Possible way to make a very profitable 50 plus ish attack for pools?

Grinder

September 12, 2011, 07:18:36 PM

I have searched the forum and Google, but I not find a discussion of this particular variation.

Say one pool starts paying 150% reward to attract more than 50% of the miners. At the next difficulty change block it mines until it solves it, and sets difficulty to 25%. Then it can mine on this chain and solve 4x as many blocks as everybody else, and still have the most difficult block chain. Is there any way to stop this, other than hoping that miners don’t join?

Stephen Gornick

September 12, 2011, 08:26:42 PM

Quote from: Grinder on September 12, 2011, 07:18:36 PM

until it solves it, and sets difficulty to 25%

The difficulty is set by an algorithm and not something set by the miner. The miner could try changing the rules re: difficulty (i.e., and mine a blockchain that forks) but other clients wouldn’t accept blocks as those blocks violate the Bitcoin protocol.

Sekioh

September 12, 2011, 09:02:28 PM

Quote from: Stephen Gornick on September 12, 2011, 08:26:42 PM

The difficulty is set by an algorithm and not something set by the miner. The miner could try changing the rules re: difficulty (i.e., and mine a blockchain that forks) but other clients wouldn’t accept blocks as those blocks violate the Bitcoin protocol.

But this isn’t quite true, as seen on the Geist coin and possibly soon the Namecoin, there’s an attack where all other 49% accept the new forking of the chain as the original, and the 51% attacker can set accepts/declines at will. They ‘mined’ 20000 blocks in a hour instead of the previous 20000 taking days. ‘violating’ the protocol is only applicable to the few that are in the 49% and try to change rules, whoever has more than half makes the rules it seems from what I gather based on these attacks.
The majority can't change the rules used by the minority. The minority will always reject invalid blocks, even if the minority is just one person.

His coins will not be worth very much then...

Erm, no, those blocks are *valid*.

By exploiting the fact that retargeting ignores one block interval every period, it's possible for an attacker's fork chain to "jump backwards in time" and create lots of blocks at low difficulty without running nTime off into the far future.

Bitcoin (and most *coin) rules re. block timestamps:

nTime has to be > median of prev 11 blocks.
nTime has to be < now() + some buffer.

let's say we have a chain with 4-block interval and 10 sec/block.

Official chain, current diff for hashrate, blocks found at nominal time:

<table>
<thead>
<tr>
<th>blk#</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>150</td>
</tr>
</tbody>
</table>

Now here's the weird part, we retarget after blocks 3, 7, 11, 15, and for block 3 we use 0 as first and 3 as last, for 7 we use 4 as first and 7 as last, ...

so what happens if an attacker's chain has blk timestamps like this:

<table>
<thead>
<tr>
<th>blk#</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>30</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>70</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>110</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>150</td>
</tr>
</tbody>
</table>

? first period (#3 - #0) is 30s as before 2nd period is (#7 - #4) ... 66s 3rd period is (#11 - #8) ... 104s Whoops.

Obviously this ignores the "problem" of the attackers chain having way lower sum-of-difficulty but that's easy to fix:

<table>
<thead>
<tr>
<th>blk#</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
</tr>
</tbody>
</table>

just keep driving diff up at maximum speed until you have the same total work as the real chain. result-> the attackers chain does not violate the block timestamp rules, finishes at a *earlier* block timestamp than the real chain, ends up at a higher total work as the real chain, but contains way more blocks.

This was done on the GeistGeld chain yesterday/today, so it's not a theoretical problem.

Thanks for the clear explanation ArtForz!

It's clear that the calculation needs to be made piecewise continuous, so for 7 we use 3 as the first.
real chain, no matter how many blocks is in your chain or how you've manipulated the difficulty? Please explain further...

ByteCoin

ArtForz  
Sr. Member  
Activity: 406

Re: Possible way to make a very profitable 50 plus ish attack for pools?  
September 12, 2011, 10:10:58 PM  
No, obviously you need to do the same or more total hashes as the real chain (it is a 51% attack...) The "bad" part is that you can make your chain have more blocks while having the same start and end nTime. And yes, it *should* be using 3-7, 7-11, ... but it doesn't. (probably to avoid the issue of the first interval after genesis, as you'd need to know when hashing of genesis started = the timestamp of the block before genesis). The code I'm currently playing with gets around this by special-casing that first retarget to have a nInterval-1 span instead of nInterval.

ByteCoin  
Sr. Member  
Activity: 416

Re: Possible way to make a very profitable 50 plus ish attack for pools?  
September 12, 2011, 10:48:48 PM  
Oh, and this also means this isn't a too massive issue for bitcoin simply due to the scale and time required (attacker needs more hashrate than the real net over at least 4-5 real retargeting periods for it to be somewhat effective, more time = more profit), but it could be pretty bad news for altchains.

Depends, obviously this one is destructive, but there's a few variations on it. A "fun" one that doesn't require orphaning massive parts of the chain:
With decently > 50%, only fork the chain if legit miners find > 5 blocks in a row (as that'd reset the median to where it should be) or the last block in a diff period (as that would prevent the wanted diff drop).
Broadcast blocks as usual.
Result: while things go mostly as normal (well, with quicker blocks the network now has over twice the hashpower and a few shortish chain forks getting orphaned), nTime of the chain except for the last block in each period basically stops increasing, and difficulty starts dropping after 2-3 retargeting periods. So basically "a majority of miner hashpower can decide to near-arbitrarily lower difficulty, and there's nothing short of a chain lockin or changing the retarget rules to stop em"

Re: Possible way to make a very profitable 50 plus ish attack for pools?  
September 12, 2011, 11:08:32 PM  
another variant, which doesn't orphan anything: get a *massive* majority of miners to switch to nTime rules like this for mining:
if ((pPrevBlock->nHeight+2) % retarget_block_count == 0)
set nTime as normal
else
block->nTime = medianofprev11()+1
Once enough miners switch to this so that "legit miners finding 6 blocks in a row" gets unlikely enough so it doesn't occur over 2+ diff periods, difficulty starts decreasing.

Adding some limits to make it head towards any arbitrary difficulty instead of going for "maximum decrease" is pretty simple.

Re: Possible way to make a very profitable 50 plus ish attack for pools?

September 13, 2011, 01:17:34 PM

The code I'm currently playing with gets around this by special-casing that first retarget to have a nInterval-1 span instead of nInterval.

Presumably if the Bitcoin developers decided they needed to change the difficulty algorithm fix this issue, they could just as easily change over from nInterval-1 to nInterval at a specified block some time in the future?

Re: Possible way to make a very profitable 50 plus ish attack for pools?

September 13, 2011, 04:29:22 PM

Yes. Obviously you'd put a chain checkpoint right before the switchover to prevent any future 51% attack from mucking with earlier blocks.

GeistGeld reloaded is the first chain testing retargeting over the full period, we'll see how that goes.

Quote from: ctoon6 on September 16, 2011, 12:52:13 AM

if somebody makes a block that does not fit in the rules of your client, it rejects it, simple as that. the rules your client has set can not be changed no matter how many bad blocks it rejects. the only thing an attack like that would do is waste yours and their time and their energy and money. sure they could make some transactions look like they never existed, but i don't see that going on for long, as it would cost large amounts of energy.

we are at 13.8 Thash/s only to be predicted to grow. id imagine, in less than 60 days, it will be over 14Thash/s

thats about 18,000 6990s

and i doubt you could even buy half that many. and at USD770 each, it would be USD13M, not counting the energy to run all that.

or another scenario, you use cpus, since there several orders of magnitude more cpus than gpus, it would take like 650000 1100T's to get less than half of the hashing power.

overall, its not possible, you literally have to match or beat everything the ENTIRE network throws at you. it would be a never ending stream of spending money. although you do not need 50% to do damage, you could start doing damage at like 30% on an irregular basis.

What part of the word "pool" do you not understand? I'm not talking about beating the total hash rate of the network, but about enough miners colluding to maximize profits. For every miner that joins, the hash rate of the honest part will get smaller and easier to beat. I didn't get the technical part quite right, but ArtForz has already explained how it could be done.