prévia de chaves criptográficas. Entretanto, algumas informações financeiras dos clientes foram vazadas no trafego capturado, como saldo e limite do cartão de crédito, por exemplo. Ainda assim, é necessário realizar pinagem de chave pública, como o Bradesco. Do ponto de vista de segurança, um dos servidores precisa de uma melhor configuração.

Resultados

Santander

- Dada a vulnerabilidade a ataques MITM comuns, o aplicativo opta por utilizar um protocolo adicional customizado na camada de aplicação, ao invés de utilizar somente o protocolo SSL/TLS. Nesse protocolo, o servidor envia uma chave publica RSA de 2048 bits que e utilizada pelo cliente para cifrar uma chave pública. efêmera de 1024 bits, gerada a cada conexão. Ao recuperar a chave pública efêmera utilizando sua chave privada, o servidor cifra uma chave de sessão para proteger a comunicação subsequente com o cliente.
- Verificou-se por engenharia reversa do aplicativo que o protocolo customizado também não faz nenhuma autenticação de chaves públicas, tornando-se trivialmente vulnerável ao ataque MITM. A correção pode ser feita ao não depender do protocolo customizado e utilizar o protocolo TLS de maneira correta, pois desta maneira, haverá pelo menos dois benefícios: ganho de desempenho por ter apenas um protocolo utilizado e correção dos ataques detectados por este trabalho. O servidor utiliza o protocolo SSL 3.0, mas apesar da correção do ataque POODLE, e necessário abandonar este protocolo. O servidor ainda utiliza algoritmos inseguros, como RC4 e SHA-1.

Conclusão

Espera-se que os resultados deste trabalho sejam uteis para aprimoramento de segurança de aplicativos bancários, pela adoção de duas medidas principais:

- Aplicações do lado do cliente precisam avaliar cuidadosamente chaves públicas do servidor enviadas durante a conexão. Durante a validação feita pelo cliente, deve-se também checar CRL, OCSP e pinagem de certificado.
- Servidores também devem aumentar o nível de segurança. Este objetivo pode ser alcançado abandonando algoritmos e protocolos criptográficos obsoletos, além de implementar novas medidas de segurança (Segredo Futuro, por exemplo)

Conclusão

Os principais desafios dos bancos são a prevenção e a resposta aos incidentes das transações fraudulentas, a mitigação de riscos relacionados ao roubo de identidade e de dados bancários e a gestão da segurança da informação para aplicativos móveis.

A preocupação com a arquitetura, desenho, implementação e configuração dos requisitos de segurança em aplicativos móveis tem aumentado.

Os bancos elaboraram políticas e normas para gestão de conformidade e resposta a incidentes relacionados a segurança e fraude em aplicativos móveis.

Conclusão

As instituições financeiras estão cada vez mais avaliando possíveis vulnerabilidades no mobile banking, bem como a eficiência dos controles de segurança presentes para a proteção de autenticação, tráfego de rede e engenharia reversa.

Detectar vulnerabilidades, identificar vazamento de dados, detectar problemas na criptografia, examinar a segurança no método de autenticação e detectar possíveis falhas no código do mobile banking que podem levar a exploração de vulnerabilidade tornaram-se uma das atividades do dia a dia das equipes de Auditoria, Tecnologia da Informação, Prevenção a Fraudes e Segurança da Informação. Porém, encontramos alguns cenários onde a equipe de desenvolvimento de software não utiliza as boas práticas de desenvolvimento seguro para aplicativos móveis.

Não se esqueça! Os criminosos quase sempre estão alguns passos a frente dos sistemas de segurança.

Conclusões

Como segundo objetivo, o artigo buscava compreender como a conveniência e segurança interfere no comportamento de intenção de uso do mobile banking.

Constatou-se que apenas a conveniência (dimensão de benefício) é um antecedente significativo da intenção de uso. Ou seja, os resultados comprovaram que os universitários pesquisados não levam em consideração os custos (dimensão segurança) envolvidos na adoção do mobile banking.

Esta descoberta vai contra os achados de diversas pesquisas citadas no referencial teórico deste estudo.