Search for Vulnerabilities Using Static Code Analysis

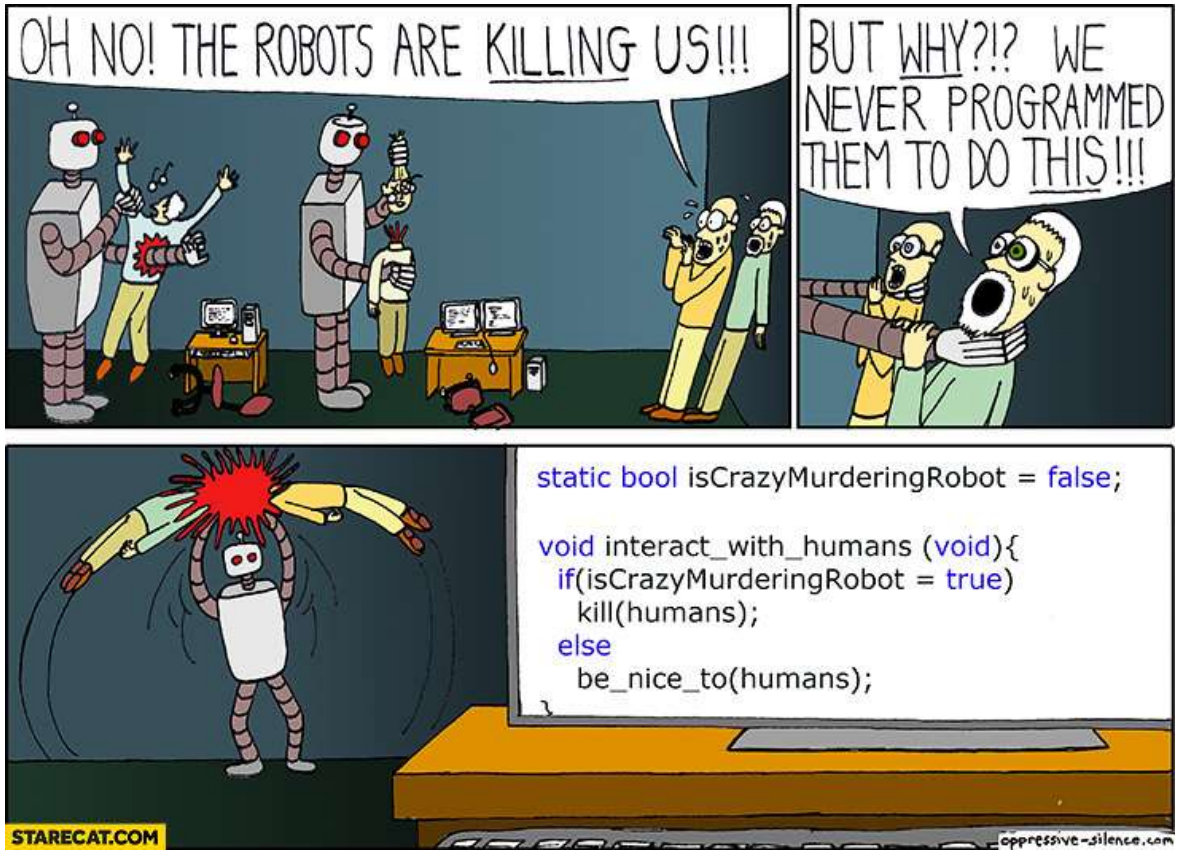
Andrey Karpov
Karpov@viva64.com

Evgeniy Ryzhkov
evg@viva64.com

www.viva64.com

Do We Actually Need It?

- Vulnerabilities are the same things as common errors.
- Why do we distinguish them?
- Do this, if you want to earn more money.



Analogy

- Servers
- Farms
- Clusters
- Data storages
- Boring stuff

Clouds

- Fashionable
- Youthful
- Prestigious

- **You get paid more**

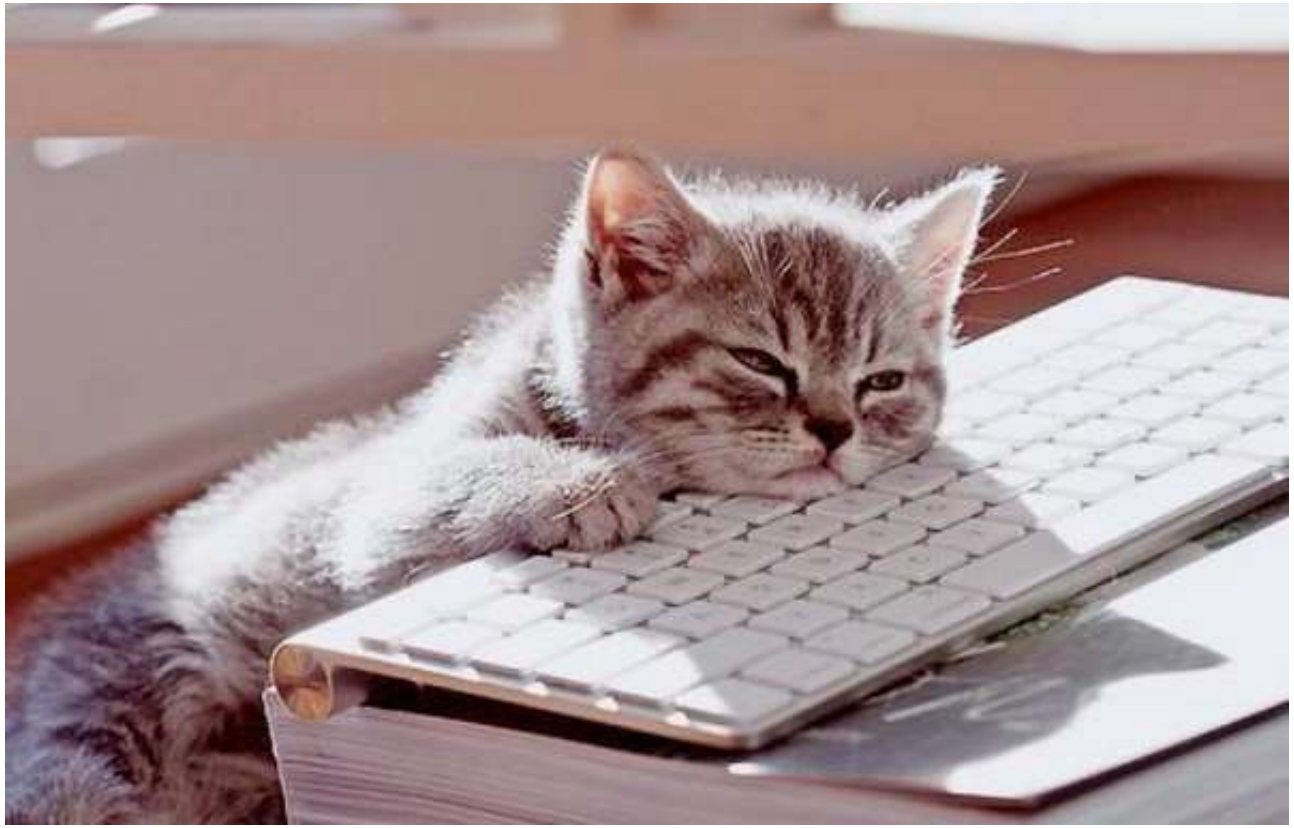
A Banter or Not a Banter, That Is the Question

- Now it is better to be, for example, not an admin, but a DevOps-specialist.
- It is even better to be SecDevOps!

Let’s Talk About How to Become a ~~Programmer~~ More Valuable Security Expert.

We Cannot Do Without a Boring Terminology

- CWE - Common Weakness Enumeration
- CVE - Common Vulnerabilities and Exposures
- Relationship



Sweetheart Bug

```
const char *err = strchr(cp, ':')+2;
tor_assert(err);
```

Tor project

- A check that checks nothing.
- PVS-Studio: V769 The 'strchr(cp, ':')' pointer in the 'strchr(cp, ':') + 2' expression could be nullptr. In such case, resulting value will be senseless and it should not be used. dns.c 163

Vulnerability

Illumos-gate project

```
char *ptr;
....
ptr = strchr(ptr + 1, '/') + 1;
rw_exit(&sdvp->sdev_contents);
sdev_iter_datasets(dvp, ZFS_IOC_DATASET_LIST_NEXT, ptr);
```

- **CVE-2014-9491**: The devzvol_readdir function in illumos does not check the return value of a strchr call, which allows remote attackers to cause a denial of service (NULL pointer dereference and panic) via unspecified vectors.
- PVS-Studio: V769 The 'strchr(ptr + 1, '/')' pointer in the 'strchr(ptr + 1, '/') + 1' expression could be nullptr. In such case, resulting value will be senseless and it should not be used.

Error (a quite suitable CVE candidate)

```
char buffer[1001];
int len;
while ((len = pBIO_read(bio, buffer, 1000)) > 0)
{
  buffer[len] = 0;
  fprintf(file, buffer);
}
```

WinSCP project

- PVS-Studio: V618 It's dangerous to call the 'fprintf' function in such a manner, as the line being passed could contain format specification. The example of the safe code: printf("%s", str); asyncsslsocketlayer.cpp 2247

Vulnerability

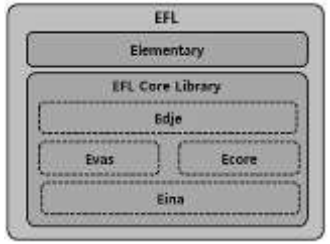
```
if (NasConfig.DoDaemon) {    /* daemons use syslog */
  openlog("nas", LOG_PID, LOG_DAEMON);
  syslog(LOG_DEBUG, buf);
  closelog();
} else {
  errfd = stderr;
```

Network Audio System project

- **CVE-2013-4258**: Format string vulnerability in the osLogMsg function in server/os/aulog.c in Network Audio System (NAS) 1.9.3 allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via format string specifiers in unspecified vectors, related to syslog.
- PVS-Studio: V618 It's dangerous to call the 'syslog' function in such a manner, as the line being passed could contain format specification. The example of the safe code: printf("%s", str);

This is Not a Bug, It's a Feature

```
char *str = malloc(vlen + dlen + 1);
memcpy(str, val, vlen);
memcpy(str + vlen, _dexts[i], dlen);
```



EFL Core Libraries project

- PVS-Studio: V575 The potential null pointer is passed into 'memcpy' function. Inspect the first argument. main.c 112
- In the Core EFL Libraries I've found about **700** of such "features".
- According to author Carsten Haitzler, this is normal. These are not real bugs.

Vulnerability

```
vl->data = calloc(vl->size, sizeof(WORD));
temp_word = SwapWord((BYTE*)d, sizeof(WORD));
memcpy(vl->data, &temp_word, vl->size);
```

Yerase's TNEF Stream Reader project

- **CVE-2017-6298**: An issue was discovered in ytnef before 1.9.1. This is related to a patch described as "1 of 9. Null Pointer Deref / calloc return value not checked."
- PVS-Studio: V575 The potential null pointer is passed into 'memcpy' function. Inspect the first argument.

Not a Bug, But a Nonsense

```
u8 other = memcmp(requester->frame_rcvd.iaf.sas_addr,
                  iphy->frame_rcvd.iaf.sas_addr,
                  sizeof(requester->frame_rcvd.iaf.sas_addr));

if (other == 0) {
   ....
}
```

Linux Kernel

- PVS-Studio: V642 Saving the 'memcmp' function result inside the 'unsigned char' type variable is inappropriate. The significant bits could be lost breaking the program's logic. host.c 1789

Vulnerability

```
typedef char my_bool;
my_bool check_scramble(....)
{
  ....
  return memcmp(hash_stage2, hash_stage2_reassured, SHA1_HASH_SIZE);
}
```

MySQL project

- **CVE-2012-2122**: sql/password.c in Oracle MySQL 5.1.x before 5.1.63, ... , when running in certain environments with certain implementations of the memcmp function, allows remote attackers to bypass authentication by repeatedly authenticating with the same incorrect password, which eventually causes a token comparison to succeed due to an improperly-checked return value.
- PVS-Studio: V642 Saving the 'memcmp' function result inside the 'char' type variable is inappropriate. The significant bits could be lost breaking the program's logic. password.c

Error

```
} else {
  goto no_match; //ATC
  cc = iselCondCode( env, guard );
}
```

проект Valgrind

- V779 Unreachable code detected. It is possible that an error is present. host_arm_isel.c 461

Vulnerability

```
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
  goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
  goto fail;
  goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
  goto fail;
```

iOS project

- CVE-2014-1266: The SSLVerifySignedServerKeyExchange function in libsecurity_ssl/lib/sslKeyExchange.c in the Secure Transport feature in the Data Security component in Apple iOS 6.x ..... does not check the signature in a TLS Server Key Exchange message, which allows man-in-the-middle attackers to spoof SSL servers by (1) using an arbitrary private key for the signing step or (2) omitting the signing step.
- V640 The code's operational logic does not correspond with its formatting. The statement is indented to the right, but it is always executed. It is possible that curly brackets are missing.
- V779 Unreachable code detected. It is possible that an error is present

The Way Big Bosses See the World

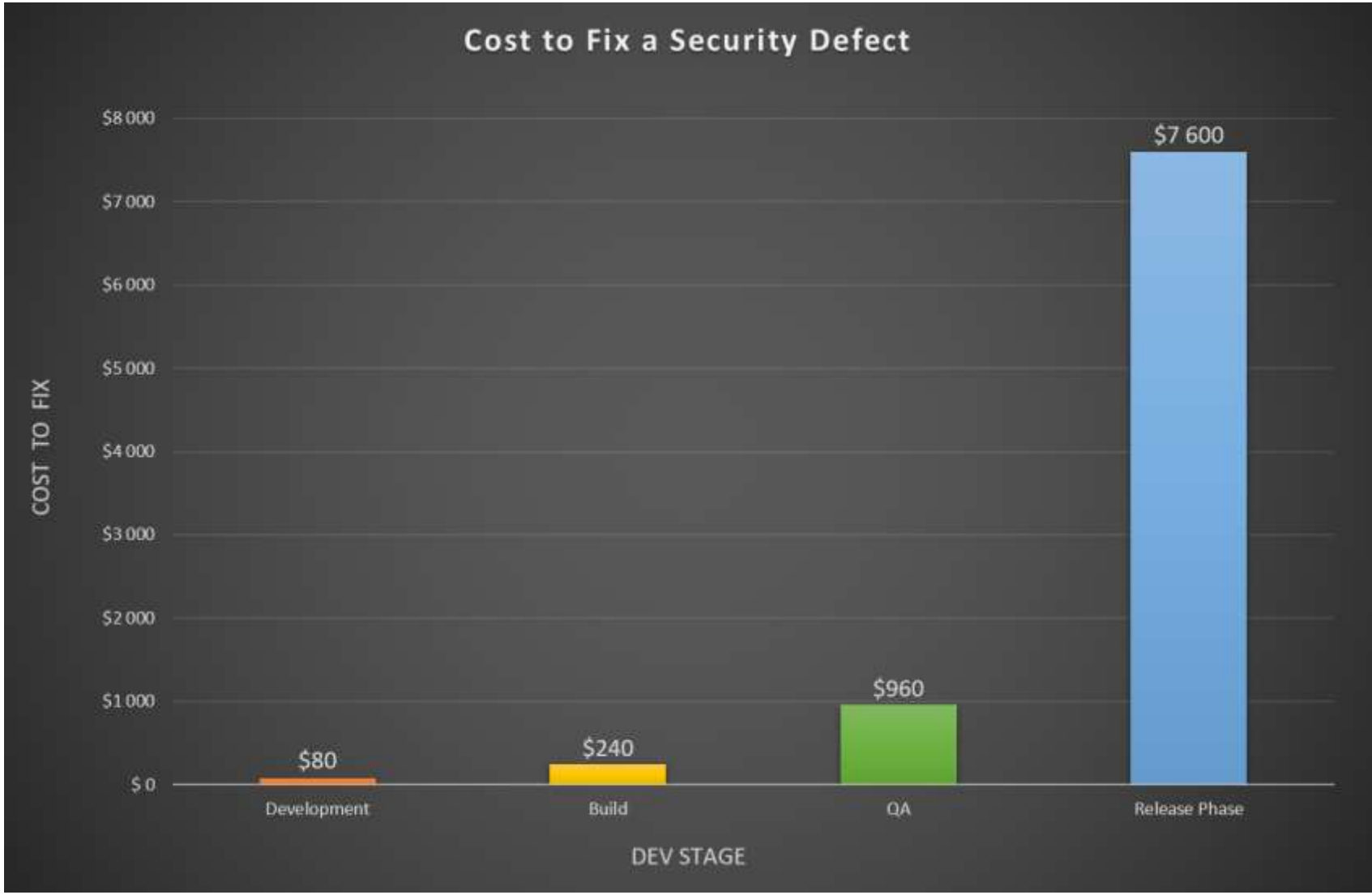
- From the point of view of a programmer, errors are about the same as potential vulnerabilities.
- Although non-programmers perceive them in a different way:

Error

Vulnerability

Panic for Nothing?

- No.
- There is some point in it.



NIST: National Institute of Standards and Technology

What Should You Do?

- Explain that if there are such errors, someone can benefit from this in this way and in that way:
  - Unreliable data sources.
  - Denial of service, etc.

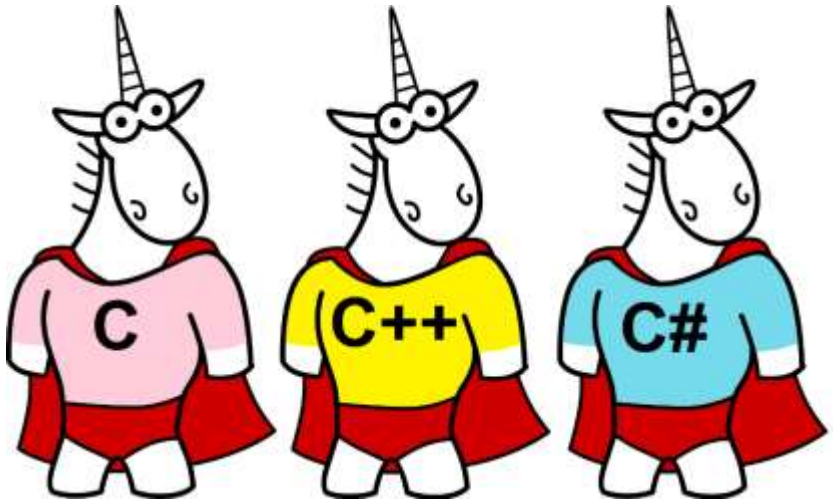
- So, in general, the ways of dealing with errors are the same, but it’s better to use other words and terms.

- Tools:
  - static analyzers
  - dynamic analyzers
  - binary code analyzers

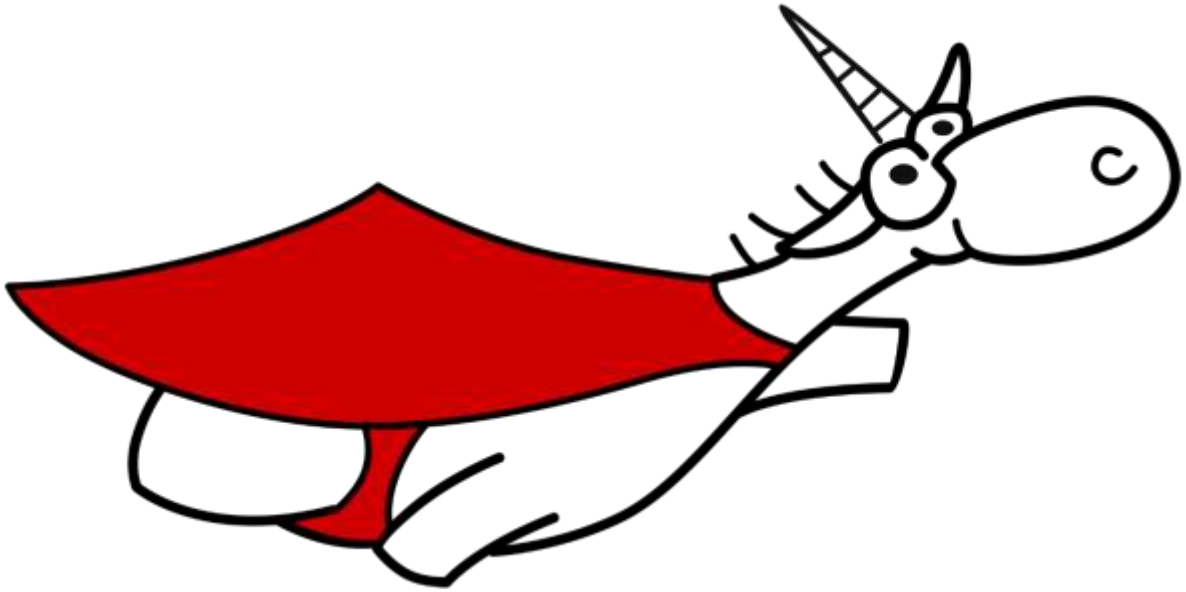
- **Now using Valgrind you're searching not for a memory leak, but for a denial of service!**

Our Own Plans

- We told you how to earn more money;
- What about us?
- We will introduce a classification of errors by CWE in PVS-Studio;
- Finding a couple of vulnerabilities would be perfect for ads;
- Who will find a vulnerability for us?
- We give researchers a free license!



Answers to the Questions



Andrey Karpov karpov@viva64.com

Evgeniy Ryzhkov evg@viva64.com

PVS-Studio web site: https://www.viva64.com

Twitter @Code_Analysis