Hacking Ginga: uma avaliação de segurança da plataforma de aplicações interativas da TV digital brasileira

Alexandre Melo Braga[1], Gilmara Santos Restani[1]

[1]Fundação CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações
Rod. Campinas-Mogi-Mirim (SP-340) km 118,5 – 13086-902 – Campinas – SP – Brazil

ambraga@cpqd.com.br, grestani@cpqd.com.br

Abstract. *This paper presents preliminary results obtained by CPqD concerning security assessments performed on some technologies of the Brazilian interactive digital television (TVDi). Particularly, the interactive applications and set-top boxes with embedded Ginga-NCL and Lua. Several vulnerabilities of insecure programming were identified and documented. Vulnerability scans and security tests of set-top boxes revealed serious vulnerabilities not only on the configuration of underlying systems, but also on the protection of applications.*

Resumo. *Este artigo apresenta resultados preliminares obtidos pelo CPqD na avaliação de segurança de algumas tecnologias da televisão digital interativa (TVDi) brasileira. Em particular, as aplicações interativas e os receptores de TVDi de prateleira embarcados com Ginga-NCL e Lua. Foram identificadas e documentadas diversas vulnerabilidades de programação insegura. Varreduras de vulnerabilidades e testes de segurança em receptores revelaram vulnerabilidades graves tanto de configuração dos sistemas subjacentes quanto de proteção de aplicações.*

1. Introdução

A interatividade no ambiente de radiodifusão digital traz oportunidades para a oferta de serviços. A grande capilaridade da TV nos lares brasileiros e o oferecimento de uma plataforma de computação e comunicação de baixo custo para a população, o receptor digital interativo, fazem do Sistema Brasileiro de Televisão Digital (SBTVD) uma ferramenta poderosa não somente na promoção da inclusão digital e da cidadania, mas também na geração de riqueza pela facilitação de transações comerciais.

A possibilidade de realização de transações comerciais no ambiente da televisão digital interativa tem desafios grandes quanto à segurança das transações e à guarda das informações. Por exemplo, a segurança de uma transação bancária realizada por meio da televisão digital interativa teria que oferecer ao usuário final pelo menos a mesma confiança que a transação equivalente realizada a partir de um computador pessoal em website de comercio eletrônico na Internet.

No âmbito do SBTVD, a plataforma de software da Televisão Digital Interativa (TVDi), oferecida pelo middleware Ginga, poderá se tornar um ambiente propício para a realização de transações comerciais; em particular, as de comércio eletrônico pela

televisão (*t-commerce*), *homebanking* pela televisão (*t-banking*) e governo eletrônico pela televisão (*t-gov*).

Por outro lado, os receptores de TVDi tendem a se tornarem, nos próximos anos, dispositivos de consumo programáveis por seus usuários, tal e qual são os telefones celulares e computadores domésticos e portáteis. Porém, algumas das linguagens componentes do middleware Ginga foram idealizadas para projetos modestos em tamanho e de complexidade reduzida. Neste sentido, o aumento da exposição destas linguagens, pela sua disponibilidade em receptores de TVDi, seria uma extrapolação inesperada do uso previsto originalmente. Por isto, muitas vulnerabilidades latentes podem emergir devido ao novo cenário de ameaças revelado pela nova utilização.

Conforme Barbosa e Soares (2008), o middleware Ginga é uma combinação de tecnologias padronizadas e inovações brasileiras. Ele é subdividido em três componentes principais interligados. Os subsistemas são chamados de Ginga-J (para aplicações procedimentais Java), Ginga-NCL (para aplicações declarativas NCL e NCLua) e o Ginga-CC (o núcleo comum). Além disso, há diversas APIs, pacotes de software e outros serviços de mais baixo nível, oferecidos ao middleware pelo sistema operacional ou outros componentes.

Este trabalho foi direcionado aos aspectos de segurança da linguagem de programação Lua, sua biblioteca padrão e algumas poucas bibliotecas externas, bem como à avaliação de segurança de receptores de TVDi embarcados com Ginga-NCL e Lua. Esta decisão de escopo se deve ao fato de, até o momento da escrita deste texto, não haver uma implementação comercial do Ginga-J disponível para a equipe de avaliação de segurança. O trabalho foi realizado sobre implementações do Ginga-NCL, com a ponte NCLua, e Lua 5.1. Ainda, o texto é principalmente voltado para proteção das aplicações e da privacidade e integridade dos dados de usuários finais.

O texto está organizado da seguinte forma. Na seção 2 é apresentado um breve resumo histórico das contribuições do CPqD para a televisão digital brasileira. Na seção 3 os autores fazem comentários sobre trabalhos diretamente relacionados. Na seção 4 os autores apresentam aspectos gerais de segurança em TVDi e identificam categorias de ameaças. Na seção 5 são mostrados alguns resultados preliminares na identificação e na documentação de vulnerabilidades da linguagem de programação Lua. Na seção 6 os autores apresentam os resultados experimentais de avaliação de segurança realizados sobre receptores TVDi de prateleira. E por fim, na seção 7 são apresentadas as considerações finais.

2. O CPqD e a TV digital interativa

O CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações, antigo braço tecnológico do Sistema Telebrás e hoje fundação autônoma, possui vasta experiência no tema TV Digital e suas contribuições são resumidas a seguir.

Desde de 1985, tem sido prestadas diversas assessorias ao Sistema Telebrás, à Comissão de Comunicação, Ciência e Tecnologia, à Anatel e ao Ministério das Comunicações. Com a criação, pelo Governo brasileiro, do projeto SBTVD – Sistema Brasileiro de TV Digital – o CPqD passou a prestar apoio técnico e administrativo ao Grupo Gestor do Sistema Brasileiro de TV Digital, com o objetivo de realizar atividades

de pesquisa, desenvolvimento e análises, em duas vertentes. Primeira, as análises para a elaboração do Modelo de Referência para a implantação da TV Digital Terrestre no Brasil. Segunda, a elaboração do modelo tecnológico sistêmico da TV Digital e coordenação dos trabalhos de P&D realizados por consórcios universitários. Após a definição do Sistema Brasileiro pelo governo, o CPqD dedicou-se a contribuir com o processo de implantação da TV Digital no Brasil, por meio do desenvolvimento e testes de serviços e aplicativos. O CPqD desenvolveu as primeiras provas de conceito do uso do terminal de TV Digital como plataforma de fruição de serviços interativos plenos, independentes das redes utilizadas. Ainda, para a realização de testes, foi implantado um laboratório multiplataforma de TV Digital. Além disso, foi desenvolvida a prova de conceito de uma plataforma compacta de TV Digital, para minimizar os custos de digitalização de emissoras pequenas.

Este artigo é derivado do trabalho realizado no projeto de pesquisa aplicada Serviço Multiplataforma de TV Interativa – SMTVI, na meta Serviço T-Commerce (M6). Em relação aos aspectos de segurança para TV Digital brasileira, entre outros trabalhos, o CPqD tem atuado no Fórum Brasileiro de TV Digital e colaborado na elaboração da norma brasileira neste assunto (ABNT NBR 15605).

3. Trabalhos relacionados

Foram realizadas diversas pesquisas por trabalhos sobre segurança de aplicações interativas para a plataforma brasileira de TVDi e tecnologias associadas. Das bibliografias encontradas, detalhadas a seguir, há trabalhos sobre proteção de conteúdo e mídia digital. Porém, há relativamente poucos trabalhos sobre proteção de aplicações, sendo na sua maioria implementações de mecanismos de segurança específicos. Em relação à segurança para proteção de conteúdo, a norma ABNT NBR 15605 especifica os mecanismos do sistema de segurança para o sistema brasileiro de televisão digital terrestre. A primeira parte desta norma, ABNT NBR 15605-1 (2008), trata de controle de conteúdo; isto é, DRM (*Digital Rights Management*). Carvalho et al (2007) descreve a implementação de um mecanismo de proteção criptográfica e esteganográfica da integridade de arquivos de mídia digital em receptores de TVDi brasileiros, embarcados com o sistema operacional Linux.

Em relação à segurança de aplicações interativas, a norma ABNT NBR 15605-2 (em preparação) define os mecanismos de autenticação dos receptores, dos dispositivos externos e de usuários, além das questões de segurança e autenticação de aplicativos interativos, assim como do canal de interatividade. Honorato e Barbosa (2010) propuseram uma ferramenta para inspeção de código fonte NCL, para reforçar o uso de estilos de programação e o controle de qualidade do código fonte.

Os pacotes de software LuaCrypto (luacrypto.luaforge.net) e LuaSec (www.inf.puc-rio.br/~brunoos/luasec) são implementações de interfaces Lua para funções criptográficas disponíveis no receptor e protocolos de comunicação segura, como o SSL, respectivamente. O pacote de software LuaMD5 (www.keplerproject.org/md5) é uma biblioteca criptográfica simples para scripts Lua. A Comunidade de Usuários Lua (Lua-users.org) contém algumas recomendações curtas e genéricas sobre o uso de APIs Lua consideradas perigosas e sobre o uso de sand-boxing como estratégia de contenção de código.

4. Aspectos de segurança em TVDi, receptores e aplicações interativas

Esta seção analisa as questões de segurança relacionadas ao receptor de TVDi e ao Middleware Ginga. Certamente ameaças e vulnerabilidades emergirão de implementações específicas do middleware Ginga em receptores TVDi de prateleira, assim como também das aplicações interativas construídas para, e executadas sobre, estas implementações. A possibilidade de realização de transações comerciais pelo receptor de TVDi põe em evidência as questões de segurança nos receptores.

A Figura 1, inspirada em Ravi et al (2004), ilustra as necessidades de segurança de cada participante da cadeia de valor da TVDi, desde a produção de hardware, passando pela geração de conteúdo televisivo, até o oferecimento de serviços interativos, vinculados à programação, para o usuário final. Na Figura 1, no lado esquerdo (borda quadrada) estão as ameaças mais comuns associadas a cada participante. No lado esquerdo, o requisito de segurança correspondente à mitigação da ameaça em questão.

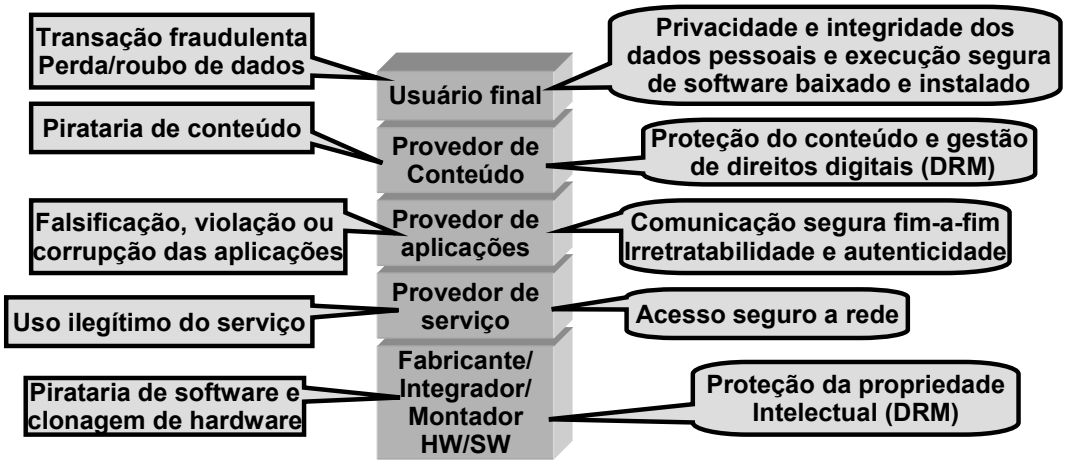

Figura 1: Necessidades de segurança na cadeia de valor da TVDi brasileira.

O desafio em relação às linguagens de programação e plataformas de software do receptor de TVDi brasileiro está em oferecer a estas tecnologias o mesmo grau de confiança de que gozam as linguagens e plataformas mantidas pelos gigantes da indústria de software (Java/JEE, C#/.NET, entre outros) para computação de massa.

Não é feita a afirmação de que a tecnologia empregada nos receptores de TVDi brasileiros seja mais insegura ou inferior aos produtos em moda na indústria de software. Por outro lado, é fato que por questões fora do escopo deste texto as implementações de linguagens de programação da plataforma de TVDi não sofreram o mesmo escrutínio daquelas desenvolvidas pelos grandes produtores mundiais de software. Além disso, por ainda não serem tão universais quanto os sistemas de mercado, as plataformas de software e linguagens de programação da TVDi brasileira não sofreram a mesma exposição às ameaças comuns aos sistemas de Internet que outras plataformas.

4.1. Segurança comparada: receptores de TVDi e smartphones

Pode ser feito um paralelo entre a evolução das técnicas de ataque ocorrida nos PCs conectados a Internet e, mais recentemente, nos aparelhos de telefones celulares inteligentes (*smartphones*), e os receptores de TVDi. Conforme ilustrado pela Figura 2.

A tendência mundial de evolução das técnicas de ataque aos smartphones e às redes subjacentes de telecomunicações é comparável ao estado das técnicas de ataque sobre PCs conectados a Internet. No caso dos PCs, foi na década de 80 que a evolução das técnicas de ataque, em particular, os softwares maliciosos, começou a afetar o público em geral. Atualmente existem redes inteiras comprometidas e controladas pelo crime organizado para realização dos mais diversos tipos de ataques, incluindo os ataques maciços coordenados.

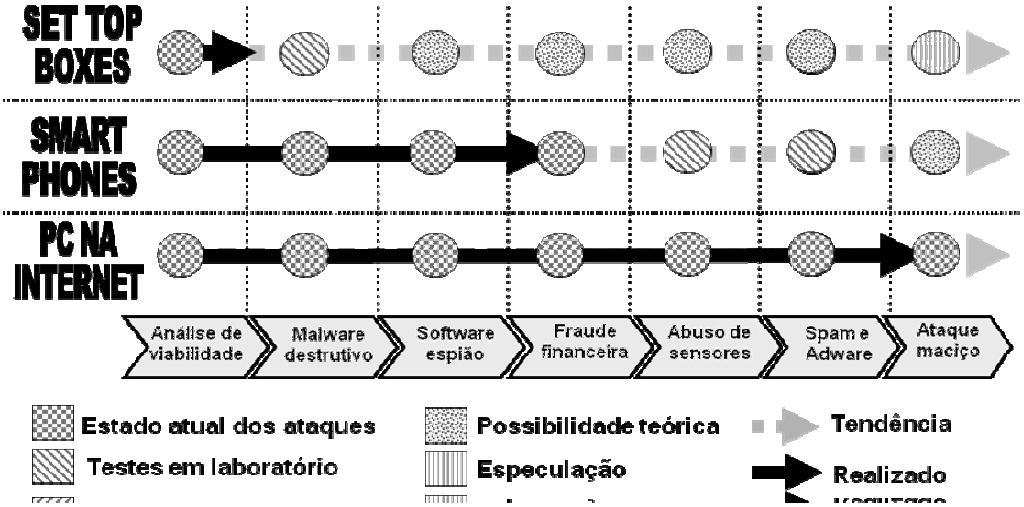

Figura 2: Evolução dos ataques – PCs, smartphones e receptores de TVDi.

De acordo com Hypponen (2007), o primeiro vírus de smartphone surgiu em 2004, desde então há relatos de centenas de softwares maliciosos para estes aparelhos, conforme Bickford, et al (2010). Os ataques mais sofisticados às redes via smartphones, com o controle remoto de diversos aparelhos para a realização de ataques coordenados, já são considerados, em textos acadêmicos, uma possibilidade técnica, conforme Traynor, et al (2009). Segundo Oberheide and Jahanian (2010), Oberheide, et al (2008) e Cai, Machiraju and Chen (2009), os smartphones serão uma fonte de vazamento de informações privadas em redes públicas e ambientes de computação em nuvem, serão ainda vetores de ataques maciços às redes de computadores e de telecomunicações, assim como representarão a próxima fronteira de proliferação dos softwares maliciosos.

De forma análoga ao que tem ocorrido com os PCs e ao que ocorre com os smartphones, espera-se ainda nesta década um aumento significativo da quantidade de incidentes de segurança envolvendo os receptores de TVDi brasileiros. Neste sentido, os estudos e experimentos realizados pelo CPqD levam a um cenário no qual os receptores de TVDi, como o smartphones, poderão ser vetores de ataque. Por outro lado, a semelhança entre os aspectos de segurança de smartphones e de receptores de TVDi vão

além do paralelismo das técnicas de ataque, abrangendo também, com grande similaridade, as técnicas de proteção.

Para Oberheide and Jahanian (2010), as plataformas de software modernas para smartphones implementam uma arquitetura de segurança baseada em três pilares. Primeiro, a entrega segura de aplicações, a qual está relacionada à habilidade de uma plataforma em verificar a integridade e a autenticidade de origem de uma aplicação a ser instalada no dispositivo. Segundo, níveis de confiança, que determinam graus de segurança e privilégios por mecanismos de controle de acesso. Terceiro, o isolamento de aplicações e do sistema operacional, que se refere à habilidade de uma plataforma em isolar uma aplicação em particular, como uma estratégia de prevenção contra o comprometimento de outras aplicações ou o próprio sistema operacional.

O modelo de segurança adotado pela norma brasileira ABNT NBR 15605-2 (2008) não é fundamentalmente diferente daquele adotado pelos fabricantes de dispositivos móveis. Não por acaso, a norma brasileira de segurança de aplicações interativas para TV digital oferece uma arquitetura de segurança semelhante em diversos aspectos à arquitetura de segurança das plataformas móveis modernas. De modo análogo ao oferecido pelas plataformas de aplicativos para dispositivos móveis, um receptor de TVDi, com middleware Ginga, aderente a norma brasileira, poderá possuir as seguintes características de segurança: autenticação de usuário, segurança no canal de interatividade com SSL/TLS e suporte à autenticação de aplicativos com o mecanismo de controle de acesso correspondente.

Em relação à segurança na distribuição das aplicações, a norma ABNT NBR 15605-2 (2008) estabelece que as aplicações sejam autenticadas e assinadas por entidade responsável com um certificado de identidade ICP-Brasil, em processo semelhante à assinatura de código móvel das plataformas existentes.

O mecanismo de controle de acesso defendido pela norma tem semelhantes com o modelo mandatório. A autonomia de uma aplicação interativa sempre será limitada às restrições de acesso do receptor de TV digital. Uma aplicação não autenticada nunca poderá solicitar a ampliação de sua autonomia pela modificação das restrições impostas pelo receptor. Já uma aplicação autenticada poderia solicitar permissões extras, mas ainda assim limitada, no máximo, ao conteúdo de um certificado de atributos com requisições de permissões. O mecanismo de isolamento de aplicações seria semelhante ao sand-box utilizado em outras tecnologias de código móvel.

4.2. O receptor de TVDi como sistema embarcado seguro

Enquanto sistema embarcado de hardware restrito, os receptores de TVDi de prateleira disponíveis atualmente são extremamente limitados em memória de trabalho, armazenamento de dados e processamento de informação. Os receptores de TVDi, na segurança de aplicativos, são similares aos outros sistemas embarcados, como por exemplo os telefones celulares, apesar daqueles não dependerem de baterias.

Conforme Kocher et al (2004) e Ravi et al (2004), as restrições de hardware impõem limitações severas ao desempenho de mecanismos de segurança. Sendo um dos motivos da grande dificuldade no oferecimento de serviços de segurança plenos. De fato, as restrições de memória dos receptores de TVDi, experimentadas durante este trabalho, representaram um desafio grande para a execução de testes de segurança.

Vale ressaltar o caso das bibliotecas criptográficas. De acordo com Anderson (1993), a maioria das falhas de segurança, devidas ao mau uso de criptografia, são de fato erros de configuração, de implementação e de funções administrativas; como por exemplo, a gestão inadequada de chaves criptográficas, a utilização de configurações vulneráveis de criptografia forte ou o uso de criptografia reconhecidamente fraca.

Em uma pesquisa, previamente preliminar foi constatado que não há atualmente nem em Ginga-NCL e nem em Lua uma biblioteca criptográfica completa, plenamente funcional. O fato de tanto a biblioteca LuaCripto, quanto a biblioteca LuaMD5, somente oferecerem acesso a algoritmos de hash ou algoritmos de cifração fracos (DES de 56 bits) é um grande limitante da aplicabilidade destas bibliotecas. Além disto, não foi encontrado um receptor de TVDi que oferecesse às aplicações outra fonte de serviços criptográficos, diferente de LuaMD5, LuaCripto, ou LuaSec. Esta última é uma fachada para estabelecimento de conexões SSL/TLS via uma instalação do OpenSSL subjacente. LuaSec não oferece acesso via API às funções criptográficas do OpenSSL.

A interface restrita para a interação humano-computador (IHC) é outro aspecto diferenciado no teste de segurança das aplicações para os receptores de TVDi brasileiros. De modo geral, o único dispositivo de entrada de comandos pelo usuário é o controle remoto da TV, o qual é geralmente mais limitado em opções de comandos que um telefone celular comum com teclado numérico e poucas teclas extras.

Esta dificuldade relativa de interação homem-máquina afeta não somente a usabilidade dos aplicativos, mas também a exploração, a partir da interface com usuário, das vulnerabilidades presentes nas camadas de software subjacentes. Deve ser observado que a dificuldade citada aqui independe da resolução da imagem ou do tamanho da tela da TV. Trata-se neste caso do uso do receptor como plataforma de computação, daí a dificuldade de interação e entrada de dados e de comandos. Apesar destas limitações de IHC, foi possível avaliar a segurança de aplicações, realizar testes de intrusão e injetar comandos no receptor.

Figura 3: Camadas de software para receptores TVDi seguros.

O receptor de TVDi apresenta uma série de desafios para segurança da informação em sistemas embarcados. A Figura 3 ilustra, em um diagrama de blocos, as camadas de software de um receptor de TVDi ideal, no qual seriam atendidas as necessidades de segurança de aplicações e de dados manipulados por elas. A figura ilustra as seguintes características: (1) Isolamento de aplicações, proporcionado pelo middleware; (2) comunicação segura de informações em trânsito; (3) armazenamento seguro de dados e de programas; (4) biblioteca criptográfica plenamente funcional; (5) placa aceleradora criptográfica, independente do hardware restrito dedicado às

funcionalidades principais do receptor; (6) um sistema operacional robustecido, livre de vulnerabilidades comumente encontradas em computadores de uso geral.

5. Vulnerabilidades de programação insegura em Lua

Foram identificadas e documentadas diversas vulnerabilidades de programação insegura em Lua. Tais como injeção de comandos, condição de competição, corrupção de arquivos e código malicioso, script cruzado armazenado, referência insegura a tabelas, injeção de SQL e mau uso de criptografia. Estas vulnerabilidades, em outras linguagens de programação, estão amplamente documentas em catálogos conhecidos, como o OWASP Top 10 (2010) e o CWE/SANS Top 25 (2010). Não foi encontrada documentação de vulnerabilidades de programação insegura em Lua e nem boas práticas de segurança de software em Lua.

A seguir são apresentadas apenas as vulnerabilidades consideradas pelos autores como de maior risco imediato. As estratégias de mitigação foram omitidas deste texto e soluções genéricas às vulnerabilidades identificadas podem ser encontradas em Viega and McGraw (2001) e Howard and LeBlanc (2002), assim como em OWASP Top 10 (2010) e CWE/SANS Top 25 (2010). O catálogo completo das vulnerabilidades pode ser encontrado em documentação interna do CPqD, Braga e Restani (2010). A linguagen de programação Lua está amplamente documentada em Lerusalimschy (2003), Lerusalimschy, Figueiredo e Celes (2006) e Lerusalimschy (2009).

5.1. Injeção de comandos

Em Lua, a vulnerabilidade de injeção de comando é exposta quando um programa usa a rotina `loadstring()` para executar um trecho de código Lua construído a partir de material digitado pelo usuário. O código do Programa 1 é uma calculadora simples que lê uma expressão aritmética e a resolve, apresentando o resultado. O código usa a facilidade Lua de resolver expressões aritméticas e executar seqüências de caracteres como comandos da linguagem. Esta vulnerabilidade em Lua tem semelhanças com a vulnerabilidade A1 de OWASP Top 10 (2010) e com o erro de programação número 9 de CWE/SANS Top 25 (2010).

```
01    i = 0;
02    f = "i= "..io.read();
03    g = assert(loadstring(f));
04    g();
05    print(i);
```

Programa 1: Calculadora de expressões aritméticas simples.

A calculadora de expressões pode ser abusada do seguinte modo. Quando o programa solicita a entrada de dados, o usuário digita a seqüência maliciosa de caracteres "`1 ; os.execute('dir');`". Neste trecho de código, "`1;`" fecha a string da expressão aritmética. O sinal de ponto e vírgula ao final do código injetado encerra o comando e evita concatenações defeituosas. O trecho `os.execute('dir')` injeta um comando de sistema operacional. O resultado é a apresentação da listagem dos arquivos da pasta corrente antes da apresentação do resultado da expressão aritmética, "1".

5.2. Corrupção de arquivos por código malicioso

Em Lua, conforme descrito no capítulo 12 de Lerusalimschy (2003), arquivos de dados podem ser lidos como programas, de modo que as estruturas de dados são carregadas diretamente, em um mecanismo simples de persistência e de recuperação de dados persistidos. Em contrapartida, a facilidade de construção sacrifica a segurança. Dois aspectos devem ser observados como causas raiz da vulnerabilidade. Primeiro, o ambiente de execução Lua não verifica a integridade e nem a autenticidade de arquivos carregados em tempo de execução. Segundo, visto que não haverá diferença entre dados e programas, códigos maliciosos podem ser embutidos em arquivos de dados. Estes códigos maliciosos serão executados quando a estrutura de dados for interpretada. Esta vulnerabilidade em Lua tem semelhanças com a vulnerabilidade A1 de OWASP Top 10 (2010) e com o erro de programação número 20 de CWE/SANS Top 25 (2010).

Usando a estratégia de persistência de dados ilustrada no Programa 2 e no Programa 3, para explorar a vulnerabilidade, basta acrescentar o código malicioso ao arquivo de dados. No Programa 2, o comando `os.execute("Dir ..")` foi acrescentado ao final do arquivo de dados. O código malicioso é carregado pelo Programa 3 e executado antes da saída esperada do programa original. Outra possibilidade de corrupção seria a substituição completa do arquivo de dados por outro arquivo Lua contendo apenas código malicioso.

```
01   Registro { nome = "Alexandre", }
02   Registro { nome = "Alex",}
03   -- comando injetado corrompe o arquivo de dados
04   os.execute("dir ..");
```

Programa 2: Arquivo de dados Registro.lua.

```
01   regs = {}
02   function Registro (r)
03      if r.nome then regs[r.nome] = true end
04   end
05   dofile("Registro.lua")
06   for nome in pairs (regs) do print(nome) end
```

Programa 3: Carga e impressão de dados de um arquivo.

5.3. Script cruzado armazenado

Nesta vulnerabilidade, o aplicativo armazena dados em arquivo (ou outro meio de armazenamento). O atacante manipula a aplicação para que um script malicioso seja armazenado junto com os dados. Mais tarde, o dado perigoso (código malicioso) é carregado para a aplicação e incluído no conteúdo dinâmico, vitimando outro usuário. Esta vulnerabilidade em Lua tem semelhanças com a vulnerabilidade A2 de OWASP Top 10 (2010) e com o erro de programação número 1 de CWE/SANS Top 25 (2010).

O Programa 4 lê dados de entrada e, com uma estratégia simples de serialização, os salva no arquivo do Programa 6. Os colchetes duplos usados como delimitadores de string pelo Programa 4 permitem a inclusão de caracteres especiais. O Programa 5 recupera o dado armazenado, o qual pode conter o script injetado, e o apresenta ao usuário. Deste modo, scripts podem ser injetados durante a leitura de dados pelo Programa 4 e serão armazenados e refletidos na resposta ao usuário. A seqüência de

caracteres a seguir fecha uma string em aberto, injeta o script malicioso e reconstitui a atribuição de variável.

```
]] ; print("Olá") ; aaaaa = [[
```

O Programa 6 mostra o resultado. A seqüência a seguir fecha uma string em aberto, injeta o script malicioso e transforma em comentário tudo o que estiver após o trecho injetado.

```
]] ; print("olá") --
```

```
01   function serialize (o)
02      if type(o) == "string" then
03         io.write("variavel = [[", o, "]]")
04      end
05   end
06   io.output(".\\serial.lua")
07   o = io.read();
08   serialize("isto é um texto de teste "..o);
09   io.close();
```

Programa 4: Usando strings delimitadas por colchetes duplos.

```
01    dofile(".\\serial.lua");
02    print("\n")
03    print(variável)
```

Programa 5: Reconstituição da variável a partir de string armazenada.

```
01    variavel = [[isto é um texto de teste ]] ; print("Olá")
02    ; aaaaa = [[]]
```

Programa 6: variável armazenada, violada por código malicioso.

5.4. Referência insegura a tabelas

Diversas linguagens de programação procedimentais, que possuem vetores e acesso a dados por referência, sofrem de uma dificuldade bastante peculiar: o acesso múltiplo, por meio de diversas referências, a um mesmo dado. Nesta condição, o dado referenciado pode ser modificado por qualquer um dos detentores de referências, sem que os outros detentores tomem conhecimento, em tempo hábil, da modificação. Esta vulnerabilidade em Lua tem semelhanças com a vulnerabilidade A4 de OWASP Top 10 (2010) e com o erro de programação número 25 de CWE/SANS Top 25 (2010).

Em Lua, esta vulnerabilidade se manifesta no acesso a tabelas cujas referências são compartilhadas. A exploração da vulnerabilidade é construída sobre o conceito de proxies de tabelas, com metatables e metamethods de Lua. O Programa 7 contém o código de um proxy de acesso, um monitor de referências, que serve de intermediário ou mediador em todo acesso ao conteúdo de uma dada tabela original. O monitor de referências, `MR`, possui duas rotinas principais. Primeira, a função `MR.monitor(t)`, responsável pela construção da metatable, o proxy propriamente dito. Segunda, a função `MR.interno()` que retorna a tabela original.

A vulnerabilidade está na manutenção de uma referência interna, não uma cópia ou instância única, da tabela mediada (a variável `MR._t`), e em externá-la, espalhando descontroladamente as referências. Podendo levar até a uma condição de corrida.

```
01     MR = { _t = nil}
02
03    function MR.monitor(t)
04                MR._t = t
05           local proxy = {}
06           local    mt = {
07           __index    = function (t,k) return MR._t[k] end,
08           __newindex = function (t,k,v) MR._t[k] ="MR: " .. v end
09           }
10           setmetatable(proxy, mt)
11           return proxy
12    end
13
14    function MR.interno() return MR._t end
```

Programa 7: Monitor de referências a tabelas.

6. Avaliação de segurança e testes de intrusão em receptores de TVDi

Esta seção descreve experimentos de varredura de vulnerabilidades e de testes de intrusão realizados em laboratório sobre receptores de TVDi de prateleira. Foram realizadas varreduras simples, não invasivas, com ferramentas comuns utilizadas em varreduras de redes de computadores: Nessus (versão livre/gratuita), Nmap/ZenMap (livre/gratuito) e WireShark (livre/gratuito). Os alvos das varreduras foram receptores com conexão de rede (porta Ethernet) e middleware Ginga-NCL com Lua. Informações do fabricante, a marca e o modelo dos receptores testados foram omitidos deste texto.

Os resultados do experimento foram os seguintes. Foram capturados dados HTTP em claro enviados e recebidos por aplicação interativa instalada no receptor (a aplicação não possuía comunicação segura com HTTPS). Ainda, foi possível obter informações gerais dos receptores. Foram determinadas as seguintes informações: endereços IP e MAC, sistema operacional, e traceroute. Finalmente, foram identificadas vulnerabilidades exploráveis e comuns aos equipamentos conectados em rede.

Uma vez tendo sido identificado o sistema operacional do receptor, foi possível, a partir de ferramenta desenvolvida para tal, a realização de injeção de comandos de sistema e a execução de scripts, com a obtenção das seguintes informações: arquivos com permissão de escrita e leitura; quantidade de arquivos sem proprietários; permissões de todos os arquivos nos diretórios de sistema; permissões do arquivo de senhas; informações gerais do sistema (versão do sistema, quantidade de memória RAM, utilização de discos); tempo de vida do dispositivo; conexões abertas; relação de pacotes instalados no sistema operacional; quantidade de contas de usuário no sistema; as contas de usuário não bloqueadas; e os usuários ativos/logados.

A partir da injeção de comandos e análise dos resultados, foram feitas duas descobertas críticas para a segurança dos receptores. Primeira, o usuário root estava ativo em pelo menos um modelo de receptor testado. Neste mesmo modelo de receptor, as aplicações ginga em execução estavam associadas ao usuário de sistema root. Segunda, em pelo menos um modelo de receptor testado, não havia isolamento entre aplicações NCL, NCLua ou Lua. Do ponto de vista do sistema operacional, as aplicações não estavam isoladas umas das outras. Nos experimentos realizados foi possível para uma aplicação (script) de teste alterar os arquivos e até remover outras aplicações da área de aplicações do usuário. Do ponto de vista da máquina de execução Ginga-NCL, também não há isolamento entre aplicações. Nos experimentos, bastou que

a aplicação de testes soubesse o arquivo do componente a invocar para assim ter acesso à outra aplicação.

7. Considerações finais

Este texto relata um trabalho em progresso realizado pelo CPqD na avaliação de segurança da plataforma brasileira de TV digital interativa, de receptores de TVDi de prateleira, embarcados com Ginga-NCL, e aplicações interativas em NCLua e Lua.

Foram identificadas e documentadas diversas vulnerabilidades de programação insegura em Lua. Além disto, foram realizadas, em receptores de prateleira, varreduras de vulnerabilidades via interface de rede e testes caixa branca de injeção de comandos. Foram encontradas vulnerabilidades graves na configuração dos sistemas operacionais e de proteção de aplicações e seu ambiente de execução.

No futuro próximo, assim como está acontecendo com os smartphones e já aconteceu com os PCs, os receptores de TVDi se tornarão uma fonte importante de vazamento de informações privadas. Além disso, eles estarão na fronteira de proliferação dos softwares maliciosos e, junto aos computadores pessoais e aos smartphones, eles poderão ser vetores de ataques maciços às redes de computadores e de telecomunicações.

Os estudos e experimentos realizados neste projeto de pesquisa levam a um cenário preocupante em curto prazo, no qual as seguintes ameaças devem ser monitoradas com maior atenção: a contaminação de receptores por softwares maliciosos, o uso destes receptores contaminados como vetores de ataques maciços de negação de serviço e as violações de sigilo e de privacidade decorrentes de receptores comprometidos.

Muito pode ser aprendido com as estratégias de segurança adotadas pelos fabricantes de dispositivos móveis e de sistemas embarcados. As restrições de hardware e de IHC impõem desafios tanto aos atacantes quanto aos profissionais de segurança. Quando confrontadas por problemas semelhantes, as indústrias de telefones e de receptores responderam de modo análogo. Não é uma surpresa que a norma brasileira para segurança de aplicativos Ginga seja semelhante, em diversos aspectos, ao modelo de segurança das plataformas modernas de smartphones.

7.2. Trabalhos futuros

Este texto apresenta o início de um esforço contínuo no tratamento da segurança da plataforma de televisão digital interativa brasileira. Há diversas oportunidades para trabalhos futuros. Em primeiro lugar, a investigação contínua das vulnerabilidades de programação, não somente em Ginga-NCL e Lua, mais também em Ginga-J. Levando a expansão do catálogo de vulnerabilidades. Ainda, a elaboração de boas práticas de programação e desenvolvimento de software seguro para a plataforma de TVDi. Em seguida, avaliação de segurança em implementações de mercado da plataforma Ginga de TVDi, incluindo receptores, simuladores e outros pacotes de software, podendo levar a determinação de normas de segurança mais específicas. Além disso, outra área a ser explorada é a definição e especificação de requisitos de segurança para serviços ou aplicações de T-commerce, T-gov e T-banking, que oferecem no mínimo o mesmo grau de segurança dos serviços equivalentes na Internet. Finalmente, a construção de

ferramentas e de componentes de segurança específicos para a plataforma de TVDi brasileira. Abrangendo tanto ferramentas de apoio ao desenvolvimento seguro, quanto os mecanismos de segurança que atendam aos requisitos das aplicações.

Referências

ABNT NBR 15605-1 (2008). Associação Brasileira de Normas Técnicas. **Televisão digital terrestre — Tópicos de Segurança. Parte 1: Controle de cópias**. ABNT 2008. ISBN 978-85-07-01041-8

ABNT NBR 15605-2 (em preparação). Associação Brasileira de Normas Técnicas. **Televisão digital terrestre — Tópicos de Segurança. Parte 2: Mecanismos de segurança para aplicativos interativos**.

Anderson, R. (1993). **Why cryptosystems fail**. In Proceedings of the 1st ACM Conference on Computer and Communications Security (Fairfax, Virginia, United States, November 03 - 05, 1993). CCS '93. ACM, New York, NY, 215-227.

Barbosa, S. D. J. e Soares, L. F. G. (2008). **TV Digital Interativa no Brasil se Faz com Ginga – Fundamentos, Padrões, Autoria Declarativa e Usabilidade**. Livro da Jornada de Atualização em Informática (JAI), Capítulo 3. Congresso da Sociedade Brasileira de Computação. 2008.

Bickford, J., O'Hare, R., Baliga, A., Ganapathy, V., and Iftode, L. (2010) **Rootkits on smart phones: attacks, implications and opportunities**. In Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications (Annapolis, Maryland, February 22 – 23, 2010). HotMobile '10. ACM, New York, NY, 49-54. 2010.

Braga, A. M. e Restani, G.S. (2010). **Análise de segurança de receptores de TV Digital Interativa, de GINga-NCL e de Lua**. Relatório Técnico PD.30.12.34A.0013A/RT-02-AA, resultado do projeto Serviço Multiplataforma de TV Interativa – SMTVI, meta Serviço T-Commerce (M6), junho, 2010. Fundação CPqD, Campinas, SP.

Cai, L., Machiraju, S., and Chen, H. (2009) **Defending against sensor-sniffing attacks on mobile phones**. In Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications For Mobile Handhelds (Barcelona, Spain, August 17 – 17, 2009). MobiHeld '09. ACM, New York, NY, 31-36. 2009.

Carvalho, D. F., Milanez, M. G., Avelino, M. J. B., Bruschi, S. M., e Goularte, R. (2007) **SecBox: Uma abordagem para segurança de set-top boxes em TV Digital. Anais do VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**. ISBN: 978-85-7669-127-3.

CWE/SANS Top 25 (2010). **CWE/SANS Top 25 Most Dangerous Programming Errors**. Version 2.0, 2010. cwe.mitre.org/top25 e www.sans.org/top25-programming-errors.

Honorato, G. d. and Barbosa, S. D. (2010). **NCL-inspector: towards improving NCL code**. In Proceedings of the 2010 ACM Symposium on Applied Computing (Sierre, Switzerland, March 22 - 26, 2010). SAC '10. ACM, New York, NY, 1946-1947

Howard, M. and LeBlanc, D. (2002). **Writing Secure Code**, Second Edition. December 04, 2002. ISBN 9780735617223

Hypponen, M. (2007). **State of Cell Phone Malware in 2007**. USENIX. Disponível on-line na URL www.usenix.org/events/sec07/tech/hypponen.pdf

Kocher, P., Lee, R., McGraw, G., and Raghunathan, A. (2004). **Security as a new dimension in embedded system design**. In Proceedings of the 41st Annual Design Automation Conference (San Diego, CA, USA, June 07 - 11, 2004). DAC '04. ACM, New York, NY, 753-760.

Lerusalimschy, R. (2003). **Programming in Lua**, 1st. Ed. 2003. ISBN 85-903798-1-7. www.lua.org/pil.

Lerusalimschy, R.(2009). **Uma Introdução à Programação em Lua**. Livro da Jornada de Atualização em Informática (JAI), Capítulo 3. Congresso da Sociedade Brasileira de Computação. 2009.

Lerusalimschy, R., Figueiredo L. H. e Celes, W. (2006). **Lua 5.1 Reference Manual**. 2006. ISBN 85-903798-3-3. www.lua.org/manual/5.1/pt.

Oberheide, J. and Jahanian, F. (2010) **When mobile is harder than fixed (and vice versa): demystifying security challenges in mobile environments**. In Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications (Annapolis, Maryland, February 22 – 23, 2010). HotMobile '10. ACM, New York, NY, 43-48. 2010.

Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., and Jahanian, F. (2008) **Virtualized in-cloud security services for mobile devices**. In Proceedings of the First Workshop on Virtualization in Mobile Computing (Breckenridge, Colorado, June 17 – 17, 2008). MobiVirt '08. ACM, New York, NY, 31-35. 2008.

OWASP Top 10 (2010). **The Ten Most Critical Web Application Security Risks**. 2010. www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

Ravi, S., Raghunathan, A., Kocher, P., and Hattangady, S. 2004. **Security in embedded systems: Design challenges**. ACM Trans. Embed. Comput. Syst. 3, 3 (Aug. 2004), 461-491.

Traynor, P., Lin, M., Ongtang, M., Rao, V., Jaeger, T., McDaniel, P., and La Porta, T. (2009) **On cellular botnets: measuring the impact of malicious devices on a cellular network core**. In Proceedings of the 16th ACM Conference on Computer and Communications Security (Chicago, Illinois, USA, November 09 – 13, 2009). CCS '09. ACM, New York, NY, 223-234. 2009.

Viega, J. and McGraw, G. (2001). **Building Secure Software: How to Avoid Security Problems the Right Way**. Addison-Wesley Professional. October 4, 2001. ISBN 978-0201721522.