MythX

Analysis b94cb32d-dee4-4f6c-85b4-3f24c2949910 MythX

Started	Mon Feb 19 2024 21:54:40 GMT+0000 (Coordinated Universal Time)
Finished	Mon Feb 19 2024 21:54:47 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Vscode-Extension
Main Source File	/Flatten/Simpleerc721psihx.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	21

ISSUES

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
224 function average(uint256 a, uint256 b) internal pure returns (uint256) {
225 // (a + b) / 2 can overflow.
226 return (a & b) + (a ^ b) / 2;
227 }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
224 function average(uint256 a, uint256 b) internal pure returns (uint256) {
225 // (a + b) / 2 can overflow.
226 return (a & b) + (a ^ b) / 2;
227 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
235 function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
236 // (a + b - 1) / b can overflow on addition, so we distribute.
237 return a == 0 ? 0 : (a - 1) / b + 1;
238 }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
235 function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
236 // (a + b - 1) / b can overflow on addition, so we distribute.
237 return a == 0 ? 0 : (a - 1) / b + 1;
238 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
235 function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
236 // (a + b - 1) / b can overflow on addition, so we distribute.
237 return a == 0 ? 0 : (a - 1) / b + 1;
238 }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
261 // The surrounding unchecked block does not change this fact.
262 // See https://docs.soliditylang.org/en/latest/control-structures.html#checked-or-unchecked-arithmetic.
263 return prod0 / denominator;
264 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
286
287 // Does not overflow because the denominator cannot be zero at this stage in the function.
288 uint256 twos = denominator & (~denominator + 1);
289 assembly {
290 // Divide denominator by twos.
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
299
300 // Shift in bits from prod1 into prod0.
301 prod0 |= prod1 * twos;
302
303 // Invert denominator mod 2^256. Now that denominator is an odd number, it has an inverse modulo 2^256 such
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
304 // that denominator * inv = 1 mod 2^256. Compute the inverse by starting with a seed that is correct for
305 // four bits. That is, denominator * inv = 1 mod 2^4.
306 uint256 inverse = (3 * denominator) ^ 2;
307
308 // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
```

UNKNOWN Arithmetic operation "*=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
308 // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
309 // in modular arithmetic, doubling the correct bits in each step.
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
308 // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
309 // in modular arithmetic, doubling the correct bits in each step.
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
308 // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
309 // in modular arithmetic, doubling the correct bits in each step.
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
```

UNKNOWN Arithmetic operation "*=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
309 // in modular arithmetic, doubling the correct bits in each step.
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
309 // in modular arithmetic, doubling the correct bits in each step.
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
309 // in modular arithmetic, doubling the correct bits in each step.
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
```

UNKNOWN Arithmetic operation "*=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
310 inverse *= 2 - denominator * inverse; // inverse mod 2^8
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
```

UNKNOWN Arithmetic operation "*=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
311 inverse *= 2 - denominator * inverse; // inverse mod 2^16
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

UNKNOWN Arithmetic operation "*=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
312 inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
312 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
312 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
313 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

UNKNOWN Arithmetic operation "*=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
313 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
316 |
317 | // Because the division is now exact we can divide by multiplying with the modular inverse of denominator.
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
313 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
316 |
317 | // Because the division is now exact we can divide by multiplying with the modular inverse of denominator.
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
313 inverse *= 2 - denominator * inverse; // inverse mod 2^64
314 inverse *= 2 - denominator * inverse; // inverse mod 2^128
315 inverse *= 2 - denominator * inverse; // inverse mod 2^256
316
317 // Because the division is now exact we can divide by multiplying with the modular inverse of denominator.
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
319 // less than 2^256, this is the final result. We don't need to compute the high bits of the result and prod1
320 // is no longer required.
321 result = prod0 * inverse;
322 return result;
323 }
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
330 uint256 result = mulDiv(x, y, denominator);
331 if (rounding == Rounding.Up && mulmod(x, y, denominator) > 0) {
332 result += 1;
333 }
334 return result;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
362 // into the expected uint128 result.
363 unchecked {
364 result = (result + a / result) >> 1;
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
362 // into the expected uint128 result.
363 unchecked {
364 result = (result + a / result) >> 1;
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
363 unchecked {
364 result = (result + a / result) >> 1;
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
363 unchecked {
364 result = (result + a / result) >> 1;
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
364 result = (result + a / result) >> 1;
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
364 result = (result + a / result) >> 1;
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
365 result = (result + a / result) >> 1;
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
366 result = (result + a / result) >> 1;
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
371 return min(result, a / result);
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
367 result = (result + a / result) >> 1;
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
371 return min(result, a / result);
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
371 return min(result, a / result);
372 }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
368 result = (result + a / result) >> 1;
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
371 return min(result, a / result);
372 }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
369 result = (result + a / result) >> 1;
370 result = (result + a / result) >> 1;
371 return min(result, a / result);
372 }
373 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
379 unchecked {
380 uint256 result = sqrt(a);
381 return result + (rounding == Rounding.Up && result * result < a ? 1 : 0);
382 }
383 }
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
379 unchecked {
380 uint256 result = sqrt(a);
381 return result + (rounding == Rounding.Up && result * result < a ? 1 : 0);
382 }
383 }
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
392 if (value >> 128 > 0) {
393 value >>= 128;
394 result += 128;
395 }
396 if (value >> 64 > 0) {
397 value >>= 64;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
396 if (value >> 64 > 0) {
397 value >>= 64;
398 result += 64;
399 }
400 if (value >> 32 > 0) {
401 value >>= 32;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
400 if (value >> 32 > 0) {
401 value >>= 32;
402 result += 32;
403 }
404 if (value >> 16 > 0) {
405 value >>= 16;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
404 if (value >> 16 > 0) {
405 value >>= 16;
406 result += 16;
407 }
408 if (value >> 8 > 0) {
409 value >>= 8;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
408 if (value >> 8 > 0) {
409 value >>= 8;
410 result += 8;
411 }
412 if (value >> 4 > 0) {
413 value >>= 4;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
412 if (value >> 4 > 0) {
413 value >>= 4;
414 result += 4;
415 }
416 if (value >> 2 > 0) {
417 value >>= 2;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
416 if (value >> 2 > 0) {
417 value >>= 2;
418 result += 2;
419 }
420 if (value >> 1 > 0) {
421 result += 1;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
419 }
420 if (value >> 1 > 0) {
421 result += 1;
422 }
423 }
424 return result;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
432 unchecked {
433 uint256 result = log2(value);
434 return result + (rounding == Rounding.Up && 1 << result < value ? 1 : 0);
435 }
436 }
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
443 uint256 result = 0;
444 unchecked {
445 if (value >= 10 ** 64) {
446 value /= 10 ** 64; value /= 10 ** 64;
447 result += 64;
448 }
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
444 unchecked {
445 if (value >= 10 ** 64) {
446 value /= 10 ** 64;
447 result += 64;
448 }
449 if (value >= 10 ** 32) {
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
444 unchecked {
445 if (value >= 10 ** 64) {
446 value /= 10 ** 64;
447 result += 64;
448 }
449 if (value >= 10 ** 32) {
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
445 if (value >= 10 ** 64) {
446 value /= 10 ** 64;
447 result += 64;
448 }
449 if (value >= 10 ** 32) {
450 value /= 10 ** 32;
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
447 result += 64;
448 }
449 if (value >= 10 ** 32) {
450 value /= 10 ** 32; value /= 10 ** 32;
451 result += 32;
452 }
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
448 }
449 if (value >= 10 ** 32) {
450 value /= 10 ** 32;
451 result += 32;
452 }
453 if (value >= 10 ** 16) {
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
448 }
449 if (value >= 10 ** 32) {
450 value /= 10 ** 32;
451 result += 32;
452 }
453 if (value >= 10 ** 16) {
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
449 if (value >= 10 ** 32) {
450 value /= 10 ** 32;
451 result += 32;
452 }
453 if (value >= 10 ** 16) {
454 value /= 10 ** 16;
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
451 result += 32;
452 }
453 if (value >= 10 ** 16) {
454 value /= 10 ** 16; value /= 10 ** 16;
455 result += 16;
456 }
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
452 }
453 if (value >= 10 ** 16) {
454 value /= 10 ** 16;
455 result += 16;
456 }
457 if (value >= 10 ** 8) {
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
452 }
453 if (value >= 10 ** 16) {
454 value /= 10 ** 16;
455 result += 16;
456 }
457 if (value >= 10 ** 8) {
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
453 if (value >= 10 ** 16) {
454 value /= 10 ** 16;
455 result += 16;
456 }
457 if (value >= 10 ** 8) {
458 value /= 10 ** 8;
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
455 result += 16;
456 }
457 if (value >= 10 ** 8) {
458 value /= 10 ** 8; value /= 10 ** 8;
459 result += 8;
460 }
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
456 }
457 if (value >= 10 ** 8) {
458 value /= 10 ** 8;
459 result += 8;
460 }
461 if (value >= 10 ** 4) {
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
456 }
457 if (value >= 10 ** 8) {
458 value /= 10 ** 8;
459 result += 8;
460 }
461 if (value >= 10 ** 4) {
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
457 if (value >= 10 ** 8) {
458 value /= 10 ** 8;
459 result += 8;
460 }
461 if (value >= 10 ** 4) {
462 value /= 10 ** 4;
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
459 result += 8;
460 }
461 if (value >= 10 ** 4) {
462 value /= 10 ** 4; value /= 10 ** 4;
463 result += 4;
464 }
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
460 }
461 if (value >= 10 ** 4) {
462 value /= 10 ** 4;
463 result += 4;
464 }
465 if (value >= 10 ** 2) {
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
460 }
461 if (value >= 10 ** 4) {
462 value /= 10 ** 4;
463 result += 4;
464 }
465 if (value >= 10 ** 2) {
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
461 if (value >= 10 ** 4) {
462 value /= 10 ** 4;
463 result += 4;
464 }
465 if (value >= 10 ** 2) {
466 value /= 10 ** 2;
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
463 result += 4;
464 }
465 if (value >= 10 ** 2) {
466 value /= 10 ** 2; value /= 10 ** 2;
467 result += 2;
468 }
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
464 }
465 if (value >= 10 ** 2) {
466 value /= 10 ** 2;
467 result += 2;
468 }
469 if (value >= 10 ** 1) {
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
464 }
465 if (value >= 10 ** 2) {
466 value /= 10 ** 2;
467 result += 2;
468 }
469 if (value >= 10 ** 1) {
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
465 if (value >= 10 ** 2) {
466 value /= 10 ** 2;
467 result += 2;
468 }
469 if (value >= 10 ** 1) {
470 result += 1;
```

UNKNOWN Arithmetic operation "**" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
467 result += 2;
468 }
469 if (value >= 10 ** 1) {
470 result += 1; result += 1;
471 }
472 }
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
468 }
469 if (value >= 10 ** 1) {
470 result += 1;
471 }
472 }
473 return result;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
481 unchecked {
482 uint256 result = log10(value);
483 return result + (rounding == Rounding.Up && 10 ** result < value ? 1 : 0);
484 }
485 }
```

UNKNOWN Arithmetic operation "**" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
481 unchecked {
482 uint256 result = log10(value);
483 return result + (rounding == Rounding.Up && 10 ** result < value ? 1 : 0);
484 }
485 }
```

UNKNOWN Arithmetic operation "+=" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
496 if (value >> 128 > 0) {
497 value >>= 128;
498 result += 16;
499 }
500 if (value >> 64 > 0) {
501 value >>= 64;
```

UNKNOWN Arithmetic operation "+=" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
500 if (value >> 64 > 0) {
501 value >>= 64;
502 result += 8;
503 }
504 if (value >> 32 > 0) {
505 value >>= 32;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
504 if (value >> 32 > 0) {
505 value >>= 32;
506 result += 4;
507 }
508 if (value >> 16 > 0) {
509 value >>= 16;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
508 if (value >> 16 > 0) {
509 value >>= 16;
510 result += 2;
511 }
512 if (value >> 8 > 0) {
513 result += 1;
```

UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
511 }
512 if (value >> 8 > 0) {
513 result += 1;
514 }
515 }
516 return result;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
524 unchecked {
525 uint256 result = log256(value);
526 return result + (rounding == Rounding.Up && 1 << (result << 3) < value ? 1 : 0);
527 }
528 }
529 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
561 function average(int256 a, int256 b) internal pure returns (int256) {
562 // Formula from the book "Hacker's Delight"
563 int256 x = (a & b) + ((a ^ b) >> 1);
564 return x + (int256(uint256(x) >> 255) & (a ^ b));
565 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
562 // Formula from the book "Hacker's Delight"
563 int256 x = (a & b) + ((a ^ b) >> 1);
564 return x + (int256(uint256(x) >> 255) & (a ^ b));
565 }
566
567 /**
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
597 function toString(uint256 value) internal pure returns (string memory) {
598 unchecked {
599 uint256 length = Math.log10(value) + 1;
600 string memory buffer = new string(length);
601 uint256 ptr;
602 /// @solidity memory-safe-assembly
```

UNKNOWN Arithmetic operation "--" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
605 }
606 while (true) {
607 ptr--;
608 /// @solidity memory-safe-assembly
609 assembly {
610 mstore8(ptr, byte(mod(value, 10), _SYMBOLS))
```

UNKNOWN Arithmetic operation "/=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
610 mstore8(ptr, byte(mod(value, 10), _SYMBOLS))
611 }
612 value /= 10;
613 if (value == 0) break;
614 }
615 return buffer;
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
629 function toHexString(uint256 value) internal pure returns (string memory) {
630 unchecked {
631 return toHexString(value, Math.log256(value) + 1);
632 }
633 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
637 */
638 function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
639 bytes memory buffer = new bytes(2 * length + 2);
640 buffer[0] = "0";
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
637 */
638 function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
639 bytes memory buffer = new bytes(2 * length + 2);
640 buffer[0] = "0";
641 buffer[1] = "x";
```

UNKNOWN

Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
640 buffer[0] = "0";
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
643 buffer[i] = _SYMBOLS[value & 0xf];
644 value >>= 4;
```

UNKNOWN

Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
640 buffer[0] = "0";
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
643 buffer[i] = _SYMBOLS[value & 0xf];
644 value >>= 4;
```

UNKNOWN

Arithmetic operation "--" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
640 buffer[0] = "0";
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
643 buffer[i] = _SYMBOLS[value & 0xf]; buffer[i] = _SYMBOLS[value & 0xf];
644 value >>= 4;
645 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1565 require(bb > 0);
1566 unchecked {
1567 return bb & (0 - bb);
1568 }
1569 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1584 bb |= bb >> 1;
1585
1586 return (bb >> 1) + 1;
1587 }
1588 }
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1593 function bitScanForward256(uint256 bb) pure internal returns (uint8) {
1594 unchecked {
1595 return uint8(LOOKUP_TABLE_256[(isolateLS1B256(bb) * DEBRUIJN_256) >> 248]);
1596 }
1597 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1602 function bitScanReverse256(uint256 bb) pure internal returns (uint8) {
1603 unchecked {
1604 return 255 - uint8(LOOKUP_TABLE_256[((isolateMS1B256(bb) * DEBRUIJN_256) >> 248)]);
1605 }
1606 }
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1602 function bitScanReverse256(uint256 bb) pure internal returns (uint8) {
1603 unchecked {
1604 return 255 - uint8(LOOKUP_TABLE_256[((isolateMS1B256(bb) * DEBRUIJN_256) >> 248)]);
1605 }
1606 }
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1608 function log2(uint256 bb) pure internal returns (uint8) {
1609 unchecked {
1610 return uint8(LOOKUP_TABLE_256[(isolateMS1B256(bb) * DEBRUIJN_256) >> 248]);
1611 }
1612 }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1704
1705 unchecked {
1706 if(bucketStartIndex + amount < 256) {
1707 bitmap._data[bucket] |= MASK_FULL << (256 - amount) >> bucketStartIndex;
1708 } else {
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1705 unchecked {
1706 if(bucketStartIndex + amount < 256) {
1707 bitmap._data[bucket] |= MASK_FULL << (256 - amount) >> bucketStartIndex;
1708 } else {
1709 bitmap._data[bucket] |= MASK_FULL >> bucketStartIndex;
```

UNKNOWN

Arithmetic operation "-=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1708 } else {
1709 bitmap._data[bucket] |= MASK_FULL >> bucketStartIndex;
1710 amount -= (256 - bucketStartIndex);
1711 bucket++;
1712
1713 while(amount > 256) {
```

UNKNOWN

Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1708 } else {
1709 bitmap._data[bucket] |= MASK_FULL >> bucketStartIndex;
1710 amount -= (256 - bucketStartIndex);
1711 bucket++;
1712
1713 while(amount > 256) {
```

UNKNOWN

Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1709 bitmap._data[bucket] |= MASK_FULL >> bucketStartIndex;
1710 amount -= (256 - bucketStartIndex);
1711 bucket++;
1712
1713 while(amount > 256) {
1714 bitmap._data[bucket] = MASK_FULL;
1715 amount -= 256;
```

UNKNOWN Arithmetic operation "-=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1713 while(amount > 256) {
1714 bitmap._data[bucket] = MASK_FULL;
1715 amount -= 256;
1716 bucket++;
1717 }
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1714 bitmap._data[bucket] = MASK_FULL;
1715 amount -= 256;
1716 bucket++;
1717 }
1718
1719 bitmap._data[bucket] |= MASK_FULL << (256 - amount);
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1717 }
1718
1719 bitmap._data[bucket] |= MASK_FULL << (256 - amount);
1720 }
1721 }
1722 }
```

UNKNOWN

Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1732
1733 unchecked {
1734 if(bucketStartIndex + amount < 256) {
1735 bitmap._data[bucket] &= ~(MASK_FULL << (256 - amount) >> bucketStartIndex);
1736 } else {
```

UNKNOWN

Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1733 unchecked {
1734 if(bucketStartIndex + amount < 256) {
1735 bitmap._data[bucket] &= ~(MASK_FULL << (256 - amount) >> bucketStartIndex);
1736 } else {
1737 bitmap._data[bucket] &= ~(MASK_FULL >> bucketStartIndex);
```

UNKNOWN

Arithmetic operation "-=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1736 } else {
1737 bitmap._data[bucket] &= ~(MASK_FULL >> bucketStartIndex);
1738 amount -= (256 - bucketStartIndex);
1739 bucket++;
1740
1741 while(amount > 256) {
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1736 } else {
1737 bitmap._data[bucket] &= ~(MASK_FULL >> bucketStartIndex);
1738 amount -= (256 - bucketStartIndex);
1739 bucket++;
1740
1741 while(amount > 256) {
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1737 bitmap._data[bucket] &= ~(MASK_FULL >> bucketStartIndex);
1738 amount -= (256 - bucketStartIndex);
1739 bucket++;
1740
1741 while(amount > 256) {
1742 bitmap._data[bucket] = 0;
1743 amount -= 256;
```

UNKNOWN Arithmetic operation "-=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1741 while(amount > 256) {
1742 bitmap._data[bucket] = 0;
1743 amount -= 256;
1744 bucket++;
1745 }
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1742 bitmap._data[bucket] = 0;
1743 amount -= 256;
1744 bucket++;
1745 }
1746
1747 bitmap._data[bucket] &= ~(MASK_FULL << (256 - amount));
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1745 }
1746
1747 bitmap._data[bucket] &= ~(MASK_FULL << (256 - amount));
1748 } }
1749 }
1750 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1768 if(bb > 0) {
1769 unchecked {
1770 setBitIndex = (bucket << 8) | (bucketIndex - bb.bitScanForward256());
1771 }
1772 } else {
```

UNKNOWN Arithmetic operation "--" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1774 require(bucket > 0, "BitMaps: The set bit before the index doesn't exist.");
1775 unchecked {
1776 bucket--;
1777 }
1778 // No offset. Always scan from the least significiant bit now.
1779 bb = bitmap._data[bucket];
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1781 if(bb > 0) {
1782 unchecked {
1783 setBitIndex = (bucket << 8) | (255 - bb.bitScanForward256());
1784 break;
1785 }
1786 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1867 */
1868 function _totalMinted() internal view virtual returns (uint256) {
1869 return _currentIndex - _startTokenId();
1870 }
1871
```

UNKNOWN Arithmetic operation "++" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1900
1901 uint count;
1902 for( uint i = _startTokenId(); i < _nextTokenId(); ++i ){
1903 if(_exists(i)){ if(_exists(i)){
1904 if( owner == ownerOf(i)){
1905 ++count;
```

UNKNOWN Arithmetic operation "++" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1903 if(_exists(i)){
1904 if( owner == ownerOf(i)){
1905 ++count;
1906 }
1907 }
1908 }
```

UNKNOWN Arithmetic operation "+=" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2172
2173 _beforeTokenTransfers(address(0), to, nextTokenId, quantity);
2174 _currentIndex += quantity;
2175 _owners[nextTokenId] = to;
2176 _batchHead.set(nextTokenId);
2177 _afterTokenTransfers(address(0), to, nextTokenId, quantity);
```

UNKNOWN Arithmetic operation "+" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2178
2179 // Emit events
2180 for(uint256 tokenId=nextTokenId; tokenId < nextTokenId + quantity; tokenId++){
2181 emit Transfer(address(0), to, tokenId);
2182 }
```

UNKNOWN Arithmetic operation "++" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2178
2179 // Emit events
2180 for(uint256 tokenId=nextTokenId; tokenId < nextTokenId + quantity; tokenId++){
2181 emit Transfer(address(0), to, tokenId);
2182 }
2183 }
```

UNKNOWN Arithmetic operation "+" discovered

SWC-101

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2213 _approve(address(0), tokenId);
2214
2215 uint256 subsequentTokenId = tokenId + 1;
2216
2217 if(!_batchHead.get(subsequentTokenId) &&
2218 subsequentTokenId < _nextTokenId()
2219 ) {
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2262 if (to.isContract()) {
2263 r = true;
2264 for(uint256 tokenId = startTokenId; tokenId < startTokenId + quantity; tokenId++){
2265 try IERC721Receiver(to).onERC721Received(_msgSender(), from, tokenId, _data) returns (bytes4 retval) {
2266 r = r && retval == IERC721Receiver.onERC721Received.selector;
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2262 if (to.isContract()) {
2263 r = true;
2264 for(uint256 tokenId = startTokenId; tokenId < startTokenId + quantity; tokenId++){
2265 try IERC721Receiver(to).onERC721Received(_msgSender(), from, tokenId, _data) returns (bytes4 retval) {
2266 r = r && retval == IERC721Receiver.onERC721Received.selector;
2267 } catch (bytes memory reason) {
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2303 uint256 tokenIdsLength = balanceOf(owner);
2304 uint256[] memory tokenIds = new uint256[](tokenIdsLength);
2305 for (uint256 i = _startTokenId(); tokenIdsIdx != tokenIdsLength; ++i) {
2306 if (_exists(i)) { if (_exists(i)) {
2307 if (ownerOf(i) == owner) {
2308 tokenIds[tokenIdsIdx++] = i;
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2306 if (_exists(i)) {
2307 if (ownerOf(i) == owner) {
2308 tokenIds[tokenIdsIdx++] = i;
2309 }
2310 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2413 */
2414 function totalSupply() public view virtual override returns (uint256) {
2415 return _totalMinted() - _burned();
2416 }
2417
2418 /**
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2421 function _burned() internal view returns (uint256 burned){
2422 uint256 startBucket = _startTokenId() >> 8;
2423 uint256 lastBucket = (_nextTokenId() >> 8) + 1;
2424
2425 for(uint256 i=startBucket; i < lastBucket; i++) {
2426 uint256 bucket = _burnedToken.getBucket(i);
2427 burned += _popcount(bucket);
```

UNKNOWN

Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2423 uint256 lastBucket = (_nextTokenId() >> 8) + 1;
2424
2425 for(uint256 i=startBucket; i < lastBucket; i++) {
2426 uint256 bucket = _burnedToken.getBucket(i); uint256 bucket = _burnedToken.getBucket(i);
2427 burned += _popcount(bucket);
2428 }
```

UNKNOWN

Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2425 for(uint256 i=startBucket; i < lastBucket; i++) {
2426 uint256 bucket = _burnedToken.getBucket(i);
2427 burned += _popcount(bucket);
2428 }
2429 }
```

UNKNOWN

Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2434 function _popcount(uint256 x) private pure returns (uint256 count) {
2435 unchecked{
2436 for (count=0; x!=0; count++)
2437 x &= x - 1;
2438 }
2439 }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2435 unchecked{
2436 for (count=0; x!=0; count++)
2437 x &= x - 1;
2438 }
2439 }
2440 }
```

UNKNOWN Arithmetic operation "++" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2466 ) external onlyRole(MINTER_ROLE) {
2467 uint256 pre = totalSupply();
2468 for (uint i = 0; i < ids.length; i++) {
2469 require(origin == ownerOf(ids[i]), "ERC721PsiHX: Not token owner"); require(origin == ownerOf(ids[i]), "ERC721PsiHX: Not token owner");
2470
2471 _burn(ids[i]);
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2473 uint256 post = totalSupply();
2474
2475 require(pre - post == ids.length, "ERC721PsiHX: Burning error");
2476 }
```

UNKNOWN Compiler-rewritable "<uint> - 1" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

/flatten/simpleerc721psihx.sol

Locations

```
235 function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
236 // (a + b - 1) / b can overflow on addition, so we distribute.
237 return a == 0 ? 0 : (a - 1) / b + 1;
238 }
```

UNKNOWN

SWC-101

Compiler-rewritable "<uint> - 1" discovered

This plugin produces issues to support false positive discovery within MythX.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2435 unchecked{
2436 for (count=0; x!=0; count++)
2437 x &= x - 1;
2438 }
2439 }
2440 }
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
8  // OpenZeppelin Contracts v4.4.1 (access/IAccessControl.sol)
9
10 pragma solidity ^0.8.0;
11
12 /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
100 // OpenZeppelin Contracts (last updated v4.9.4) (utils/Context.sol)
101
102 pragma solidity ^0.8.0;
103
104 /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
132 // OpenZeppelin Contracts v4.4.1 (utils/introspection/IERC165.sol)
133
134 pragma solidity ^0.8.0;
135
136 /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
161 // OpenZeppelin Contracts v4.4.1 (utils/introspection/ERC165.sol)
162
163 pragma solidity ^0.8.0;
164
165 /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
192 // OpenZeppelin Contracts (last updated v4.9.0) (utils/math/Math.sol)
193
194 pragma solidity ^0.8.0;
195
196 /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
535 // OpenZeppelin Contracts (last updated v4.8.0) (utils/math/SignedMath.sol)
536
537 pragma solidity ^0.8.0;
538
539 /**
540 * @dev Standard signed math utilities missing in the Solidity language.
541 */
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
582 // OpenZeppelin Contracts (last updated v4.9.0) (utils/Strings.sol)
583
584 pragma solidity ^0.8.0;
585
586
587 /**
588 * @dev String operations.
589 */
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
669 // OpenZeppelin Contracts (last updated v4.9.0) (access/AccessControl.sol)
670
671 pragma solidity ^0.8.0;
672
673
674
675
676 /**
677 * @dev Contract module that allows children to implement role-based access
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
919 // OpenZeppelin Contracts (last updated v4.9.0) (token/ERC721/IERC721.sol)
920
921 pragma solidity ^0.8.0;
922
923 /**
924 * @dev Required interface of an ERC721 compliant contract.
925 */
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1053 // OpenZeppelin Contracts v4.4.1 (interfaces/IERC721.sol)
1054
1055 pragma solidity ^0.8.0;
1056
1057
1058 // File @openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol@v4.9.5
1059
1060 // Original license: SPDX_License_Identifier: MIT
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1061 // OpenZeppelin Contracts v4.4.1 (token/ERC721/extensions/IERC721Metadata.sol)
1062
1063 pragma solidity ^0.8.0;
1064
1065 /**
1066 * @title ERC-721 Non-Fungible Token Standard, optional metadata extension
1067 * @dev See https://eips.ethereum.org/EIPS/eip-721
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.24"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1088
1089 // Original license: SPDX_License_Identifier: MIT
1090 pragma solidity ^0.8.24;
1091
1092 interface IERC721HX is IERC721 {
1093 function MINTER_ROLE() external returns (bytes32);
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1120 // OpenZeppelin Contracts (last updated v4.6.0) (token/ERC721/IERC721Receiver.sol)
1121
1122 pragma solidity ^0.8.0;
1123
1124 /**
1125 * @title ERC721 token receiver interface
1126 * @dev Interface for any contract that wants to support safeTransfers
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.1"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1151 // OpenZeppelin Contracts (last updated v4.9.0) (utils/Address.sol)
1152
1153 pragma solidity ^0.8.1;
1154
1155 /**
1156 * @dev Collection of functions related to the address type
1157 */
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1400 // This file was procedurally generated from scripts/generate/templates/StorageSlot.js.
1401
1402 pragma solidity ^0.8.0;
1403
1404 /**
1405 * @dev Library for reading and writing primitive types to specific storage slots.
1406 *
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. T

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1552 */
1553
1554 pragma solidity ^0.8.0;
1555
1556
1557 library BitScan {
1558 uint256 constant private DEBRUIJN_256 = 0x818283848586878898a8b8c8d8e8f929395969799a9b9d9e9faaeb6bedeeff;
1559 bytes constant private LOOKUP_TABLE_256 =
hex"0001020903110a19042112290b311a3905412245134d2a550c5d32651b6d3a7506264262237d468514804e8d2b95569d0d495ea533a966b11c886eb93bc176c9071727374353637324837e9b47af86c7155181ad4fd18ed32c9096db57d59ee30e2e
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1629
1630 */
1631 pragma solidity ^0.8.0;
1632
1633 /**
1634 * @dev This Library is a modified version of Openzeppelin's BitMaps library.
1635 * Functions of finding the index of the closest set bit from a given index are added.
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1811 */
1812
1813 pragma solidity ^0.8.0;
1814
1815
1816
1817
1818
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2367
2368 */
2369 pragma solidity ^0.8.0;
2370
2371
2372 abstract contract ERC721PsiBurnable is ERC721Psi {
2373 using BitMaps for BitMaps.BitMap;
2374 BitMaps.BitMap private _burnedToken;
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.24"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2444
2445 // Original license: SPDX_License_Identifier: MIT
2446 pragma solidity ^0.8.24;
2447
2448
2449
2450
2451 abstract contract ERC721PsiHX is IERC721HX, ERC721Psi, ERC721PsiBurnable, AccessControl {
2452 string public baseURI;
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.24"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2518
2519 // Original license: SPDX_License_Identifier: MIT
2520 pragma solidity ^0.8.24;
2521
2522 contract SimpleERC721PsiHX is ERC721PsiHX {
2523 constructor(
2524 address _defaultAdmin,
```

UNKNOWN

SWC-110

Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
638 function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
639 bytes memory buffer = new bytes(2 * length + 2);
640 buffer[0] = "0";
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
```

UNKNOWN

SWC-110

Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
639 bytes memory buffer = new bytes(2 * length + 2);
640 buffer[0] = "0";
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
643 buffer[i] = _SYMBOLS[value & 0xf];
```

UNKNOWN **Out of bounds array access**

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
643 buffer[i] = _SYMBOLS[value & 0xf];
644 value >>= 4;
645 }
```

UNKNOWN **Out of bounds array access**

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
641 buffer[1] = "x";
642 for (uint256 i = 2 * length + 1; i > 1; --i) {
643 buffer[i] = _SYMBOLS[value & 0xf];
644 value >>= 4;
645 }
646 require(value == 0, "Strings: hex length insufficient");
```

UNKNOWN **Out of bounds array access**

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1593 function bitScanForward256(uint256 bb) pure internal returns (uint8) {
1594 unchecked {
1595 return uint8(LOOKUP_TABLE_256[(isolateLS1B256(bb) * DEBRUIJN_256) >> 248]);
1596 }
1597 }
```

UNKNOWN

Out of bounds array access

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1602 function bitScanReverse256(uint256 bb) pure internal returns (uint8) {
1603 unchecked {
1604 return 255 - uint8(LOOKUP_TABLE_256[((isolateMS1B256(bb) * DEBRUIJN_256) >> 248)]);
1605 }
1606 }
```

UNKNOWN

Out of bounds array access

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
1608 function log2(uint256 bb) pure internal returns (uint8) {
1609 unchecked {
1610 return uint8(LOOKUP_TABLE_256[(isolateMS1B256(bb) * DEBRUIJN_256) >> 248]);
1611 }
1612 }
1613 }
```

UNKNOWN

Out of bounds array access

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2306 if (_exists(i)) {
2307 if (ownerOf(i) == owner) {
2308 tokenIds[tokenIdsIdx++] = i;
2309 }
2310 }
```

UNKNOWN Out of bounds array access

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2467 uint256 pre = totalSupply();
2468 for (uint i = 0; i < ids.length; i++) {
2469 require(origin == ownerOf(ids[i]), "ERC721PsiHX: Not token owner");
2470
2471 _burn(ids[i]);
```

UNKNOWN Out of bounds array access

SWC-110

The index access expression can cause an exception in case of use of invalid array index value.

Source file

/flatten/simpleerc721psihx.sol

Locations

```
2469 require(origin == ownerOf(ids[i]), "ERC721PsiHX: Not token owner");
2470
2471 _burn(ids[i]);
2472 }
2473 uint256 post = totalSupply();
```