

Forum:[Hack Forums](#)

Thread: [\[Preview/Opinion Needed\] Shebang's Byte Difference Engine \(SBDE\)](#)

shebang - January 2, 2014, 9:24 am

I was really wondering if it would be worth finishing this project for people to use. I was given the idea when updating a game I play, and the file size was massive. I figured there had to be a better way.

I'm absolutely positive this has been done before, but I've never seen it included in a HackForums program. Most of them use updaters that require the user to update their entire application through Net Seal or an updater they've coded themselves. The instruction file this generates would make the updating extremely easy, and require less bandwidth (especially for small updates).

Obviously this isn't beneficial when there are massive differences in files, as they will probably end up being similar in size, but small updates to large applications could benefit from a byte difference engine to patch what's needed in a few bytes. Here's a GIF example where I modified 4 bytes of MonsterSocial (realistically this would be more):

[img]<http://i.minus.com/iIP4EASJabThD.gif>[/img]

Is this something worth continuing? I'm not sure if I should even bother finishing the instruction interpreter, or making a library that could easily be implemented by the program creator.

Ace of Spades - January 2, 2014, 9:28 am

(Jan 2, 2014 10:24 AM)shebang Wrote: [\[CITE\]](#) I was really wondering if it would be worth finishing this project for people to use. I was given the idea when updating a game I play, and the file size was massive. I figured there had to be a better way.

I'm absolutely positive this has been done before, but I've never seen it included in a HackForums program. Most of them use updaters that require the user to update their entire application through Net Seal or an updater they've coded themselves. The instruction file this generates would make the updating extremely easy, and require less bandwidth (especially for small updates).

Obviously this isn't beneficial when there are massive differences in files, as they will probably end up being similar in size, but small updates to large applications could benefit from a byte difference engine to patch what's needed in a few bytes. Here's a GIF example where I modified 4 bytes of MonsterSocial (realistically this would be more):

[img]<http://i.minus.com/iIP4EASJabThD.gif>[/img]

Is this something worth continuing? I'm not sure if I should even bother finishing the instruction interpreter, or making a library that could easily be implemented by the program creator.

I vote finish it, this is very nice looking and it's not a Theme/Crypter/Rat/SMS Bomber. So keep going and post updates!

Golden Kappa - January 2, 2014, 9:36 am

Yeah I agree with Frozen, you should definitely finish that, it looks very good.

shebang - January 2, 2014, 9:40 am

Just ended up finding a little bit of a fun issue. Accidentally removed 8 bytes from the beginning of a file, not realizing it would offset everything else. Left me with a 240MB text file :P

frontenhack - January 2, 2014, 9:48 am

Looks promising, keep it up.
My vote is for continuing :D

Temploit - January 2, 2014, 9:58 am

Definitely continue! Actually looking forward to this, in contrary of most thing released here.

SubGenius - January 2, 2014, 10:12 am

Well it's not something new, but it's awesome.
Client patchers for Metin2 using this method were being sold very expensive.

So yes. Continue coding this.

Ajumma - January 2, 2014, 11:18 am

Your GIF died so I can't see it anymore, but I know what this is so I would say finish it ^_^
I would love a library or something where I could implement this into my own programs.

shebang - January 2, 2014, 12:01 pm

(Jan 2, 2014 12:18 PM)iStar Wrote: [\[CITE\]](#) Your GIF died so I can't see it anymore, but I know what this is so I would say finish it ^_^
I would love a library or something where I could implement this into my own programs.
Weird. The image is hosted on minus, so you shouldn't have an issue.

Sutton2k9 - January 2, 2014, 3:02 pm

This look very nice.
You should finish it

KingCobraXD - January 2, 2014, 3:09 pm

I took a quick look and it seems that you only change the bytes that are different. That's a good start but something you might want to think about is byte insertion/removal. For example, we have two files, one with a size of 100 bytes and another with a size of 101 bytes. The only difference between the two files is that there is a random byte inserted somewhere in the second file, thus making it one byte larger than the first file. Using the method you use right now will mean that every byte after the inserted byte will require a change. If you can detect removed/inserted bytes, you can definitely make it much more efficient.

Meerkat.? - January 2, 2014, 3:12 pm

(Jan 2, 2014 04:09 PM)KingCobraXD Wrote: [\[CITE\]](#) I took a quick look and it seems that you only change the bytes that are different. That's a good start but something you might want to think about is byte insertion/removal. For example, we have two files, one with a size of 100 bytes and another with a size of 101 bytes. The only difference between the two files is that there is a random byte inserted somewhere in the second file, thus making it one byte larger than the first file. Using the method you use right now will mean that every byte after the inserted byte will require a change. If you can detect removed/inserted bytes, you can definitely make it much more efficient.

This!

Also, storing the bytes as a text file is maybe not that clever, especially not since you use more text than the actual bytes itself.

The idea is definitely nice and you should continue, but there's a lot of room for improvement.

Assault - January 2, 2014, 3:15 pm

Tip. Instead of writing a set of string as your instruction, work on serializing something else
Write a class that represent a change & an enum that represents the instruction that you need to modify.

So this way you can have something like this

Code enum Instruction

```
{  
    ChangeByte,  
    AddByte,  
    RemByte,  
    etc  
}
```

class Change

```
{  
    public Instruction Instruction;  
    public object Value;  
}
```

This way you can store everything in a List<Change> rather then storing it all in a string.

shebang - January 2, 2014, 3:20 pm

(Jan 2, 2014 04:09 PM)KingCobraXD Wrote: [\[CITE\]](#) I took a quick look and it seems that you only change the bytes that are different. That's a good start but something you might want to think about is byte insertion/removal. For example, we have two files, one with a size of 100 bytes and another with a size of 101 bytes. The only difference between the two files is that there is a random byte inserted somewhere in the second file, thus making it one byte larger than the first file. Using the method you use right now will mean that every byte after the inserted byte will require a change. If you can detect removed/inserted bytes, you can definitely make it much more efficient.

So it should analyse the files entirely and look for patterns around the different bytes? I'm trying to think of how to do this..

KingCobraXD - January 2, 2014, 3:23 pm

(Jan 2, 2014 04:20 PM)shebang Wrote: [\[CITE\]](#) So it should analyse the files entirely and look for patterns around the different bytes? I'm trying to think of how to do this..

Yup. It's not gonna be easy. The first thing that comes to my mind is to find an algorithm that would get blocks of bytes that exists in both files.

shebang - January 2, 2014, 3:26 pm

(Jan 2, 2014 04:23 PM)KingCobraXD Wrote: [\[CITE\]](#) Yup. It's not gonna be easy. The first thing that comes to my mind is to find an algorithm that would get blocks of bytes that exists in both files.

So as oppose to just noting when a difference is found, check the bytes around it for shifting?

KingCobraXD - January 2, 2014, 3:29 pm

(Jan 2, 2014 04:26 PM)shebang Wrote: [\[CITE\]](#) So as oppose to just noting when a difference is found, check the bytes around it for shifting?

Yes. You want to check for shifting before you note the difference. Lol.

shebang - January 2, 2014, 3:30 pm

(Jan 2, 2014 04:29 PM)KingCobraXD Wrote: [\[CITE\]](#) Yes. You want to check for shifting before you note the difference. Lol.

Alright. It's starting to look like I've bit off more than I can chew here ? I'll see what I can get whipped up.

Edit: I will once my power comes back on ???

Nextrum - January 2, 2014, 3:35 pm

Looks likes a pretty good program you have here. I personally think you should continue with it.

CyberTechnologyInc - January 3, 2014, 12:31 pm

I think you should when done release a dll :D

nuk - January 3, 2014, 4:18 pm

Frozen has the right attitude for this :) Finish this bad boy up and let's see what happens :) Great work & original too :)

Anti - January 3, 2014, 4:53 pm

I'd say finish this, because its the first I have seen on HF and it is not a remote admin. tool nor a crypter. Nice job, OP. ?

shebang - January 10, 2014, 3:43 am

Progress Update #1

Figured while I'm on a long bus ride to get to some slopes, I figured I would update how far I've gotten. I've done a lot of code cleaning thanks to the guidance of some other members and some people on StackOverflow so all the current functions look nice. I've also successfully been able to detect when and where a byte has been inserted, but it's.. mediocre at best. It will need an overhaul and rigorous testing before I go further. In terms of the instruction files outputted, I won't be changing anything until I have the core functions (instruction read/write) working.