株式会社 ITS MORE
2020年4月設立

Digitally signed by 佐藤 豊
Date: 2020.09.03 01:26:28 +09'00'

2020年9月2日 投稿者: SATOXITS

GShell 0.2.9 – 電子署名付きHTML

社長：今日はこの件に着手しましょう。

開発：ようやく (^-^)/

基盤：夢のお告げでしょうか？

社長：何度か、クライアントからサーバにつないで何かの機能を設定したり制御するみたいな夢を見るのですが、夢の中ではよしこれでOKって納得しているのに、目が覚めて考えると何の事かわからないんです。

HTMLへの電子署名の方式

開発：実現方法の候補はざっくりこういうところかと思います。

1. 署名の方法
    1. gsh.html にブラウザからJavaScriptで署名する
    2. gsh.go を実行して署名する
    3. 外部ツールで署名する（ OpenSSL | PGP ）
2. 検証の方法
    1. gsh.htmlをブラウザからJavaScriptで検証する
    2. gsh.go を実行して検証する
    3. 外部ツールで検証する（ OpenSSL | PGP ）

3. 署名情報の付加形式
    1. 独自形式（ HTMLタグ | JavaScriptデータ | QRコードPNG | Goデータ ）
    2. S/MIME
    3. PGP
4. 検証情報の内容
    1. 公開鍵署名
    2. ダイジェスト
    3. タイムスタンプ

    社長：検討しましょう。

    開発：まず、この方式は gsh.go, gsh.html に限らず、一般の HTML に適用できるのが良いと思います。できれば、Go が無くても使えるようにしたい。そういう意味で、1.1、2.1 が最優先と思います。3 に関しても、既存の表現形式に合わせるための手間がかかるので 3.1 にしたい。まず原理的なフィージビリティを調べてから、標準の形式にはめ込めるか確認したい。

    基盤：PNGファイル自体にも署名して内蔵させたいということですが、あれとかそもそも既存の標準形式というのは無いんじゃないですか。

    開発：HTMLのタグでやる場合、署名情報を通常は display:none かなんかで非表示にして必要に応じて表示する、各入れ子の部品に対して署名できるとかもできるのが良いと思います。

    社長：どこに入れますかね。

    開発：候補がありすぎて何なんですが、innerHTMLにする、タグの属性にする、background-imageのdata URIにする、特定の形式のclass名にする、何でもアリかともいます。

    基盤：透かしみたいのがカッコいいですね。

対応できるブラウザ

社長：JavaScriptでDOMから自身のHTMLのテキストをダンプするのって、確実にオリジナルを再現できるんでしたっけ？ブラウザ依存性とかは？

開発：gsh.go.html は現在 6071行あります。ブラウザでこれを開いて、Raw Source > Whole file で textarea に表示したテキストをコピーしてファイルに書き込み、オリジナルファイルと比較。通常 Firefox でやっていますが、完全に再現・一致します。もっとも、open を open="" にしちゃう妙な癖に引っかからないようにしています。これって、Firefox のバグじゃないかと思うんですが…

開発：他でもやってみます。Chrome … 完全一致。

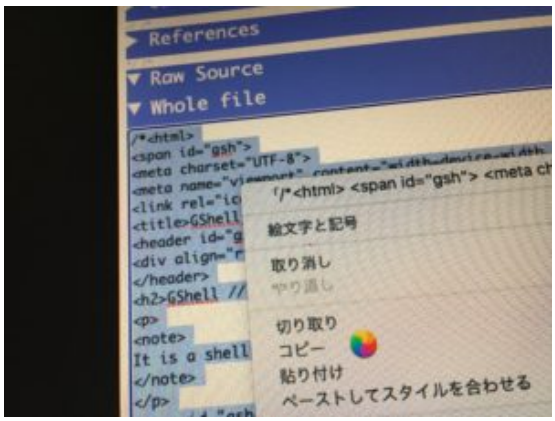
基盤：その、ファイルへのコピペをターミナルにcatするってくっそ遅いですね。15秒くらいかかる。

開発：これが一番わかり易い方法なんで。クリップボードをファイルに吐くってどうやるんですかね…　ともかく、そういう環境依存な方法ではなくて、GShell自体をHTTPサーバにして、http://xxx/yyy/gsh.goを要求されたらブラウザがダウンロード・保存できる形式で返そうと思っているのです。それはそうと、Opera では…

開発：Operaでも完全一致。Edgeでは…完全一致。問題児のSafariは…完全一致。

基盤：Chromium系4ブラウザはまあ、共通なんでしょうね。独立系の Firefox と Safari もOKと。あれ？ Vivaldi は？

開発：Vivaldiはタブの過積載で圧死寸前でして、社長がブログ書くの以外に使う気になれずw。でもやってみます。たぶん大丈夫でしょう…

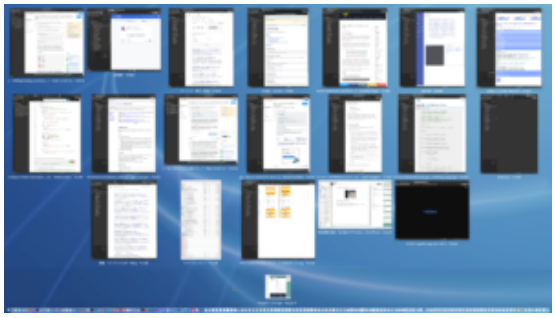


基盤：案の定、コピーで固まりましたね。

社長：案の定たけ坊ってありましたよね。小学校の国語の教科書？

基盤：ググっても出てこないですね。

開発：やはりVivaldi はブックマークが良いので、調べ物とかは Vivaldi でやってずーっと開きっぱなしなんですよ。だから溜まりまくってしまって。



基盤：タスクマネージャで見ると、だいぶサブフレームをぶら下げてるタブがありますね。

開発：一発でサブフレームプロセスを全部殺すっていう機能があると良いのですが。ぷち。ぷち。ぷち。…　少しは軽くなったような？

社長：ウィンドウとかタブごとに動作とか更新を止めるっていう機能があったように思いますが。Vivaldiだったか忘れましたが。

開発：なにはともあれ、コピペ成功。完全一致です。

基盤：全ブラウザ問題無しですね。

社長：前提になる基盤技術に問題なしということですね。

開発：あとはDOMダンプした文字列をダイジェスト関数に食わせるだけでintegrityは検証できます。class=digest みたいなタブにして追加する。このクラスのタブはダイジェストの対象から外す。まあ id 付けて DOM から消しちゃうか、検証対象の span の外に置く。ダイジェストに署名して同様に保存する。それを取り出して検証できれば完成。

基盤：span id=xxx へのダイジェストが class=digest id=xxx-digest とかだとわかりやすいですね。

開発：あとは、検証作業を実行する部分は、検証部分のソースコードを見て信頼してもらうか、そこだけは別経路でアルゴリズムを取ってインストールしといてもらうかということでしょうね。

社長：いい感じです。お昼休みにしましょう。今日は久しぶりにちかくのそば屋かな。

外は雨

社長：帰りました。外は雨です。

基盤：台風の影響ですかね。

社長：考えたら、今月はほとんど雨に出会ったことが無いです。

基盤：野菜が異常に高いのはそのせいですかね。

社長：それで、ゥエルシァでノリと煎りごまを買ってきました。ところてんのトッピングにいいかなって。

基盤：刻み海苔 vs 青ノリ対決ですね。

開発：おやつの楽しみにしましょう。

メールでのgshコード送信

社長：それで、食事しながら考えたんですが、PDF署名だ、S/MIMEだって考えてると、やりたいことの前の敷居が高すぎるなと。WordPress のプラグインだ、ブラウザの　extension だってのも回りくどい。本当にやりたいことはめっちゃシンプルなわけですから。だからもう、HTML署名でいいかなって。

開発：そもそも、部品、エレメントごとに署名する、それをさらにまとめて署名する、かつそのまま表示可能、可視・不可視選択思いのママていう使い方を考えると、現状、HTMLしか選択肢が無いように思います。

社長：それで思ったんですが、メールにこの、署名付きのHTMLは貼り付けられるだろうか？って。いやもちろん、添付ファイルとしてはできるでしょうけど。本文として。

基盤：macOS の Mail ですが…　HTMLで書いて送るツールは売られてますね。Mail 自体がどうサポートしているかはちょっとわかりません。

社長：Mailのインラインで動かせるとうれしいな。

開発：原理的にはまず送れないわけがないですよね。ただの封筒なんで。

基盤：規格外とか。

開発：定形だとは思いますよ。角型4号みたいな。

基盤：まあ、gsh.html とか gsh.go は問題なく添付できます。バイナリの gsh とか gsh.zip はGmailの検閲にかかる模様。gsh.go の場合には、go を app 形式にして準備しておく必要はありそうです。

社長：でもそれ簡単ですよね。Automatorの時にやってみましたけど。

開発：ただ、gsh.html で添付して、ブラウザで開く。ブラウザ用にダウロードボタン

を用意してダウンロードして、実行してもらうというのが、一番簡単だとは思いますが。

社長：うーん。.ghtml とかいう拡張子にしてですね、gsh.app で実行する、てな感じがうれしいですね。

基盤：gsh.img とかで送るとか。

社長：毎回それやるのは嫌ですね。一回 gsh.app をインストールしたら、あとはそれに gsh.ghtml を食わせればOKというのがうれいしい。.ghtml は gsh.app で実行するってデフォルトで設定して。

ユーザデータとしての署名付き情報コンテナ

開発：メールの話は一旦置いて、話を巻き戻しますと、要するにHTMLに電子署名を内包させられれば、それは普通ユーザレベルのデータとして転送したり保存したり表示したりできる。特に gsh.html みたいなオールインワンの形式では。

社長：昔で言えばPGPですね。ただあれは、見た目に異物感が否めなかった。

開発：表示を自分で制御する機能は想定されてないですからね。

社長：ある意味無敵のべたテキスト前提。まあそういう時代でした。PEM形式にしたって。

開発：その点、HTMLだと、自身で表示されかたを定義できるし、CSSも JavaScript も内包できるから、リッチだしダイナミック。かといってオールインワンなら外部サイトにもアクセスしないから安全。

基盤：なんだか話がうますぎると言うか簡単過ぎる感じはしますが。なぜこれまでそうなっていなかったのか。

開発：まあ、なんか大きな考え落としをしてるかも知れないですね。よくあることです。

社長：別に頓挫しても、わが社的には痛くも痒くもないです。それに、失敗はかならず貴重な資産になるものです。

とりあえずチェックサム

社長：どうもこの、飲んで食べるとしばらく仕事する気になれませんね。

開発：それが若い頃との大きな違いかなと思います。

社長：腰が重くなるんですよね。でも、一旦腰を上げればあとは結構続く。

開発：どっこいしょっと。では、とりあえずチェックサムを付けてみます。とびきり処理が遅いけど一番使いやすい社長自作のCRC32で…

社長：こんなところでお役に立てようとは。うぅぅ…

開発：あ、これは Go プログラムだからダメですねw。HTML からやるなら、JavaScript で無いと。

社長：40行程度のプログラムですから、簡単にJavaScript にもなると思いますが。

開発：なるほど。では型宣言とキャストを全部はずしてと…　おっとイキナリコンパイル通りました。HTMLから配線して。動くかな…

基盤：残念。

開発：デベロッパコンソールで見る…　あれ？引数が足りない。追加。

＊　＊　＊

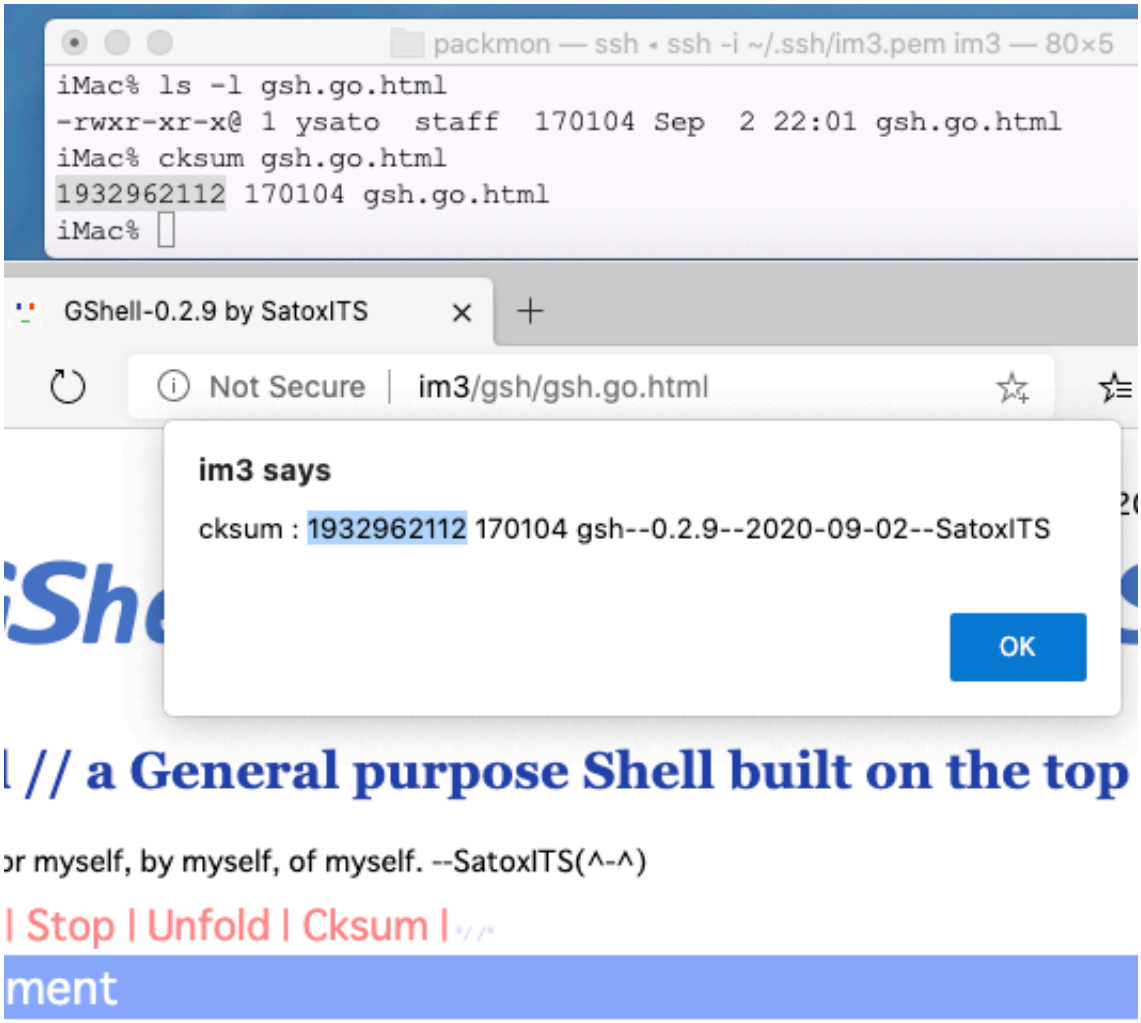
社長：で、どうなりましたかね。

開発：いやあ、今まで JavaScript の文字列とか、＋で結合するくらいしか処理したことが無かったので、その辺でつまづいてしまいました。Moz://a MDN でゼロから勉強です。

開発：ブレークスルーは、文字列の中の指定した位置のバイトの値を取り出すのが、charAt() だと思ってスタックしてたところから、ようやく charCodeAt() が正しいと気がついて脱出した時でしたw。あとは、CRCを作るビット演算をするのに整数が必要なのですが、なぜか32ビット以下には整数配列しか無い。不思議な言語です。

開発：CRCが済んだ後の課題は、自分のソースをダンプするところですが、これはHTMLエンティティがエンコードされているのを戻さないといけない。いや、エンコードされた状態でサインするというのも、別の用途でアリかなとは思います。でも、textarea 要素を作って innerHTMLに html で書いた後に、その .value プロパティでデコードされたテキストを読み出すというシャレたテクニックを Stackoverflow で引っ掛けたので、それでサクッと終わりました。

開発：で完成です。自分で自分自身の CRC32 を作れています。



基盤：おー。パチパチパチ。

社長：めでたし。

ブロッコリー

社長：ああそれで、一昨日スーパーに行った時にふと、ブロッコリーが目に入ったのです。で、これもレンジでチンしたら旨いかなと。

開発：やってみましょう。これも90度で6分くらいですかね…　ぽちっと。

レンジ：ゴー…

開発：いやはや今日は、JavaScript の整数に疲れてしまいましたが、ソーステキストのダイジェストを取るとこは通過しました。普通のハッシュアルゴリズムは Go にパッケージが完備してますから、まあ繋ぐだけですね。

レンジ：パペポ、…

開発：どうかな？ガブっ。もうちょっとですかね。でも加熱しすぎて味が壊れるのもいやだし。

基盤：マヨネーズを付けていただきましょう。もぐもぐ。うん、うまい。

社長：もぐもぐ。ブロッコリーとカリフラワーには恐ろしくマヨネーズが合うんですよね。なぜなのか不思議です。

基盤：レタス系もですが、もぐもぐ、葉や花の部分以上にこの太い芯がうまいですよね。ごちそうさまでした。

WordPress拡張HTMLの障害回避

社長：きょうは早めに和やかに終了しようと思ったのですが、WordPressに gsh.html を貼り付けたら JavaScript が動かないようになってました。昨日の版までは動いてます。

開発：今日の追加分の関係ですよね。デベロッパモードで追跡。ああ、JavaScript で構文エラーだと。まさに今日追加したCRC32の部分ですね。あー、& が & にされちゃってますね。<script> の中までそれをやるかねえ？

```
// 0.2.9 2020-0902 created chekcsum of HTML
CRC32UNIX = 0x04C11DB7 // Unix cksum
function byteCRC32add(bigcrc, octstr, octlen) {
    var crc = new Uint32Array(1)
    crc[0] = bigcrc

    let oi = 0
    for (; oi < octlen; oi++) {
        var oct = new Uint8Array(1)
        oct[0] = octstr[oi]
        for (bi = 0; bi < 8; bi++) {
            //console.log("--CRC32 "+crc[0]+" "+oct[0].toString(16)+" ["+oi+"."+bi+"]\n")
            //ovf1 = (crc[0] &#038; 0x80000000) != 0
            //ovf2 = (oct[0] &#038; 0x80) != 0
            //ovf = (ovf1 &#038;&#038; !ovf2) || (!ovf1 &#038;&#038; ovf2)
            oct[0] <<= 1
            crc[0] <<= 1
            if (ovf) {
                crc[0] ^= CRC32UNIX
            }
        }
    }
    //console.log("--CRC32 byteAdd return crc="+crc[0]+","+oi+"/"+octlen+"\n")
    return crc[0];
}
```

基盤：「拡張HTML」ブロックで見ると & に見えますから確信的ですね。

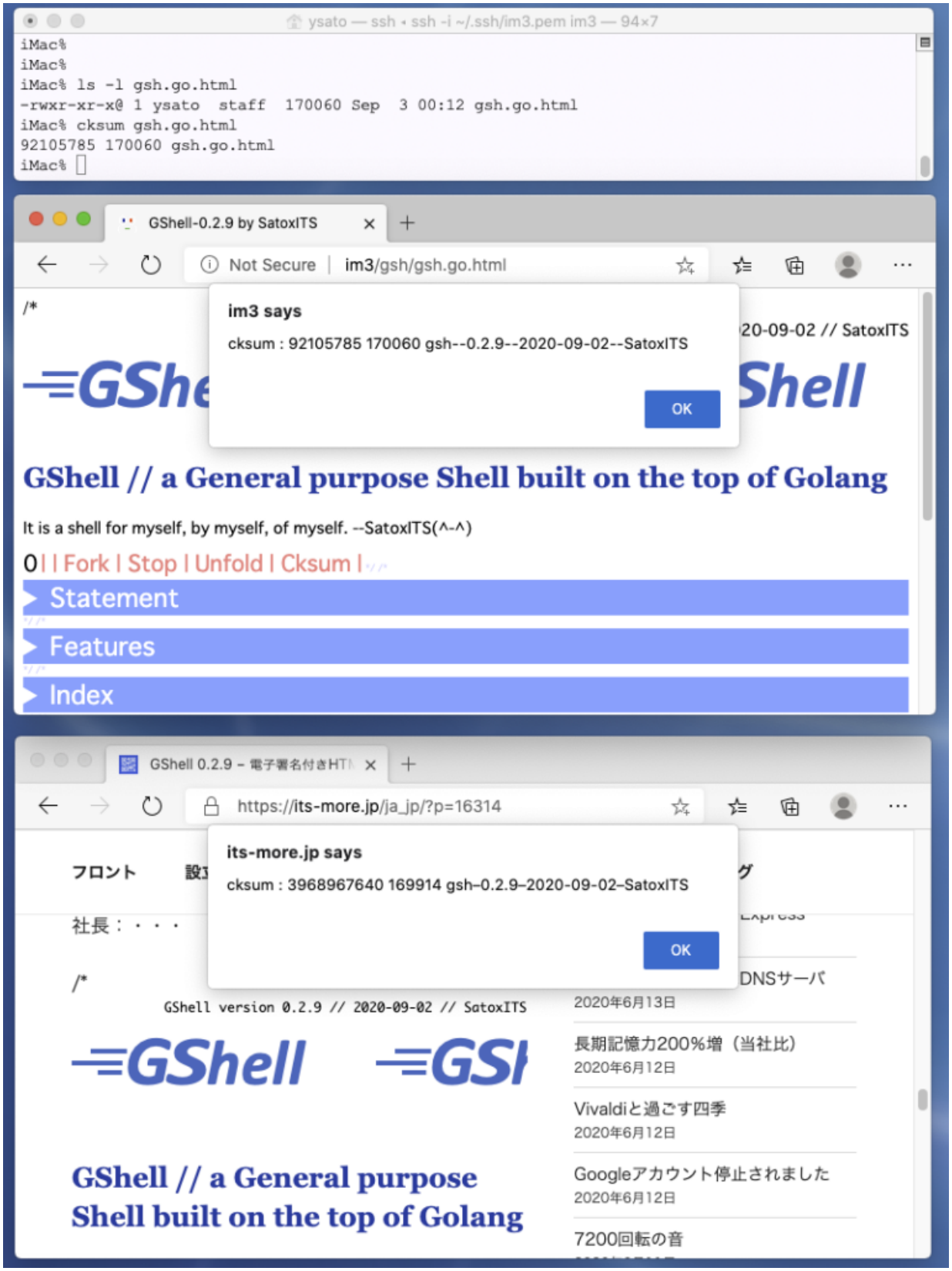
社長：というかこれ、普通に signed int32 と int8 にして、正負判定でいいんじゃないでしょうかね。

開発：社長のオリジナルコードですが。

社長：・・・

開発：& 演算子を使わないように変更しました。

社長：WordPressにアップロード。ああ動きました。



開発：WordPressの中で動くGShellはひょうきんで良いですね。

基盤：あれ？でもチェックサムが違う、というかサイズがそもそも違いますね。166バイト小さくなってます。

開発：あれま。

社長：チェックサムの効用が早速出ましたねｗ　明日対処しましょう。

基盤：checksum の情報として、行数と語数も出てると人間にはうれしいですね。

開発：そうしまふ。

— 2020-0902 SatoxITS

http-im3-gsh-gsh-0.2.9.go_ ダウンロード

```
/*
```

GShell version 0.2.9 // 2020-09-02 // SatoxITS

GShell // a General purpose Shell built on the top of Golang

```
It is a shell for myself, by myself, of myself. -SatoxITS(^-^)

0 | | Fork | Stop | Unfold | Cksum | */ /*
▶ Statement
*/ /*
▶ Features
*/ /*
▶ Index
*/ //
▶ Go Source
//
▶ Considerations
```

```
// /*
```

References

```
*/ /*
```

Raw Source

```
*/ /*
```

```
--> *///
```