Guia do Programador Web

G-Buster Access Origin Identification

Brasília, janeiro de 2011
Versão 2.1

GAS Tecnologia
http://www.gastecnologia.com.br

São Paulo – SP
Av. Paulista 777, 15º andar
CEP: 01311-100
Telefone: (11) 3145-1950

Belo Horizonte – MG
Av. Cristiano Machado 1300, sala 603
CEP: 31110-230
Telefone: (31) 2512-0147

Brasília – DF
SCLN 311 Bloco E 2º andar
CEP: 70757-550
Telefone: (61) 3349-1188
FAX: (61) 3340-7607

Confidencialidade

Este documento possui informações proprietárias referentes às tecnologias, produtos, serviços, métodos e técnicas da empresa GAS Tecnologia. Este Guia do Programador Web apresenta informações que não devem ser divulgadas fora do âmbito da organização adquirente da solução G-Buster. Sua duplicação, cópia ou qualquer outro processo de reprodução ou divulgação estão proibidos para qualquer outro propósito que não seja o de obter informações para ajustar e adequar a aplicação e-Businness e o fluxo do processo interno para disponibilização de serviços aos clientes finais.

Controle de Versão

Versão	Data	Alterações	Autor
1.0	30-jun-09	Elaboração do documento	Marlos Chida
1.1	02-set-09	Adequação do documento	Henrique Ribeiro
1.2	19-out-09	Atualização dos itens 2.1 – Pré-requisitos de Instalação, 3.2.1 – Método Getstatus, 3.2.2 – Método Function1, 3.2.3 – Método Digest1 e 3.2.4 – Método Digest2	Henrique Ribeiro
1.3	24-nov-09	Atualização do item 2.1 – Pré-Requisitos de Instalação	Henrique Ribeiro
1.4	02-dez-09	Atualização do item 2.1 – Pré-Requisitos de Instalação e inserção de novo valor no item 3.2.1 – Método Getstatus	Henrique Ribeiro
1.5	10-dez-09	Atualização dos itens 2.2 – Detalhes Técnicos da Instalação Web e 3.2.1 – Método Getstatus	Henrique Ribeiro
1.6	07-jan-10	Inclusão dos itens 3.2.6 – Método Get Version, 3.2.7 – Método Get Native Library Version e 3.2.8 – Método Encripta	Henrique Ribeiro
1.7	21-jan-10	Atualização dos itens 3.2.1 – Método GetStatus, 3.2.3 – Método Digest1, 3.2.4 – Método Digest2, 3.2.5 – Método Digest3, 3.2.6 – Método GetVersion e 3.2.8 – Método Encripta e atualização da sintaxe e dos parâmetros do item 4.2.2 – Função GetSeed	Henrique Ribeiro
1.8	24-fev-10	Inserção do item 5 – Exemplo de Fluxo de Execução	Henrique Ribeiro
1.9	30-jul-10	Correção do nome do Sistema Operacional GNU/Linux e inserção do item 3.2.6 – Método Function10	Henrique Ribeiro
2.0	10-set-10	Atualização do item 2.1 – Pré-Requisitos de Instalação	Henrique Ribeiro
2.1	19-jan-11	Adaptação do documento para o cliente Ministério da Saúde	Henrique Ribeiro

SUMÁRIO

1. INTRODUÇÃO

1.1. DESCRIÇÃO

O G-Buster Access Origin Identification é uma solução de segurança que gera um código identificador único para cada computador, tomando-se como base uma série de elementos de hardware e software. O código identificador funciona como se fosse uma “impressão digital” que não se repete em duas máquinas distintas.

Esta identificação única, denominada Machine Identification (MID), é gerada a partir de características físicas do hardware de cada equipamento e de informações de software. Esta informação é então transmitida para a aplicação do cliente que passará a utilizá-la como segundo fator de autenticação, configurando uma espécie de DNA de cada máquina.

Seu propósito é o de identificar o computador que está promovendo um acesso a uma aplicação, ativo ou recurso de rede ou mesmo a um Serviço na Internet.

A solução pode ser utilizada com os seguintes objetivos:

a) segundo fator de autenticação ou identificação positiva;

b) assegurar que os acessos a um sistema estão sendo realizados por um computador autorizado e/ou conhecido;

c) permitir rastreabilidade de acesso;

d) mecanismo complementar de prevenção a fraudes;

e) base de informações para sistema de gestão de risco;

f) possui fatores de identificação que podem variar conforme a necessidade.

1.2. OBJETIVOS

Este documento apresenta as informações técnicas necessárias para a implementação na Aplicação Web do GBAS – G-Buster Access Origin Identification.

1.3. PÚBLICO ALVO DESTE DOCUMENTO

Este documento tem como público alvo os analistas do G-Buster Access Origin Identification e os clientes da GAS Tecnologia.

2. INSTALAÇÃO NO CLIENTE

2.1. PRÉ-REQUISITOS DE INSTALAÇÃO

- A instalação do G-Buster Access Origin Identification exige que o usuário possua os seguintes privilégios:
  - Sistemas Operacionais GNU/Linux, Windows ou Mac OS X, operando em plataforma X86 (Intel-compatível).
  - Sistemas Operacionais Windows operando em plataforma X86-64 (Intel-compatível).
  - Sua implementação permite que o componente seja utilizado com sucesso em Sistemas Operacionais (GNU/Linux e Mac OS X) de 64 bits da mesma arquitetura (X86-64), desde que o ambiente de execução (navegador e máquina virtual Java) seja de 32 bits. Caso o ambiente de execução seja de 64 bits, apenas os métodos Digest1, Digest2, GetVersion e Encripta estarão disponíveis;
  - Java Virtual Machine 1.4.2 ou superior;
  - Permissão para download e execução de Applets Java;
  - Acesso através do Java à pasta temporária do usuário.

- Os privilégios supracitados somente serão necessários para o processo correspondente ao navegador e seus processos-filhos (máquina virtual Java);
- Para efetuar a atualização do produto, não é necessária nenhuma permissão adicional;
- Os seguintes navegadores foram homologados até o momento para utilização em conjunto com a solução:
  - Internet Explorer 6, 7 e 8;
  - Firefox 2.0, 3.0 e 3.5;
  - Safari 3.0 e 4.0;
  - Opera 9.64 e 10.0;
  - Google Chrome 2.0 (Versões oficiais, versão “Chromium” não homologada).

2.2. DETALHES TÉCNICOS DA INSTALAÇÃO WEB

- A instalação será realizada diretamente a partir de uma página web. Para isso, serão utilizados os recursos de download e execução de Applets Java do navegador;
- Para instalar o Produto no cliente final, a Instituição deverá acrescentar uma tag referenciando um Applet Java no código html de uma página de seu site.
  - Um parâmetro “*Seed*” correspondente à semente de criptografia deve ser fornecido ao Applet.
  - O parâmetro Seed informará ao Applet, de forma encriptada, a URL para download da biblioteca nativa do componente para validação de integridade e verificação da necessidade de download de arquivos. Também trará a URL para envio de eventos (Function3), além de informações de *timestamp* e chaves de sessão.

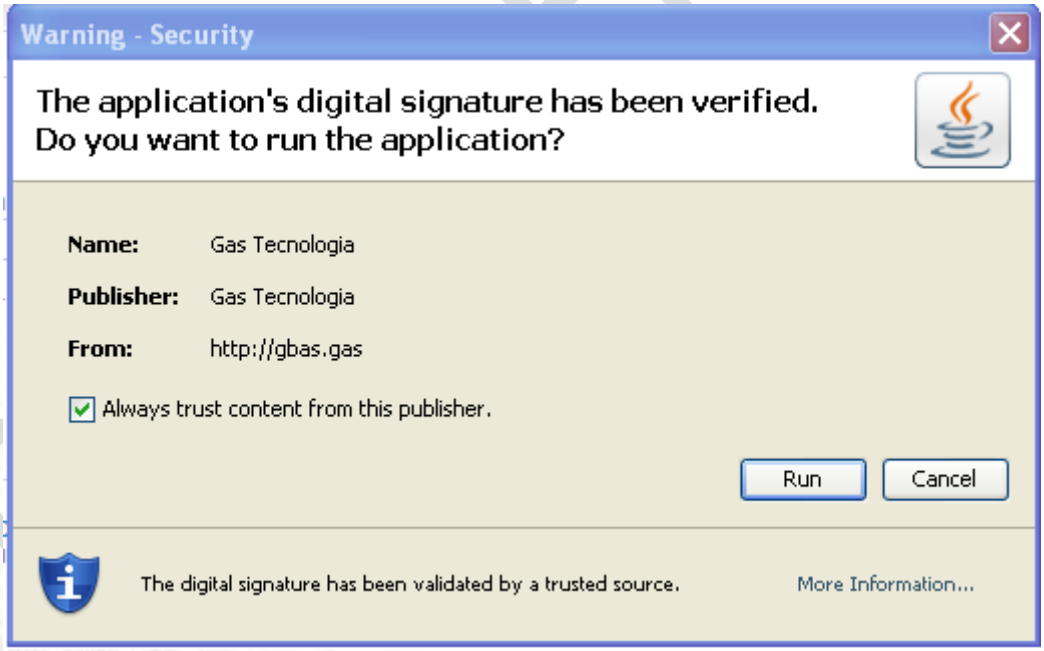
Segue abaixo exemplo de tag de instalação/execução:

```
      <Applet id="jmid" name="jmid"
archive="http://GBAS.gas/Content/GBASXXX.jar"
code="br/com/gas/mid/GBAS.class" height="1" width="1" MAYSCRIPT>
          <param
value="eNoNzcuWQzAAANAPsvBWljIxJaiWlrCj3qGEnB7y9TP3B65lO7qlXCTmzvojTVsylHF
XvQ3naYguyoyMGpcA/oKDmyQ0+a1AC1PipFxP01YqFwKwlKp6T1PZCfIU+tunnmE4JkzY0/k2A
ns8aNiKa/nemRdwQRllPf+GflGceIU1Hjpai53yWU6OIV1VRPcp0rB3NEfIvG+sSIRBYeHf1ZW
n1/AT7kPUDZyeE2I+7FVRN5Eo55IH6Fm1TTteqZ4H1w2o9hQ0U0I39pak6ew5IsUNjVgNSxkF8
ORN0G5fomkdTjGa/bFKGBovc2Z2XiNFPdWSwzRVSB6E/X92WRZxpWumdU+fKicMUKwcAs8E3Fs
rwI7dDUoMZiq+PvK8bfG9JnP7eIVR5O6yA/8AqRt5Lg==" name="seed" />
           <param name="statusChangeCallback"
value="myJavaScriptCallbackFunction" />
      </Applet>
```

Onde “GBASXXX.jar” é o nome do Applet utilizado pela Instituição.

O parâmetro “statusChangeCallback” é opcional. Caso esteja presente deverá ter como valor o nome de uma função JavaScript na mesma página que será chamada cada vez que o status do download da biblioteca nativa for atualizado. Para maiores informações consulte a documentação do método getStatus.

- O navegador interpretará a referência acima e determinará se é necessário baixar o arquivo “.jar”. Uma vez baixado, o navegador apresentará uma tela de confirmação para que o usuário autorize a execução. Veja o exemplo a seguir:



- O pacote JAR deverá estar assinado com o certificado digital da Instituição adquirente da solução ou de outra empresa que ela indique;
- Uma vez executado, o Applet desencriptará as informações existentes no parâmetro Seed, de forma a obter a URL para download da biblioteca nativa da solução. Após o download da biblioteca nativa, que é realizado de forma assíncrona, o componente estará completo e poderá responder à chamada de todos os seus métodos.

3. INTERFACE DE SCRIPT NO CLIENTE

3.1. UTILIZAÇÃO DO GBAS

Uma vez instalado e instanciado, o Applet Java pode ser utilizado para obter informações sobre a estação no qual está sendo executado, através da chamada aos seus métodos.

3.2. MÉTODOS DO APPLET

3.2.1. MÉTODO GETSTATUS

Esta função indica o status do download da biblioteca nativa.

Sintaxe:

int getStatus()

Exemplo (JavaScript). O código abaixo só é executado quando o Applet Gbas estiver pronto:

```
  function myJavaScriptCallbackFunction ()
  {
     var status = document.jmid.getStatus();

     if (status == 1)
     {
        alert("Download Concluido");
     }
     else if(status == 3)
     {
        alert("Falha na execução da biblioteca nativa ou ambiente não
suportado);
      }
     else if (status == 0)
     {
        setTimeout(CheckApplet(), 1000);
     }
      else if (status == 4) {
        alert("Execução do Applet foi cancelada");
      }
     else {
        alert("Falha no Download");
     }

  }
```

Valor de retorno:

Retorna um inteiro, que indica uma das condições abaixo:

Valor	Descrição
0	Download em andamento
1	Download concluído com êxito
2	Falha no Download
3	Download concluído, porém a execução da biblioteca nativa falhou. Estações sem suporte nativo
4	Cancelamento da execução do Applet GBAS

3.2.2. MÉTODO DIGEST1

Esta função retorna uma string referente ao Digest da máquina. O retorno somente ocorrerá caso seja passada uma senha para criptografia, e a string criptografada retornada deve ser desencriptada utilizando o framework GbCrypt da GAS Tecnologia. O retorno é um identificador único na forma de uma string hexadecimal de 8 caracteres encriptada com a chave fornecida. Este método pode ser instanciado em estações sem suporte nativo.

Sintaxe:

Digest1(DownloadTimeout, Password)

Parâmetros:

DownloadTimeout:

Número inteiro indicando por quantos milissegundos se deve esperar o término do download do módulo cliente se o mesmo não estiver previamente baixado;

Password:

Senha para encriptar a string de retorno.

Exemplo (JavaScript):

```
function digest1()
{
   var dig1 = "";

   dig1 = document.jmid.Digest1(15000, "senha");
   alert("Digest -> " + dig1);
}
```

3.2.3. MÉTODO DIGEST2

Esta função retorna uma string referente ao Digest2 da máquina. O retorno somente ocorrerá caso seja passada uma senha para criptografia, e a string criptografada retornada deve ser desencriptada utilizando o framework GbCrypt da GAS Tecnologia. O retorno é um identificador único na forma de uma string hexadecimal de 8 caracteres encriptada com a chave fornecida. É semelhante ao método Digest, porém utiliza uma forma alternativa de identificar o computador do usuário. Este método pode ser instanciado em estações sem suporte nativo.

Sintaxe:

Digest2(DownloadTimeout, Password)

Parâmetros:

DownloadTimeout:

Número inteiro indicando por quantos milissegundos se deve esperar o termino do download do módulo cliente se o mesmo não estiver previamente baixado;

Password:

Senha para encriptar a string de retorno.

Exemplo (JavaScript):

```
function digest2()
{
   var dig2 = "";

   dig2 = document.jmid.Digest2(15000, "senha");
   alert("Digest2 -> " + dig2);
}
```

3.2.4. MÉTODO FUNCTION3

Este método é utilizado para indicar os acessos efetuados pelos usuários à aplicação Web da Instituição. Ao chamar o Function3, um evento encriptado é enviado pra a URL fornecida no parâmetro Seed do Applet, contendo as informações passadas como parâmetro para a Function3 e as informações de identificação do computador. Este método, em sua utilização para registro de logon do usuário, deve ser executado uma única vez por sessão.

Sintaxe:

Function3 (DownloadTimeout, Agencia, Conta, Usuario, TipoLogin)

Exemplo:

```
var midValue = document.jmid.Function3(15000,"1234","5678",
"JohnDoe","PF");
```

Valor de retorno:

Retorna um inteiro, que indica uma das condições abaixo:

Valor	Descrição
1	Evento gerado e enviado com êxito
2	Falha na geração ou envio do evento

3.2.5. MÉTODO FUNCTION10

Este é um método alternativo ao método Function3 que pode ser utilizado para indicar os acessos efetuados pelos usuários à aplicação Web da Instituição. Ao chamar o método Function10, uma string encriptada é retornada contendo as informações passadas como parâmetro (mesma informação enviada pelo método Function3), além das informações de identificação do computador (similar aos métodos Digest1 e Digest2). Este método gera informações utilizadas internamente pelo G-Buster Access Origin Identification no computador do usuário e deve ser executado uma única vez por sessão (similar aos método Function3). A string encriptada correspondente às informações de login deve ser enviada pela aplicação Web da instituição como um evento para o servidor BDU para processamento e armazenamento no banco de dados da solução.

Sintaxe:

Function10 (DownloadTimeout, Agencia, Conta, Usuario, TipoLogin, Password)

Exemplo:

```
var midValue = document.jmid.Function10(15000,"1234","5678",
"JohnDoe","PF",”senha”);
```

3.2.6. MÉTODO GET VERSION

Este método é utilizado para obter a versão atual do Applet GBAS. Ao chamar o método getVersion, uma *string* contendo a versão é retornada. Este método pode ser instanciado em estações sem suporte nativo.

Sintaxe:

getVersion ()

Exemplo:

```
var midValue = document.jmid.getVersion();
```

Valor de retorno:

Retorna uma *string* com a versão atual do Applet GBAS.

3.2.7. MÉTODO GET NATIVE LIBRARY VERSION

Este método é utilizado para obter a versão atual da biblioteca nativa do GBAS. Ao chamar o método getNativeLibraryVersion, uma *string* contendo a versão é retornada.

Sintaxe:

getNativeLibraryVersion (DownloadTimeout)

Exemplo:

```
var midValue = document.jmid.getNativeLibraryVersion(15000);
```

Valor de retorno:

Retorna uma *string* com a versão atual da biblioteca nativa GBAS.

3.2.8. MÉTODO ENCRIPTA

Este método é utilizado para encriptar uma String. Recebe como parâmetros a String que será encriptada (StrToCrypt) e a chave de encriptação (Senha). O tamanho máximo permitido para chave de encriptação é de 16 caracteres. Este método pode ser instanciado em estações sem suporte nativo.

Sintaxe:

getNativeLibraryVersion (StrToCrypt, Senha)

Exemplo:

```
var midValue = document.jmid.Encripta("Texto a ser Encriptado","123password");
```

Valor de retorno:

Retorna uma *string* encriptada de acordo com o texto original e senha informados como parâmetro.