



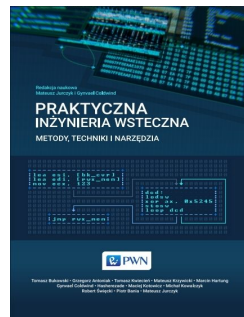
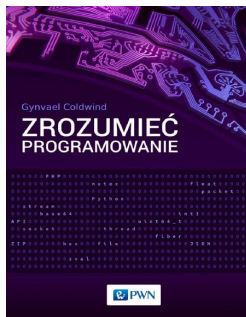
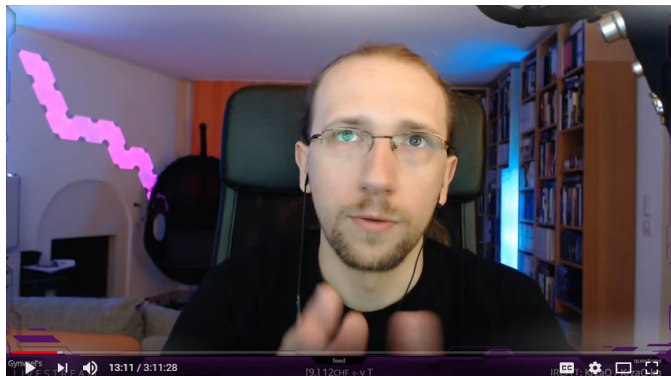
Szybkie wprowadzenie do bezpieczeństwa aplikacji

Gynael Coldwind, GDG/WTM Rzeszów - SecureIT 2018

O prelegencie

W Google od 2010.

Obecnie pracuje jako Tech Lead / Manager w ISE-RIP.

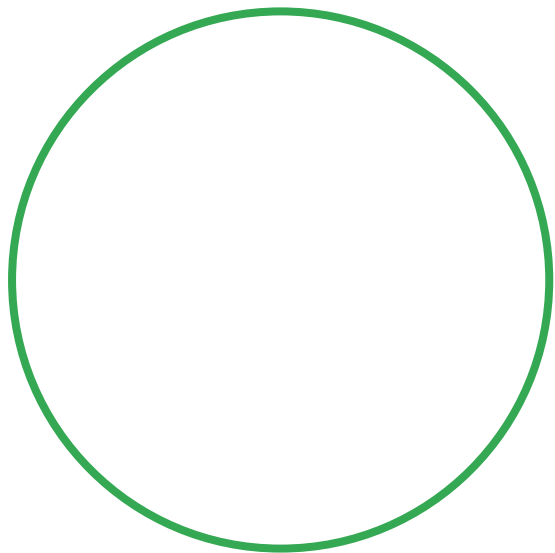


Jadłospis

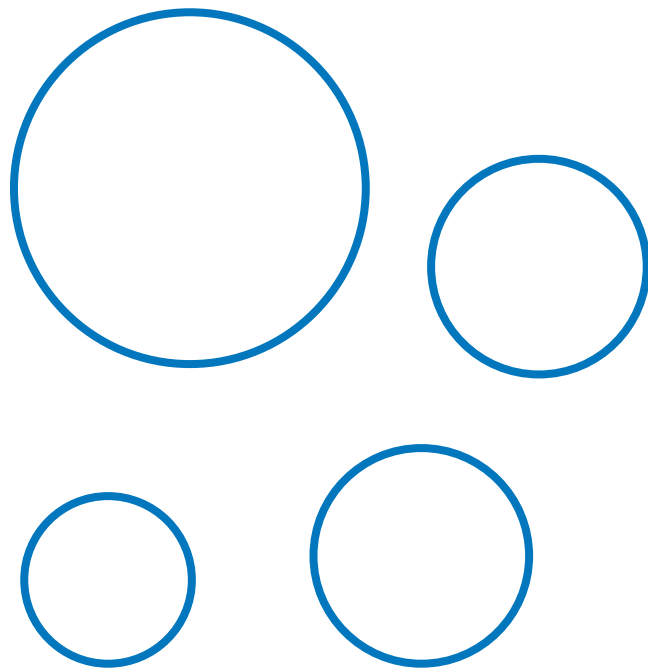
1. Długi wstęp
2. Studium przypadku (wybrane klasy błędów)
 - a. XSS
 - b. Domowe crypto
 - c. Blacklisty
 - d. Buffer Overflow
 - e. Serializacja
3. Zakończenie

Wstep

Błędy bezpieczeństwa a zwykłe bugi

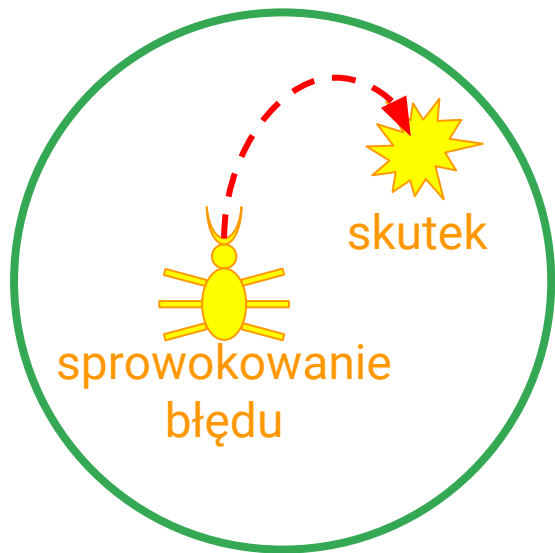


posiadany zbiór
uprawnień

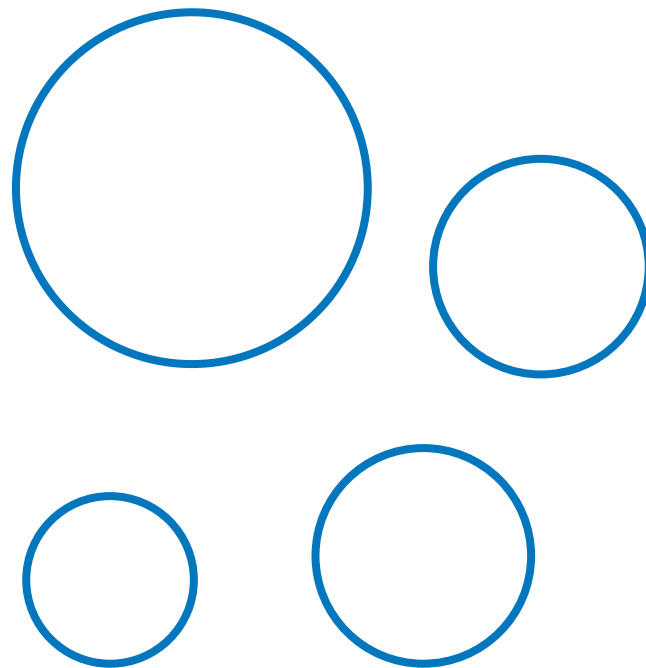


inne (nieposiadane)
uprawnienia

Błędy bezpieczeństwa a zwykłe bugi

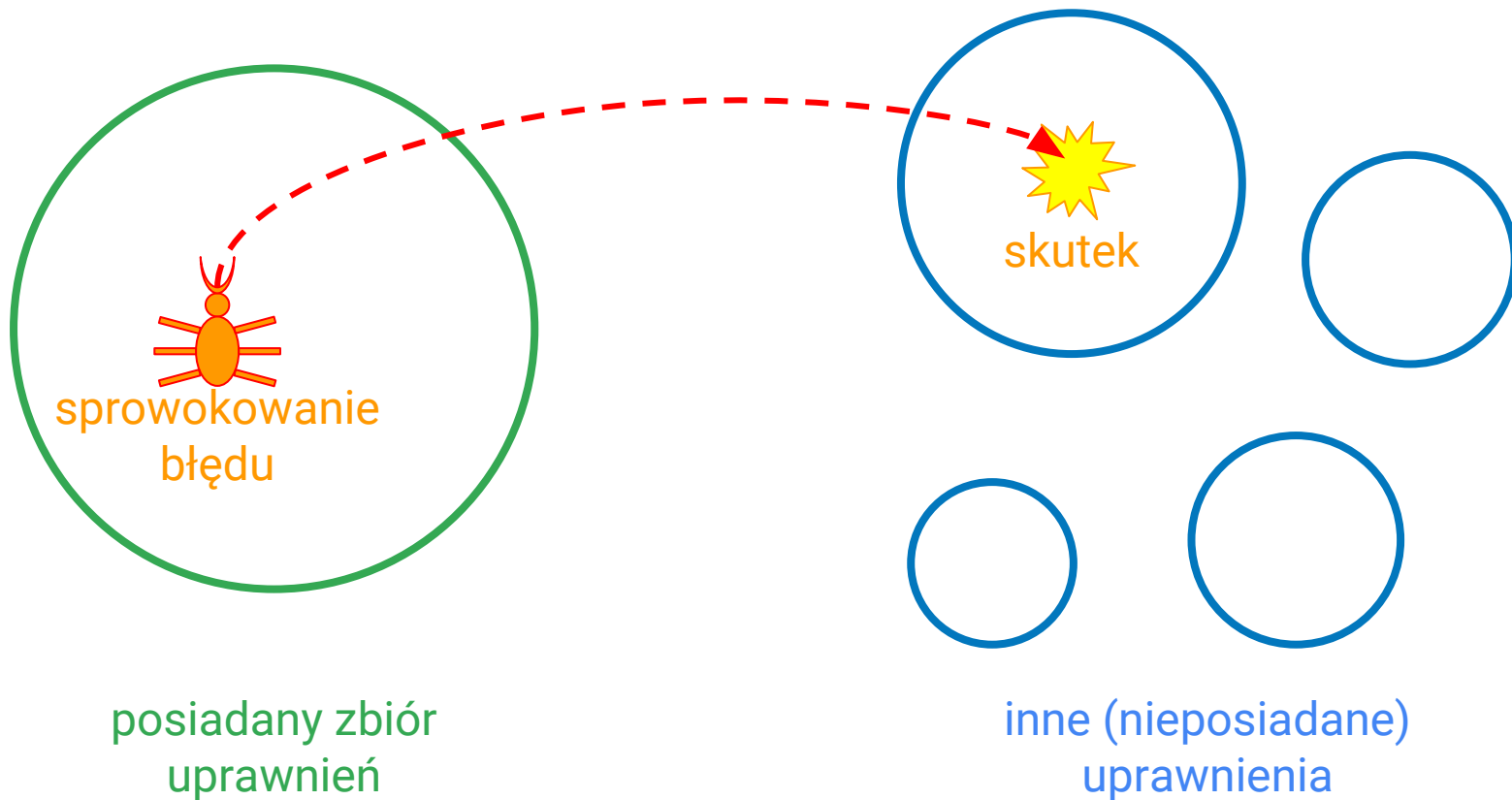


posiadany zbiór
uprawnień



inne (nieposiadane)
uprawnienia

Błędy bezpieczeństwa a zwykłe bugi



Skutki błędów bezpieczeństwa

- **Denial of Service (DoS)**

Coś wybucha.

- **Information Leak**

Coś wycieka.

- **(Local) Privilege Escalation**

Mamy roota!

- **Remote Code Execution (RCE)**

My other computer is YOUR computer.



**Arbitrary Code
Execution**

QUIZ! Bug czy Vuln, i jaki ma skutek?

1. Uruchomiłem kalkulator
2. Wpisałem `1234 * ASDF`
3. *Program wykonał nieprawidłową operację...*

BUG!

VULN!

QUIZ! Bug czy Vuln, i jaki ma skutek?

1. Uruchomiłem kalkulator
2. Wpisałem `1234 * ASDF`
3. *Program wykonał nieprawidłową operację...*

BUG!

QUIZ! Bug czy Vuln, i jaki ma skutek?

1. Wszedłem na <https://notmysite.com/>
2. Podałem użytkownika i hasło: ' **or 1=1** --
3. Zostałem zalogowany na konto administratora

BUG!

VULN!

QUIZ! Bug czy Vuln, i jaki ma skutek?

1. Wszedłem na <https://notmysite.com/>
2. Podałem użytkownika i hasło: ' **or 1=1** --
3. Zostałem zalogowany na konto administratora

Privilege Escalation
VULN!

QUIZ! Bug czy Vuln, i jaki ma skutek?

1. Otworzyłem plik PDF w moim ulubionym czytniku
2. *Program wykonał nieprawidłową operację...*

BUG!

VULN!

QUIZ! Bug czy Vuln, i jaki ma skutek?

1. Otworzyłem plik PDF w moim ulubionym czytniku
2. *Program wykonał nieprawidłową operację...*

Denial of Service ;)
Remote Code Execution

VULN!

Severity, Ryzyko, Trudność exploitacji, Wymagane interakcje

Czyli inne metryki używane przy ocenie błędów bezpieczeństwa.

Severity, Ryzyko, Trudność exploitacji, Wymagane interakcje

Jak poważne będą konsekwencje wykorzystania błędu?

RCE-->ring0

lang settings

Severity, **Ryzyko**, Trudność exploitacji, Wymagane interakcje

Jakie jest prawdopodobieństwo wykorzystania błędu?

Jakie są wymagania wstępne, żeby błąd wykorzystać?

stabilny

crypto ;)

zdalny błąd

Severity, Ryzyko, **Trudność exploitacji**, Wymagane interakcje

Jak łatwo jest dany błąd wykorzystać w praktyce?

SQLI

remote heap z
ASLR, one-shot
bez interakcji

Severity, Ryzyko, Trudność exploitacji, **Wymagane interakcje**

Jak bardzo użytkownik musi "pomóc" przy exploitacji?

Granica phishingu - Czy łatwiej jest...

1. Nakłonić użytkownika żeby "pomógł" nam wyexploitować dany błąd?
2. Czy żeby podał nam swoje hasło?

./exploit<enter>

**skomplikowany
self-XSS**

Błędy sklasyfikowane i niesklasyfikowane

Często powtarzające się rodzaje błędów zostały sklasyfikowane, np.:

- Buffer Overflow
 - Stack-based
 - Heap-based
 - Inne...
- XSS
 - Reflected
 - Stored
 - DOM
- XSRF
- SQL Injection
 - Blind SQL Injection

- Race Condition
 - Double Fetch (TOCTTOU)
 - ...

I tak dalej...

Oraz mniej znane, np.:

- HTTP Parameter Pollution
- Session Puzzling
- ...

Błędy sklasyfikowane i niesklasyfikowane

Niesklasyfikowane błędy nadal są groźne.

Testowy Serwis

DEMO

XSS

Cross Site Scripting

OWASP Top 10-2017 A7-Cross-Site Scripting (XSS)

```
<html>  
  <head>  
    <title><?=$_GET['title'];?></title>  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

Cross Site Scripting

OWASP Top 10-2017 A7-Cross-Site Scripting (XSS)

```
<html>
  <head>
    <title><?=$_GET['title'];?></title>
  </head>
  ...
```

<https://example.com/?title=ASDF>

Cross Site Scripting

OWASP Top 10-2017 A7-Cross-Site Scripting (XSS)

```
<html>  
  <head>  
    <title>ASDF</title>  
  </head>  
  ...
```

<https://example.com/?title=ASDF>



Cross Site Scripting

OWASP Top 10-2017 A7-Cross-Site Scripting (XSS)

```
<html>  
  <head>  
    <title><script>alert(1)</script></title>  
  </head>  
  ...
```

[https://example.com/?title=<script>alert\(1\)</script>](https://example.com/?title=<script>alert(1)</script>)



Cross Site Scripting

Severity?

Co można zrobić?

Cross Site Scripting

Jak ukryć ciasteczka: **httponly**

Problem XSS rozwiązany!

Cross Site Scripting - inne potencjalne rozwiązania

- X-XSS-Protection

- 0

- 1

- 1; mode=block

- Content Security Policy

- Escaping i auto-escaping, tainting

Mitygacje
Defense-in-depth

Rozwiązanie...?

XSS a upload plików

Problem:

A co jeśli atakujący wrzuci plik **.html**?

XSS a upload plików

Problem:

A co jeśli atakujący wrzuci plik ~~.html~~ **text/html**?



Content-Type: **text/html; charset=utf-8**

Content Sniffing? Flash?

XSS a upload plików - pomysł!

(1)

```
if (getimagesize($file['tmp_name']) === false) {  
    $upload_error = "getimagesize() says it's not an image file";  
    return false;  
}
```

(2)

```
$mime = mime_content_type($file);  
header("Content-Type: {$mime}");  
readfile($file);
```

XSS a upload plików - libmagic (plik sgml)

```
0          search/4096/cwt \<title
```

```
!:mime    text/html
```

```
!:strength + 5
```

```
0          search/4096/cwt \<html
```

```
!:mime    text/html
```

```
!:strength + 5
```


```
0          search/4096/cwt \<script
```

```
!:mime    text/html
```

```
!:strength + 5
```

XSS a upload plików - PHP (funkcja getimagesize)

"GIF"



```
if (!memcmp filetype, php_sig_gif, 3)) {  
    return IMAGE_FILETYPE_GIF;
```

XSS a upload plików

DEMO

XSS a upload plików - inny pomysł!

A może by tak zrekodować plik z obrazem?



JPEG



`imagecreatefromjpeg`



`imagejpeg`



JPEG

XSS a upload plików - inny pomysł!

A może by tak zrekodować plik z obrazem?

NOPE

JPEG



`imagecreatefromjpeg`



`imagejpeg`



JPEG

XSS a upload plików - inny pomysł!

A może by tak zrekodować plik z obrazem?



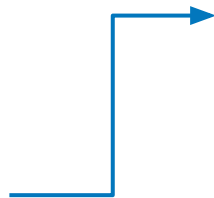
JPEG



`imagecreatefromjpeg`



`dodaj entropie`



`imagejpeg`



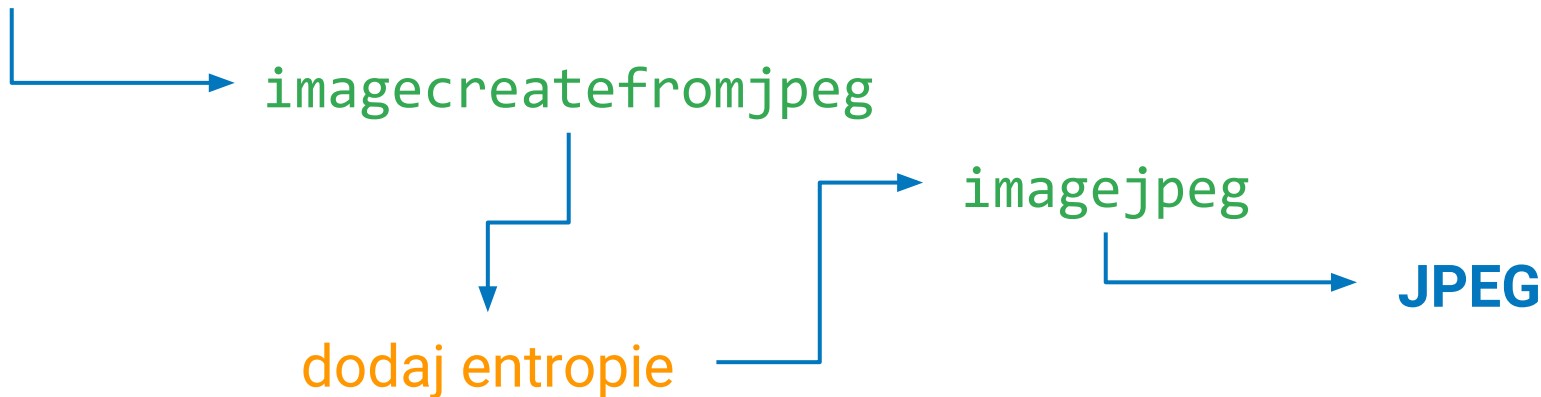
JPEG

XSS a upload plików - inny pomysł!

A może by tak zrekodować plik z obrazem?

NOPE

JPEG



XSS a upload plików - sandboxed domain

SOP

Nasza strona:

example.com

Strona z której są serwowane pliki użytkowników:

sandboxedexample.com

(to NIE jest subdomena)

Auth?

XSS a upload plików - sandboxed domain

Uwierzytelnienie?

[https://**sandboxedexample.com**/i/06837434c657a28106d6cb13bad419be8ac08f2eb4193c5fda49e5ed06c81a58a836f6e44668ad907ccdf6c172be247b75851b17f1f4f9abb427be8c79abbe83.jpg](https://sandboxedexample.com/i/06837434c657a28106d6cb13bad419be8ac08f2eb4193c5fda49e5ed06c81a58a836f6e44668ad907ccdf6c172be247b75851b17f1f4f9abb427be8c79abbe83.jpg)

enumerowanie
głębokie ukrycie

XSS a upload plików - sandboxed domain

Uwierzytelnienie?

[https://**sandboxedexample.com**/i/06837434c657a28106d6cb13bad419be8ac08f2eb4193c5fda49e5ed06c81a58a836f6e44668ad907ccdf6c172be247b75851b17f1f4f9abb427be8c79abbe83.jpg](https://sandboxedexample.com/i/06837434c657a28106d6cb13bad419be8ac08f2eb4193c5fda49e5ed06c81a58a836f6e44668ad907ccdf6c172be247b75851b17f1f4f9abb427be8c79abbe83.jpg)

?access_token=NDVjNmIzYjVmYmFmNmI1NTU4YzAwZGU2ZjBIYzBjYTBkYjQ1NjA5NToxMzc2OTg3NDIz

base64(hmac(path) + ':' + expire_time)

przykład

Kodowanie Szyfrowanie

DEMO

Kodowanie? Szyfrowanie?

Przykłady kodowań:

- UTF-8
- ASCII
- Base64
- Base32
- Base56
- Uuencode

Przykłady alg. szyfrujących:

- RC4
- AES
 - AES-128-CBC
 - AES-256-ECB
- RSA
- ECC

DEMO

Nie wdrażaj
własnego krypto

Pierwsze 42 zasady kryptografii

1. Nie wdrażaj własnego krypto
2. Nie wdrażaj własnego krypto
3. Nie wdrażaj własnego krypto
4. Nie wdrażaj własnego krypto
5. Nie wdrażaj własnego krypto
6. Nie wdrażaj własnego krypto
7. Nie wdrażaj własnego krypto
8. Nie wdrażaj własnego krypto
9. Nie wdrażaj własnego krypto
10. Nie wdrażaj własnego krypto
11. ...

"Any person can invent a security system so clever that he or she can't imagine a way of breaking it."

Schneier's law

Ale, ale...

Jeśli atakujący nie zna szyfru...

TO GO NIE POŁAMIE!

"A cryptosystem should be secure even if everything about the system, except the key, is public knowledge."

Kerckhoffs's principle

Patrz również: Security by Obscurity

Własne krypto, czyli XOR

Python-like pseudocode

```
key = "\x12\x34\x56\x78\x9A\xBC\xDE\xF0"
```

```
m = "... " # Dane do zaszyfrowania.
```

```
c = ""
```

```
for i in range(len(m)):
```

```
    c += m[i] ^ key[i % len(key)]
```

Kryptoanaliza XOR

Jak myśleć o XOR?

- Operacja na bitach
- P to wiadomość
- Q wybiera które bity zanegować
- (ew. odwrotnie)

P	Q	P XOR Q
0	0	0
0	1	1
1	0	1
1	1	0

P: 0 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1 0 0 1 1 1 1 1 0
Q: 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0
=: 0 1 1 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0

Długość klucza (OTP?)

DEMO

Prosty trick dla ASCII (0-127):

P: 0 ? ? ? ? ? ? ?

Q: 0 ? ? ? ? ? ? ?

=: 0 ? ? ? ? ? ? ?

P: 0 ? ? ? ? ? ? ?

Q: 1 ? ? ? ? ? ? ?

=: 1 ? ? ? ? ? ? ?

Kryptoanaliza XOR - *partial known plaintext*

Charakterystyka XOR:

$P \text{ XOR } Q \rightarrow X$

$X \text{ XOR } Q \rightarrow P$

$X \text{ XOR } P \rightarrow Q$

Szyfrogram \wedge Wiadomość \rightarrow Klucz

P	Q	P XOR Q
0	0	0
0	1	1
1	0	1
1	1	0

Kryptoanaliza XOR - *partial known plaintext*

DEMO

hash
hmac

Hasze, sumy kontrolne, kody korekcyjne

Przykłady:

- CRC32 (5129f3bd)
 - Hamming code
- SHA-256 (f0e4c2f76c58916ec258f246851bea091d14d42...)
- scrypt?
- hmacs?

HMAC

```
m = "... " # Wiadomość.
```

```
hmac = SHA256("supersecret" + m) # HMAC (zły).
```

Uprozczone działanie hashy

INITIALIZE:

rejestr = [stałe] (np. 4 * 32 bity)

UPDATE(blok danych): (np. 64 bajty)

rejestr = f(rejestr, blok danych)

FINALIZE(): (opcjonalnie)

rejestr = f(rejestr, resztko danych + padding)
(np. 77F869401DE682F60E0E749493AB793D)

Hash-extension attack

FINALIZE to w zasadzie **UPDATE**, więc można zrobić...

INITIALIZE

UPDATE(dane)

...

UPDATE(dane)

FINALIZE (UPDATE(dane+pad))

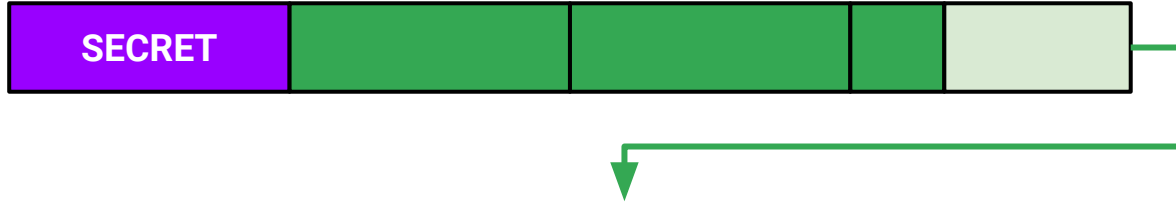
UPDATE(dodatkowe dane)

FINALIZE

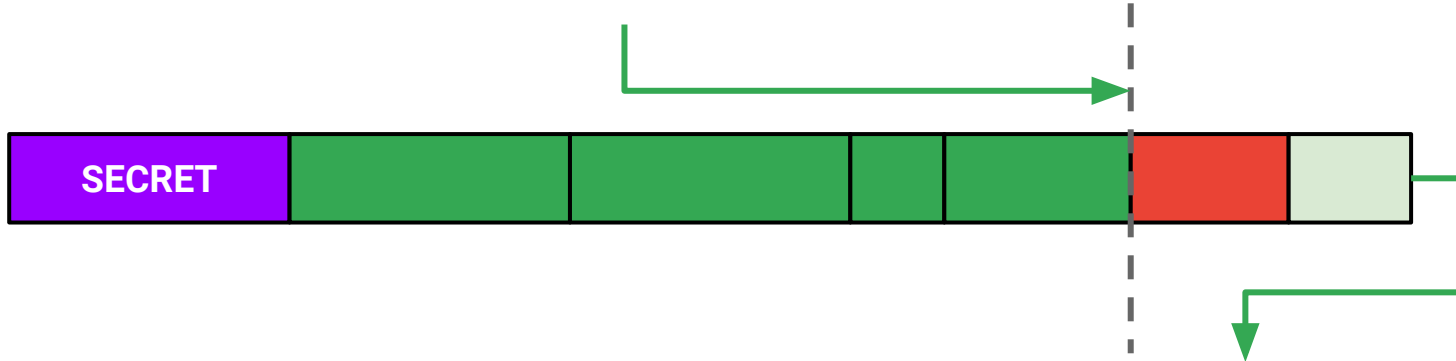
hash z
danych

hash z
danych +
stałej +
dodatkowych
danych

Hash-extension attack



77F869401DE682F60E0E749493AB793D



A3F2B9119EF874AD725EEC1322C139A0

Hash-extension attack

DEMO

blacklisty
whitelisty

Blacklisty, whitelisty - upload plików

```
$ext_blacklist = [  
    "php", "phps", "html"  
];
```

Blacklisty, whitelisty - upload plików

```
$ext_blacklist = [  
    "php", "php3", "php4", "php5", "php6",  
    "php7", "pht", "phtml", "phps",  
    "htm", "html", "xhtml"  
];
```

Blacklisty, whitelisty - upload plików

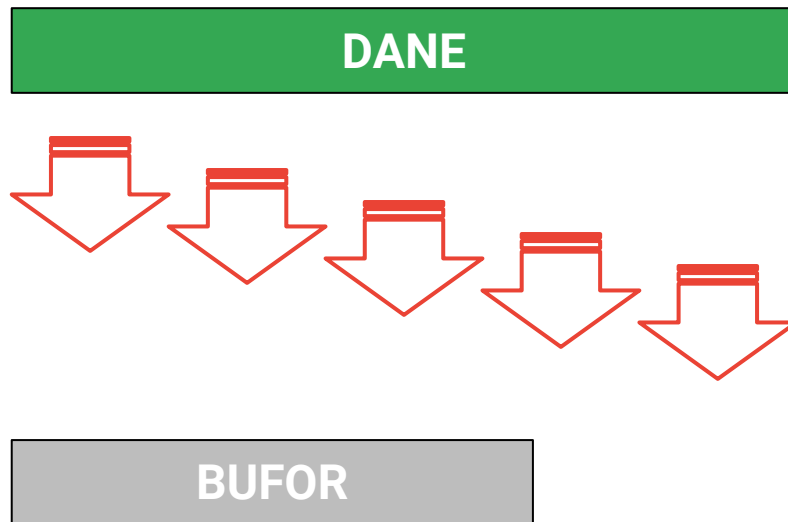
```
$ext_blacklist = [  
    "php", "php3", "php4", "php5", "php6",  
    "php7", "pht", "phtml", "phps",  
    "htm", "html", "xhtml"  
];  
<FilesMatch ".+\.ph(ar|p|tml)$">  
    SetHandler application/x-httpd-php
```

Blacklisty, whitelisty - upload plików

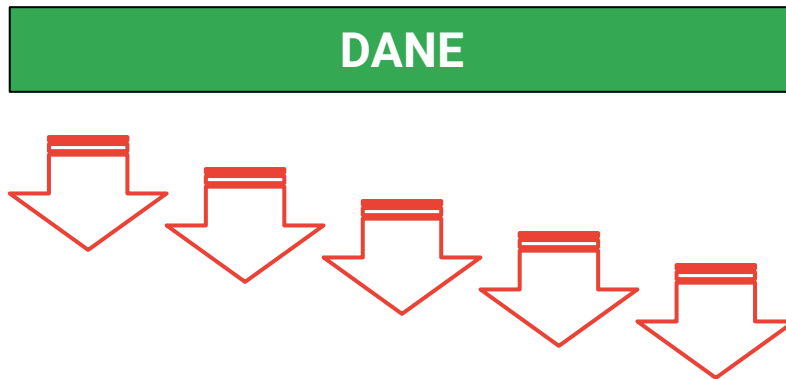
DEMO

Powrót do klasyki - błędy klasy Buffer Overfl

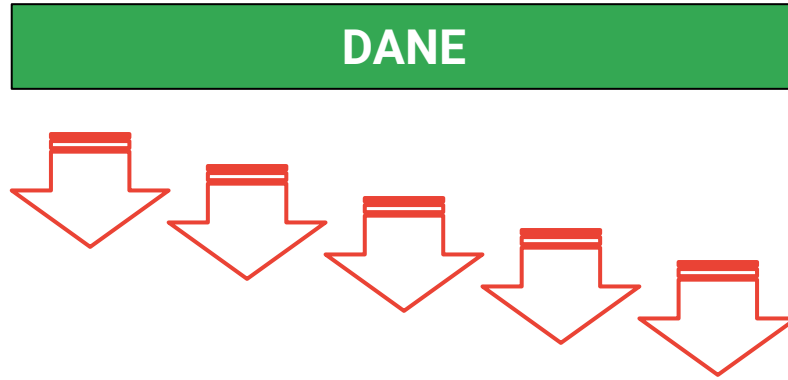
Buffer Overflow



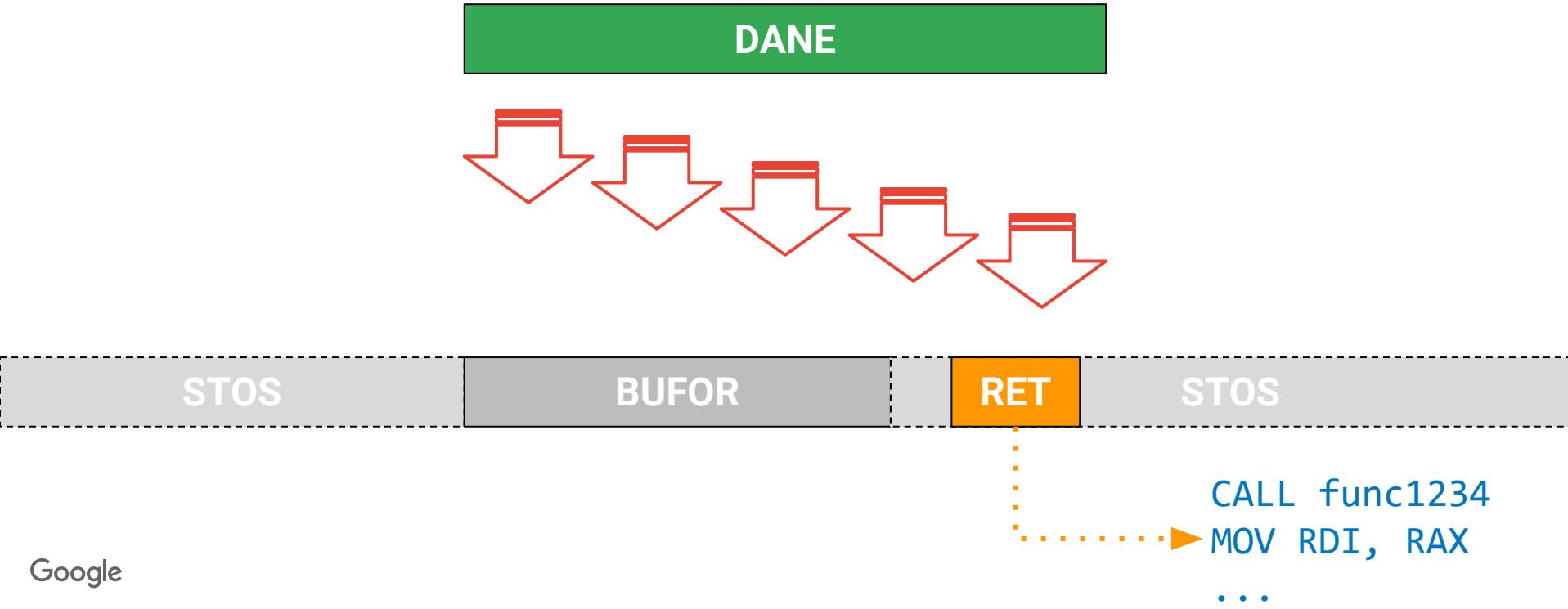
Buffer Overflow



Stack-based Buffer Overflow

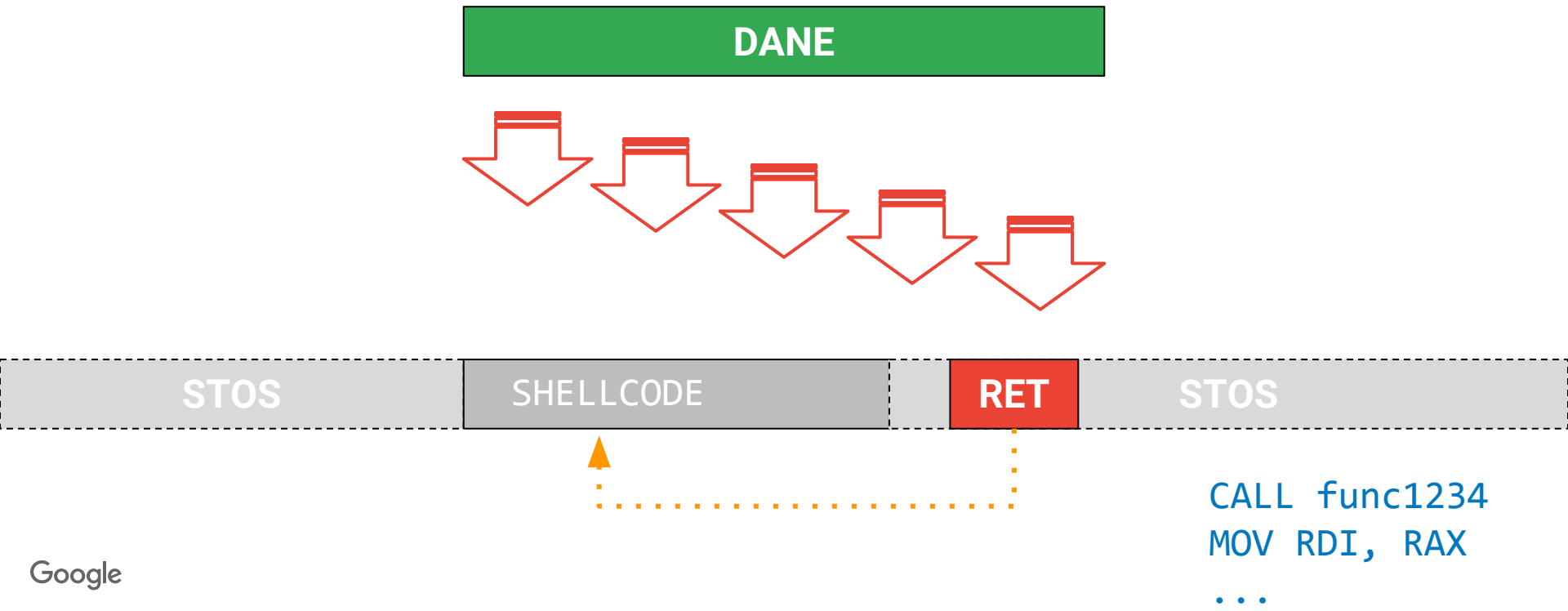


Stack-based Buffer Overflow

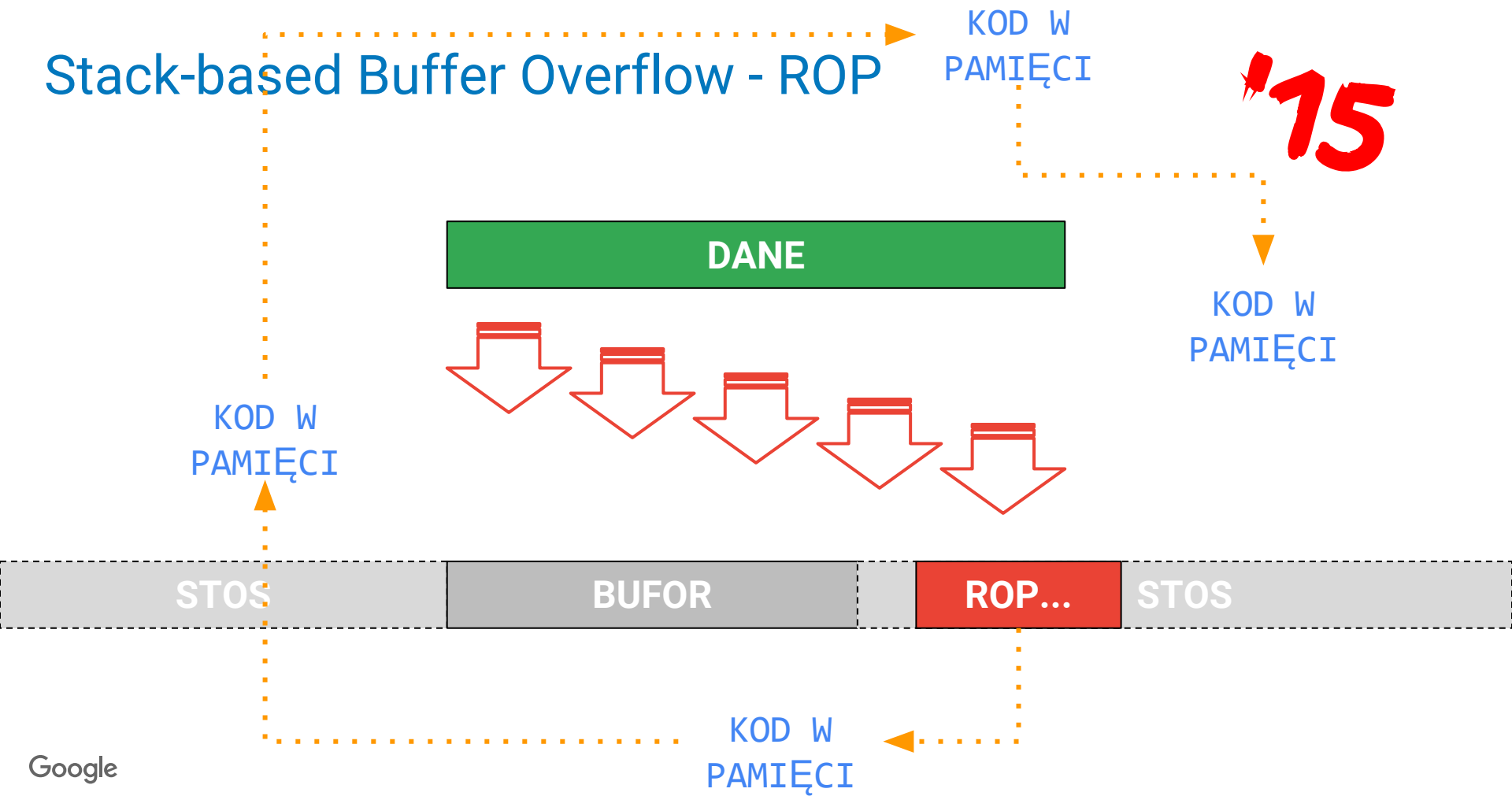


Stack-based Buffer Overflow

90



Stack-based Buffer Overflow - ROP



Return Oriented Programming (ret2libc)

Pomysł:

- **W pamięci jest dużo ciekawego kodu** gotowego do użytku.
- Może da się z niego jakoś **złożyć** coś ciekawego?
- Hint: Kontrolujemy stos.

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**

```
0x415824 pop rax  
0x415825 ret
```

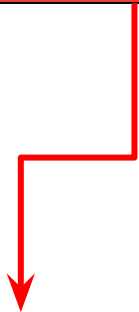
```
0x4504f5 pop rdx  
0x4504f6 ret
```

```
0x418557 mov [rdx], rax  
0x41855a ret
```

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**

RET 1



```
0x415824 pop rax
0x415825 ret
```

```
0x4504f5 pop rdx
0x4504f6 ret
```

```
0x418557 mov [rdx], rax
0x41855a ret
```

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**



0x415824 pop rax
0x415825 ret

0x4504f5 pop rdx
0x4504f6 ret

0x418557 mov [rdx], rax
0x41855a ret

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**



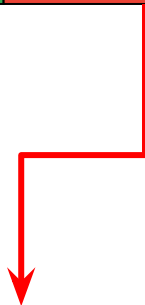
```
0x415824 pop rax  
0x415825 ret
```

```
0x4504f5 pop rdx  
0x4504f6 ret
```

```
0x418557 mov [rdx], rax  
0x41855a ret
```


Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**



```
0x415824 pop rax
0x415825 ret
```

```
0x4504f5 pop rdx
0x4504f6 ret
```

```
0x418557 mov [rdx], rax
0x41855a ret
```

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**



```
0x415824 pop rax  
0x415825 ret
```

```
0x4504f5 pop rdx  
0x4504f6 ret
```

```
0x418557 mov [rdx], rax  
0x41855a ret
```

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**



```
0x415824 pop rax
0x415825 ret
```

```
0x4504f5 pop rdx
0x4504f6 ret
```

```
0x418557 mov [rdx], rax
0x41855a ret
```

Return Oriented Programming (ret2libc)

Przykład: **Chcemy zapisać X do pamięci pod adres Y**



```
0x415824 pop rax
0x415825 ret
```

```
0x4504f5 pop rdx
0x4504f6 ret
```

```
0x418557 mov [rdx], rax
0x41855a ret
```

Return Oriented Programming (ret2libc)

Przykład: Chcemy zapisać X do pamięci pod adres Y



```
0x415824 pop rax
0x415825 ret
```

```
0x4504f5 pop rdx
0x4504f6 ret
```

```
0x418557 mov [rdx], rax
0x41855a ret
```

Return Oriented Programming (ret2libc)

W praktyce, np. wywołanie `execve("/bin/bash", ["/bin/bash", 0], 0)`.

59	sys_execve	const char *filename	const char *const argv[]	const char *const envp[]
RAX		RDI	RSI	RDX

http://blog.rchapman.org/posts/Linux_System_Call_Table_for_x86_64/

Buffer Overflow

DEMO

RPC, serializacja

Serializacja i problemy z nią

***JSON**

Brak obsługi obiektów.

Serializacja i problemy z nią

PHP/unserialize

"Statyczne" tworzenie obiektów.

Wywołanie `__wakeup()`

Wywołanie `__destruct()`

*exploitacja
podobna do ROP*

Serializacja i problemy z nią

Python/pickle

Wywołanie konstruktora
wybranej klasy
z wybranego modułu
z wybranymi parametrami.

np. `subprocess.Popen`

Warning: The `pickle` module is not intended to be secure against erroneous or maliciously constructed data. Never unpickle data received from an untrusted or unauthenticated source.

Serializacja i problemy z nią

DEMO

Podsumowanie

Diabeł tkwi w szczegółach

Punkt widzenia programisty na aplikacje

VS

To co aplikacja naprawdę robi

Hacker mindset

Patrz na aplikacje z punktu widzenia hakera

Znaj klasy błędów typowe dla swojej dziedziny

Koniec

Q&A

