

Robert Prast and Sean Coneys  
Asis CTF Writeup  
04/30/2018

Problem: Neighbour (62)

Description: For a given large integer  $n$ , find the **nearest perfect power** that is equal or smaller than  $n$ .

$N$  is given once a string( $X$ ) is found where  $\text{sha256}(X)[-6:] = \text{*Randomized hex code*}$

Overview:

The crux of the problem revolved around finding the nearest perfect square(e.g 4, 9, 16...etc) to a large given number. In its simplest form, a perfect power is simply  $m^k = n$ , where  $m$  and  $k$  have to be positive integers and therefore  $n$  will be a natural number. However instead of focusing on just one exponent, our search would have to be over a series of different exponents and bases.

Beginning Steps:

Upon connecting to the server, you are presented with the following output.

```
penguin@penguin:~/Desktop$ nc 37.139.22.174 11740
*****
Welcome to the Near Neighbor Problem, only mathematicians know the flag!! |
Your mission is to find smallest natural number  $r$  for given  $n$ , such that |
there exist integers  $x, y$ :  $n - x^y = r > 0$ ,  $x > 1$  and  $y > 1$ . good luck!! |
*****
Submit a printable string  $X$ , such that  $\text{sha256}(X)[-6:] = \text{d37879}$ 
```

From this we can extract our formula, which is  $n - x^y = r$ , where  $r > 0$ ,  $x, y > 1$ , and our goal is to find the smallest possible value for  $r$ . However, in order to get a value for  $N$  and begin the optimization problem, we must first tackle finding a string, such that  $\text{sha256}(x)[-6:] = \text{"series of psuedo random hex bytes"}$  (always 6 long). We decided on going for a simple brute force approach. Essentially, we designed a for loop to test a large number of values. Each iteration of the loop would take the iteration number, and then take the

```
import hashlib
for i in range(100000000):
    m=hashlib.sha256()
    m.update(str(i))
    code=m.hexdigest()
    if(code[-6:]=="d37879"):
        print("HERE")
        print(i)
        x=rawinput()
```

sha256 hash of that  $i$ 'th number (as a string). Then there is a simple comparison between the last 6 digits of the  $i$ 'th hash digest and the given digest on the server. Since there are many solutions due the short length, we simply break the for loop after the first value is found. This gave us the output of 268684. After the server spits back a large value for a  $n$ . Allowing us the begin the hard math.

## Part 2:

At first we planned on just doing a simple brute force, with some logical parameters. Since the given  $n$  was upwards of 100 digits, (we even got one that 500+ digits), we erroneously assumed the value for  $y$  (the exponent) would also have to be relatively large. Therefore, we wrote a quick for loop to have a set  $x$  (base) and then an ever increasing exponent value. We would then look at  $N-x^{**}y$  and quickly saw that this method had some fatal errors. For starters, the base would also have to change. This can be seen with the following example:

$N=12 \quad x=2 \rightarrow y=\{2,3\}$  . This gives the smallest value for  $R$  being 4.

$N=12 \quad x=3 \rightarrow y=\{2\}$  . This gives the smallest value for  $R$  being 3, and thus smaller than base of 2.

After reviewing this information, we assumed the new correct path to take would be to use a theorem solver, specifically z3. However, this ended up giving inconclusive results since finding a minimum value using z3 is quite difficult. Therefore we abandoned this path, in favor of a more efficient brute force.

We initially planned to write an algorithm that checks  $N-x^y$ , where  $x$  is determined in an initial for loop, and  $y$  is nested within that – essentially trying every base with every exponent (where  $x$  and  $y$  are integers). Theoretically this would work, however with these extremely large  $N$  numbers the amount of iterations would have to be extraordinary. Therefore, we implemented a binary search to find the correct exponent and base, so as not to exceed the  $n$  value/ break the equation. After doing this, finding the nearest perfect power is found very quickly. Since the challenge would give you multiple  $n$  values to find different  $r$ 's, we simply changed the  $n$  value in the script each round. This eventually led to the flag which is:  
ASIS{36812f76cce2753e482ac6f68f9d3012}

```
1 import math
2 values=[]
3 def power(x,n):
4     #Find the highest possible power value for a specific x (large number) and exponent
5     highValue = 1
6     while highValue ** n < x:
7         highValue = highValue * 2
8         lowValue = highValue/2
9
10    #Single Binary Search with comparison to middleValue**n, essentially x**y in formula
11    while lowValue < highValue:
12        middleValue = (lowValue + highValue) // 2
13        if middleValue ** n < x:
14            lowValue = middleValue
15        elif middleValue ** n > x:
16            highValue = middleValue
17        else:
18            return middleValue
19    return middleValue + 1
20
21 #X value is N in the equation
22 x=334009051292362739039654215107316536500608100057171075238353981752053421098502658566140702123330403940015363809070461564050576923174238173383678275452600806207300177435711756213679555904610389560378779139731400240710566176315073653
23
24 #Try all exponent from 2, 500 (I assumed 500 would be big enough)
25 for i in range(2,500):
26     #Find the function large number, and exponent value for comparison
27     y=power(x,i)
28     print(n,y)
29     print("*****")
30     print(x**y)
31     #Print the binary to record all the values
32     dic[x**y]=i
33     #Store the array of all the values in r=n*x**y, we want to find the smallest value
34     values.append(x**y)
35     values.sort()
36
37 #Grab lowest value, and find its key-value pair in dic
38 print("*****")
39 print(values[0])
40 print("A")
41 print(dic.get(values[0]))
```

```
penguin@penguin: ~/Desktop
penguin@penguin:~/Desktop$ nc 37.139.22.174 11740
*****
| Welcome to the Near Neighbor Problem, only mathematicians know the flag!! |
| Your mission is to find smallest natural number r for given n, such that |
| there exist integers x, y: n = x**y = r > 0, x > 1 and y > 1, good luck!! |
*****
Submit a printable string X, such that sha256(X)[-6:] = d37879
208064
Please be patient ...
It takes a few seconds to load ...
n = 225515595153145397165168587515990724578415936178213094752475883912320216249252236842378292756893859504924216552781223093243230034297281228652591803188049077915779633910587140589000271526417448667679
542280551976091149607176144809837989859529036137715282156588702909308900703345180083622008275609245437991251959122846039386297439964656608577879633716551321393095291723590992819245143252790843286098137
965483957381608451595009572928216982166356517268154455174773487140373811720585665716915507273772133758800763440951070346960199695009990894128490223863209831629026509903266874681732629019289402818353155
50036354916270154993601754972878621419432854566974961316562424359919639220432864743285263654983103039832277651418093711250683936091878946942114996012370779290386297892693586589952377435408065384589799
3207126207649833408558051137136590802147827852938583089675958031851726831865866131750754026643565705960482196141343046863783135170494197110805527549008846854480074677970677429603937337971182530883980806
00294187877318544501360937403724960674568513326673061
To win the flag, submit r :
25739777006843198548105442119683750819589285
Great! :) please pass the next stage :P
n = 57752399297762163938380084125776662113732092695855621957348962372522597368284734418949232247310536159463466729454606729719717106207433066069911014803966690651310137914258945393263198151776771289
311539763678565233776213389382631548967287118996651237307104528728418777626607412556140848646186584300143948805514011498341217156609301091437131696486757955844534074603210529993310135330092936842
08981198175167779106864788528962156704225161864769075402431457262858772770163804761089186989802932651808319575106717819411459157
To win the flag, submit r :
1214709010392021246565099346095088139575106717819411459157
Great! :) please pass the next stage :P
n = 11912778158379464430797572930571318249627241487879165243013263635134809610145820989502882293793187347348555057629642450257308402138936817512022958343619
To win the flag, submit r :
47725975324366083042994
Great! :) please pass the next stage :P
n = 1921761107440216074462658998480076381596739916668771108506378267326742883543373342045945843105644143079356404133661651210264321083265126069187746422856889137871084750928297359945502804802142513795911
411175409513010313494441542614104370860635948695407864445787763011162270982770256972973100705028273693221798326867012478541887832261564051833398763820878918000344797557437709931111093760917638861072884
079120106982362369157195304010995600484652607618690468701259969778939498323734912579163680498344826745759589804779350412541587771261284604147054979692994594516811729361440913540730426711422823250921035065
4562479926099223242079178037442669220695639748035998670210059823928733519114506715869900743690285564211983562582406076451483172607419821571127854717342869573530602840827762370504316200122350259660108518
308166980460992815310608151917821881023468495504651423401530423176255938794649
To win the flag, submit r :
165525524758075326196530557356258538375572327363265476963479079911087124676
Great! :) please pass the next stage :P
n = 25248185084993336749923203903688281666022836263859835858465149311754026835920111996485762495565515984138050173173919562464654344702527290190827940835636165316394102157947595625864729112427633579093874
6527855169604884795043446707260749022346700850651537446789361749055088074258726597704987604873216452129171064409028639365102394019123357562923746483194549544587352592066184402384166689244682941119397
8803254388291633182073301398841710325965081742560029118481160397378207920260017786804311588201280184947148562329790821789399454786726714059737741047900175183065647344221861816116388708243744538924457517
863757431665008186197994784001126618991567453215917659986035260278833496661425351542929781779618464363254712545332802651795931204553875097439668090761135442988255282670823815925600030037933189650283244
50236273223769982955532161433297069762081968432819499566666268039800233092451996745424136702652792
To win the flag, submit r :
564714910780530506660268089800233092451996745424136702652792
Great! :) please pass the next stage :P
n = 1483757174674815290809524549515816201769276697968048536071310930956394658308637865027631924485273806218947234712630383025598216531980612754298339717651236277926053976416078303500441798122809708023
8412600733902930900635151806836204754531582799068460514750937485053582699386354320402175074297159502263698514265931059781895299132718131100242038785682652105650810054595948002170414306463634107435369786
9984095489
To win the flag, submit r :
2150484983310940792817922755936019799108308698040
Great! :) please pass the next stage :P
n = 3794594895786237289627005772197979145092922967841082378018420068146421295739156929134701213346687840095594431327467378566900347334893531171646988700105250610
To win the flag, submit r :
1911001853000935791277277813826022285631262704520144656608454483
Great! :) please pass the next stage :P
n = 1238760227085610459960580367184306373471061083128941315591934315954318908810944687176202400750231184073333457938076936889074136859141919270110045391385331107416968122204790662539922087958933447386539
729978494515904195640674913282997393813681304657039643779922954400374160769556995341329885014599603775110084915561208915279790146521792191234375182992500794110874979931284276547567602862535740778179181854
03216746517920024116344002487083962094768938639848614385493941853357107404156696513627022840341045326674256382181687082742807508634551132663236648809217713351001527138530645171920276668008439723311752
42039459915577978229707546441390741609796731604880248171226143771083618793546211
```



```
penguin@penguin: ~/Desktop
To win the flag, submit r :)
25739777606843198548105442119683750819589285
Great! :) please pass the next stage :P
n = 5775239929772621639383008412577666211373209269585556219573489623725225973682847344189492322247310536159463466729464506072971917106267433060689911014803966669651310137914258945393263198151776771289
81153976036785625239337762375389382634154896732871109966512373871045287520418776266074123561408486426180584380143948805154011498341217156689301891437316966486757955844534074603218529993310135330092936842
0891119817510716779310066470052896213670425216186476987549243145726285877270166304761089186968902932653088139575106717819411459157
To win the flag, submit r :)
1214709010392021246565099346095088139575106717819411459157
Great! :) please pass the next stage :P
n = 119127701158378464440757293057131024962724148787916524301326363513480961014582098950288229379318734734855505762964245025730848213893368175120229583433619
To win the flag, submit r :)
47725975324366083042994
Great! :) please pass the next stage :P
n = 192176110744021667446265899480376381596739916668771108503678267326742883543373342045945843105644143079356404133366161521026432188326512606918774642285688913781084750928297359945502804802142513795911
411175469513010313494041542614104378086035948695407864445787763011162270982770256972973100705028273693222179832686701247854188787322615640518333987638288789180003447957557437709931111093760917638861672884
079120100982326369157195304010995600484652607618690468701259969778939498323374912579163680498344826754759589804779350412541587771261284604147054979692994594516811729361440913540738426711422823250921035065
456242792609922324207917803744266922609563974803599867021005982392873351911145867158690074369028556421198356258240607645148317266741982157112785471734286957330602848827762370504316280122350259666108518
3081669046095201531860181510717821881023468495504651423401530423176255938794649
To win the flag, submit r :)
165525524758075326196530557356258538375572327363265476963479070911087124676
Great! :) please pass the next stage :P
n = 23248106509499333671099239390368826676602283626385983585046514931175402683592011199640576240955651159841380581731739195624646543447025272901982794035636165316394102157947595625864729112427633579093874
6527851509094884795043446789726074982234070085065153744678916174905508887042587265977049870848732126452129171064409028639365102394019123357562923746483194549544587352592066184402384166689244682941119397
80032543882916331820733013988417103259650817425600291104811603973782079202600177868043115882012801849471485628329790827189399454786726714059737741047900175178306564734422186181616388708243744538924457517
8637574316650081861979947840011266189915674532159176599806352602788334966614253515429297817796184643632547125455328205217959312045538750974309680907611354429882552826708238159256800030037933189650283244
582362732367998029555321614332970697628819684328194905066602668039800233092451996745424136702652792
To win the flag, submit r :)
56471491078505305066602668039800233092451996745424136702652792
Great! :) please pass the next stage :P
n = 148375717467481582980095245495158162017692706979608485936873109360563949650300837085027631924485273886210947247323638380255982105319804175429839717651236237926053976160783835044179012280979823
841260075390293096063515806362647545315827996846051475093748505358269933663542840217507429715950226369851426593105978109529913127161311002426307850626521056501005439594860217041436643634107435369766
9984095489
To win the flag, submit r :)
2156484983310940792817922755936619739108308698040
Great! :) please pass the next stage :P
n = 259377860004649520218166421833382458640339572217668343302092114241339348317339811924058001402299132086833396388618584799365900675690553478256400063788532875951879365511674618781619011890781348195986720
3794594895786263728962700577219797914509292296784108237801042006814642129573915692913470121334668784009559443132746737856900347334893531171646988700105250610
To win the flag, submit r :)
1911001853000935759622772704138326032285631262704520144656008454483
Great! :) please pass the next stage :P
n = 12387602270856104599605803671843063734710610831289413155919343159543189088109446871762024007502311840733334579538076936889074136859141919270110045391385331107416968122204790662539922087958933447386539
729978494515904195640674913282997393813681304657039643779922954400374166769556953413298850114590663775110084915561208915279790146521792191234375182992500794116874979931284276547567602862535740778179181854
832107465719206214163448024070839628947689386390846143854939410533571074841566965136270228403341045328674256382101687082742807250863455113266323064880921771335106101527138530645171920276668008439723311752
42039459051535797782297075464413907416697560731684880248171226143771683618793546211
To win the flag, submit r :)
63498650649994461076782275734367012370644290
Great! :) please pass the next stage :P
n = 59087099047301305512023329798358746776394491204795511657883528434701354707865473175724970555358132591732193747659279142291773325145028824054884697402413683122580081248147387655921922897933832784918594
453021909101502241472614977988939591916158444506157047730241388
To win the flag, submit r :)
64201133290969596083202412
Great! :) please pass the next stage :P
n = 33400905129236273983968241518731653650026810005717107523835398175285342109850265856614078212333049394801536383098704615640585769231742381733836782754526008462873801774357117562136795590461038956037877
9513973140024071056617631587365319435035286951943828317060026410605500819053877017353146879063263432893616019610626227018292979186197591889522257164838877544393822995199447706928602603477096785324057055
0558465296959878987420397692865992781898883066151082128571689491327980460839074276553946903322245452698888292982820432776710207391242148431556433768963691930298852498655148867433775457668529013124204668
19439699514951973140081801523375308609183249026236744884406502827906414550607462400000000000000000043732107328074331
To win the flag, submit r :)
43732107328074331
Congrats! :) You got the flag: ASIS{36812f76cce2753e482ac6f8fd9d3012}
penguin@penguin: ~/Desktop
```