

# **On the Effectiveness of Configuration Error Injection Testing**

# Background

```
Program received signal SIGSEGV, Segmentation fault.  
my_mb_ctype_8bit(cs=0x1226760, ctype=0x7fffffffde00,s=0x1ad5000  
<Address 0x1ad5000 out of bounds>,e=0x10185a67f  
<Address 0x10185a67f out of bounds>) at ./strings/ctype-simple.c:1299  
1299 *ctype= cs->ctype[*s + 1];
```

⋮  
⋮

This could be **because you hit a bug**. It is also possible that this binary or one of the libraries it was linked against is **corrupt or improperly built**. This error can also be caused by **malfunctioning hardware**.

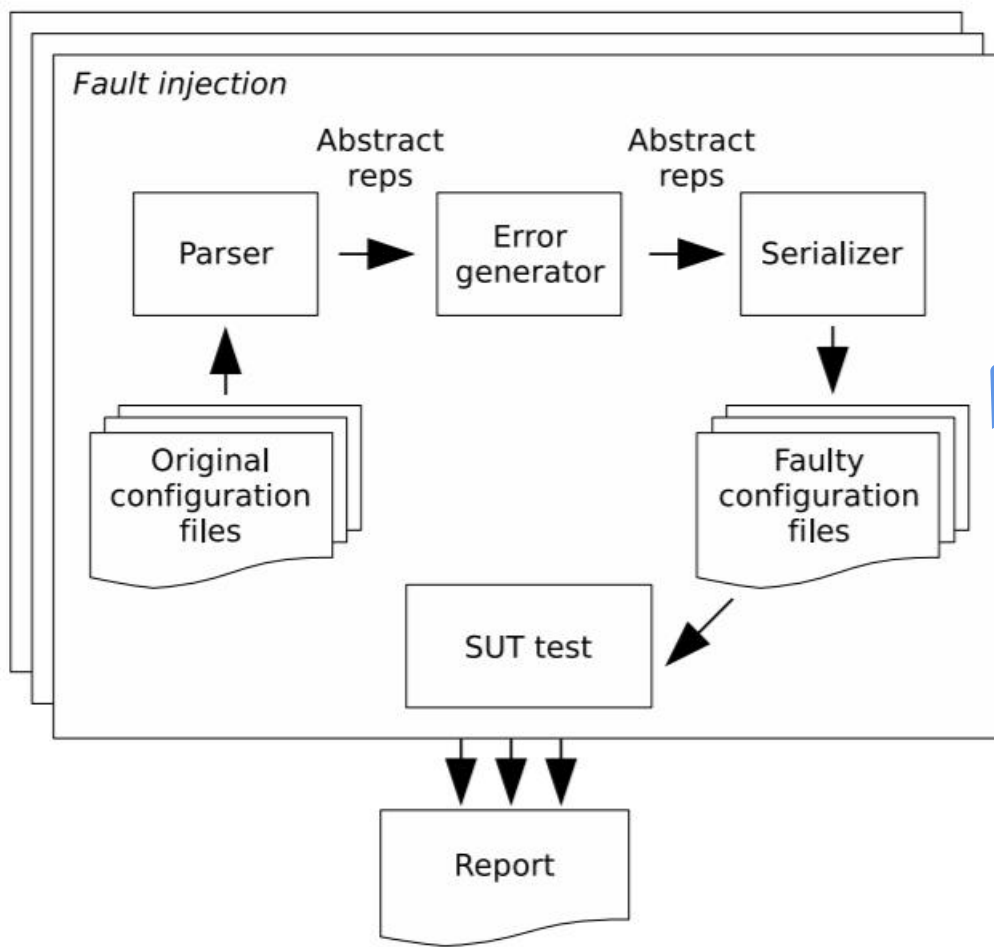
(a) Destructive system reaction (MySQL 5.5)

```
Error reading file 'stopword.dir' (Errcode: 21 - Is a directory)
```

(b) Pinpointing system reaction (MySQL 5.6)

A Real-world Example of Misconfiguration

# Related Work



A Typical Design of CEIT tool (ConfErr)

## How to generate Configuration Errors?

- Random
- Typo: Misspelling, Omission ... (ConfErr)
- Mutation: (ConfDiagDetector)

Configuration Mutation Rule	Example	
	Before	After
Delete the existing value	format = XML	format =
Randomly select values of the same data type from a pre-defined pool	format = XML	format = TXT
Randomly select values of a different data type from a pre-defined pool	format = XML	format = 123
Randomly inject spelling mistakes	format = XML	format = XLL
Change the case of text	format = XML	format = xml

- Specification Violation: (SPEX-INJ, ConfVD)

"ft\_stopword\_file": FILE (MySQL-5.5.29)

### SPEX Injects:

ft\_stopword\_file = a\_directory\_path

### Bad Reaction Exposed:

System crash! (caused by segmentation fault)

(b) Semantic-type Violation (FILE)

# Motivation

- We want to understand the challenges and opportunities of applying configuration error injection testing in real-world software engineering practice.

Categorization	Good Behavior	Useful Logs	All tests passed
Pinpointing-Correction	√	√	√
Pinpointing-Termination	√	√	×
Unstudied	?	×	√
Ungraceful Termination	×	×	×

# Research Questions

- RQ1: How about current CEIT mechanisms' abilities of exposing vulnerabilities?
- RQ2: Are there still unstudied test results?
- RQ3: What we can learn from these RQs?

# Methodology

- Target Systems

<b>Software</b>	<b>Type</b>	<b># Parameters</b>	<b>Dev. history</b>
HTTPD	Web srv.	669 (118)	1995 (24 yrs)
NGINX	Web srv.	551 (277)	2004 (15 yrs)
MySQL	DBMS	461 (114)	1995 (24 yrs)
PostgreSQL	DBMS	275 (232)	1987 (32 yrs)



# Methodology

- Studied CEIT methods:
  - 1. Random
  - 2. Mutation
  - 3. Specification Violation

Configuration Mutation Rule	Example	
	Before	After
Delete the existing value	format = XML	format =
Randomly select values of the same data type from a pre-defined pool	format = XML	format = TXT
Randomly select values of a different data type from a pre-defined pool	format = XML	format = 123
Randomly inject spelling mistakes	format = XML	format = XLL
Change the case of text	format = XML	format = xml

Example of Mutation Rules

	Type	Specification	Generation Rules
Basic	Bool/Enum	Options, value set = {"enum1", "enum2", ...}.	Use a value that doesn't belong to set.
	Numeric	Data type, value set = {Integer, Float, Long, ...}.	(1) A type error; (2) out-of-bound values (e.g., INT_MAX+1); (3) an alphabetic string.
		Valid range, range = [MIN, MAX].	Use the values beyond the valid range.
		Unit of measurements, value set = {"ms" "s", ...}.	Use a non-existent unit (e.g., "nunit").
		Value Relationship ( $P, V, \diamond$ ), $\diamond \in \{<, >, =, \neq, \geq, \leq\}$ .	Use invalid value relationship ( $P, V, \neg\diamond$ )
Semantic	Path	Syntax, $\wedge/(\wedge w+\wedge/?)+\$$	Use a random string (e.g., "f5_B0c:146C").
		Path existence, value set = {Yes, No}.	Use a non-existent file path.
		Path type, value set = {Directory, Regular}.	Use a path to a file that violates the file type, e.g., for directory file, use a regular file.
	URL	Syntax, $[a-z]+://.*$	Use a value that violates the pattern, e.g., http://www.google.com
	IP Address	Syntax, $[\d]{1,3}([\d]{1,3}){3}$	Use a value that violates the pattern, e.g., 255255.255.255
	Port	Valid range, range = [0, 65535].	Use the values beyond the valid range.
		Port should not be occupied.	Use an occupied port number.
	Permission	Data type, value set = {Octet}.	(1) A float-typed number; (2) an alphabetic string.
Valid range, range = [0000, 7777].		Use the values beyond the valid range.	
Name/ID	Specific value of string.	Use an invalid string.	

Example of Specification Violation Rules

# Methodology

- Categorization and Result Analysis
  - Tests (Pass or Fail)
  - Logs (Pinpointing information, e.g., parameter name, configuration location, etc.)

```
Error reading file 'stopword.dir' (Errcode: 21 - Is a directory)
```

(b) Pinpointing system reaction (MySQL 5.6)



# RQ1: Effectiveness and Efficiency

Software	Bad Reactions Exposed	Total Injections	Vulnerabilities Exposed Per 1000 Injections
Random			
HTTPD	0	116	0
NGINX	5(5)	277	18.1
MySQL	1(1)	114	8.8
PostgreSQL	0	232	0
Mutation			
HTTPD	0	580	0
NGINX	5(6)	1385	3.6
MySQL	1(3)	570	1.8
PostgreSQL	2(4)	1160	1.7
Specification Violation			
HTTPD	0	222	0
NGINX	6(8)	772	7.8
MySQL	1(3)	425	2.4
PostgreSQL	3(4)	695	4.3

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

Silent resolutions refer to the undesired behaviors that correct or ignore the erroneous values, without informing users with explicit log messages.

False errors fundamentally break the principle of configuration error injection testing which exercises the system behavior upon erroneous values. The injections are in fact correct values.

Inadequate tests refer to the tests that cannot expose the injected errors.

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Silent Correction

```
1 check_autovacuum_work_mem(int *newval, ...) {
2     if (*newval < 1024)
3         *newval = 1024;
4 }
```

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

Disabled Macro

```
1  #ifndef  OPENSSSL_NO_ECDH
2      nid = OBJ_sn2nid( SSLECDHCurve );
3      ecdh = EC_KEY_new_by_curve_name(nid);
4  #endif
```

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

Control dependency

("fsync", 0, ≠) ↦ "commit\_siblings"

(PostgreSQL-9.2.1)

fsyn = off

commit\_siblings = 5

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Unsafe Parsing

```
static const char *authn_cache_timeout(cmd_parms *cmd, void
*CFG,
{
...
    const char *arg)
    int secs = atoi(arg);
    cfg->timeout = apr_time_from_sec(secs);
...
}
```



# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Arbitrary String

```
static void
emit_greeting(struct vsf_session* p_sess)
{
...
else
{
    vsf_cmdio_write(p_sess, FTP_GREET, tunable_ftp_banner);
}
}
```

**Description:** parameter “tunable\_ftp\_banner” can accept any string value without constraints checking since this parameter is only used in the display of welcome page.

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Case Alteration

```
StoreController::init()
{
...
if (0 == strcasecmp(Config.store_dir_select_algorithm, "round-
robin")) { ...}
...
}
```

**Description:** Parameter “Config.store\_dir\_select\_algorithm” is case-insensitive because of api “strcasecmp”. Therefore all the case changing operation to this parameter will be seen as false error.

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				String	False Errors		Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing		Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Value Selection

Configuration Mutation Rule	Example	
	Before	After
Delete the existing value	format = XML	format =
Randomly select values of the same data type from a pre-defined pool	format = XML	format = TXT
Randomly select values of a different data type from a pre-defined pool	format = XML	format = 123
Randomly inject spelling mistakes	format = XML	format = XLL
Change the case of text	format = XML	format = xml

# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Missing Triggering Conditions

```
1 squid_signal( SIGSEGV , death , SA_NODEFER );
2
3 void death(int sig){
4     if (Config.adminEmail){
5         snprintf(command, "%s %s < %s", Config.EmailProgram ,
6             Config.adminEmail, filename);
7         system(command);
8     }
```

- Specific workload
- Failure Events
- Specific Environments



# RQ2: Unstudied Results

Software (SUT)	Silent Resolutions				False Errors			Inadequate Tests		Total
	Correction	Macro	Dep.	Unsafe Parsing	String	Case alt.	Value sel.	Oracles	Conditions	
Random	6 (3.5%)	9 (5.3%)	8 (4.7%)	41 (24.0%)	52 (30.4%)	0	0	18(10.5%)	37 (21.6%)	171
Mutation	21 (1.3%)	38 (2.3%)	30 (1.8%)	157 (9.7%)	185 (11.4%)	542 (33.3%)	424 (26.1%)	72 (4.4%)	157 (9.7%)	1626
Specification Violation	18 (5.4%)	26 (7.8%)	0	188 (56.6%)	0	0	0	21 (6.3%)	79 (23.8%)	332

## Missing Oracles

```
37 <script>var NS=' zh:services:courier' ;var JSINFO={"ga":{"trackingId":"UA-144673995-1","anonymizeIp":true,"action":'  
{"collapsibleSections":0,"fixedTopNavbar":1,"showSemanticPopup":1,"sidebarOnNavbar":0,"tagsOnTop":1,"tocAffix":1,'  
38 <script charset="utf-8" src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.min.js" defer></script>  
39 <script charset="utf-8" src="https://cdn.jsdelivr.net/npm/jquery-ui-dist@1.12.1/jquery-ui.min.js" defer></script>  
40 <script charset="utf-8" src="/assets/bootstrap3/453098ca4b71fce58604cb33b89d70d7.js" defer></script>  
41 <script type="text/x-mathjax-config">/*! [CDATA[*MathJax.Hub.Config({  
42   tex2jax: {  
43     inlineMath: [ ["$", "$"], ["\\(", "\\)"] ],  
44     displayMath: [ ["$$", "$$"], ["\\[", "\\]"] ],  
45     processEscapes: true  
46   }  
47 })
```

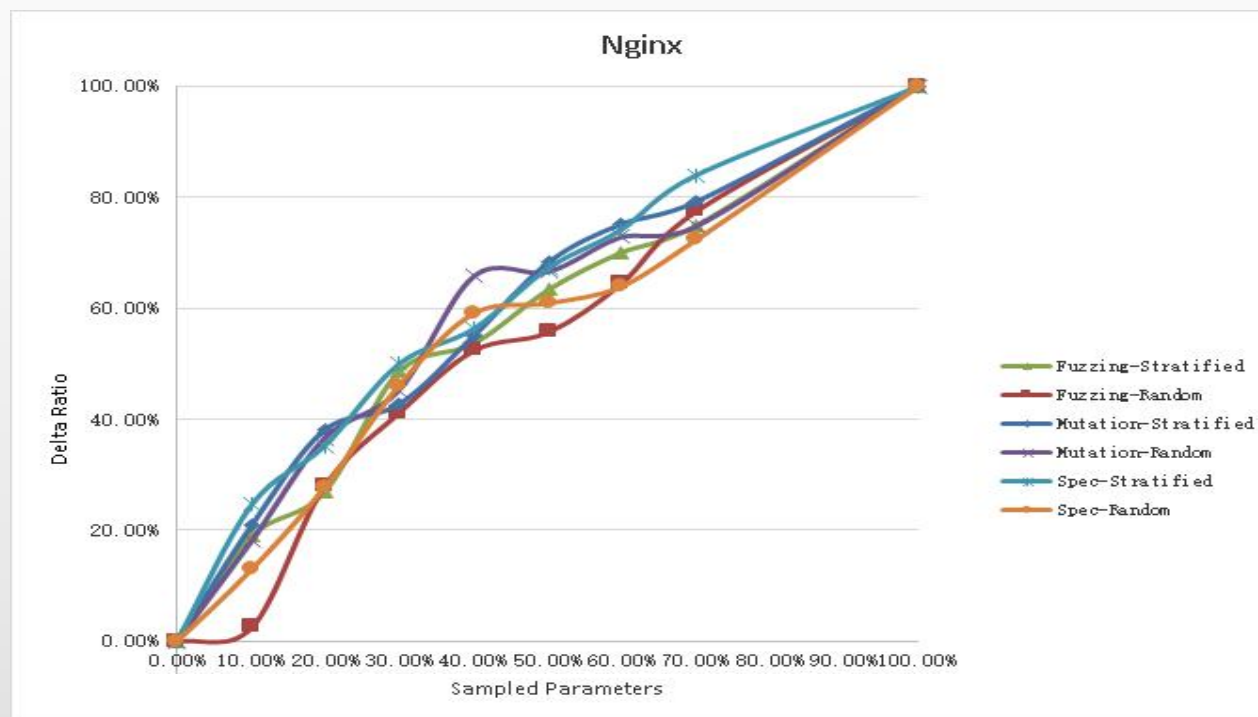
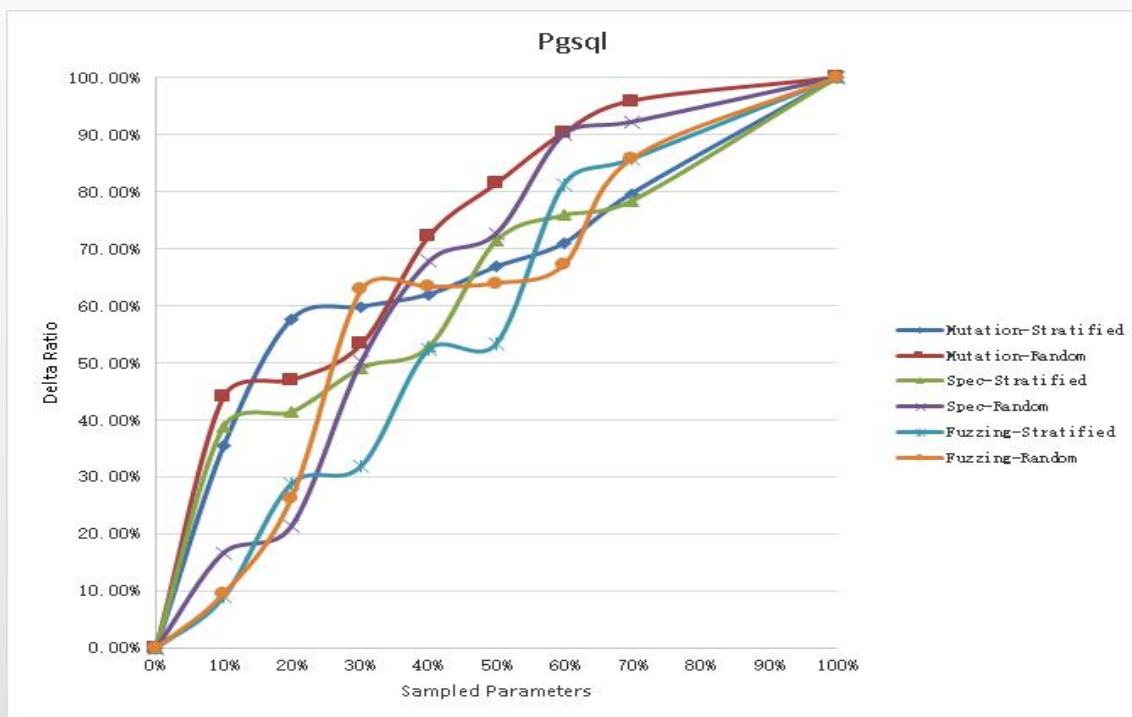
# RQ3: Remedies

- How to improve efficiency?
  - Sampling Parameters
  - Reducing Redundant Tests
  - Smarter CEIT methods
- How to reduce ineffectiveness?
  - Software Design
  - Less False Errors
  - Adequate Tests



# RQ3: Remedies

- How to improve efficiency?
  - Sampling Parameters

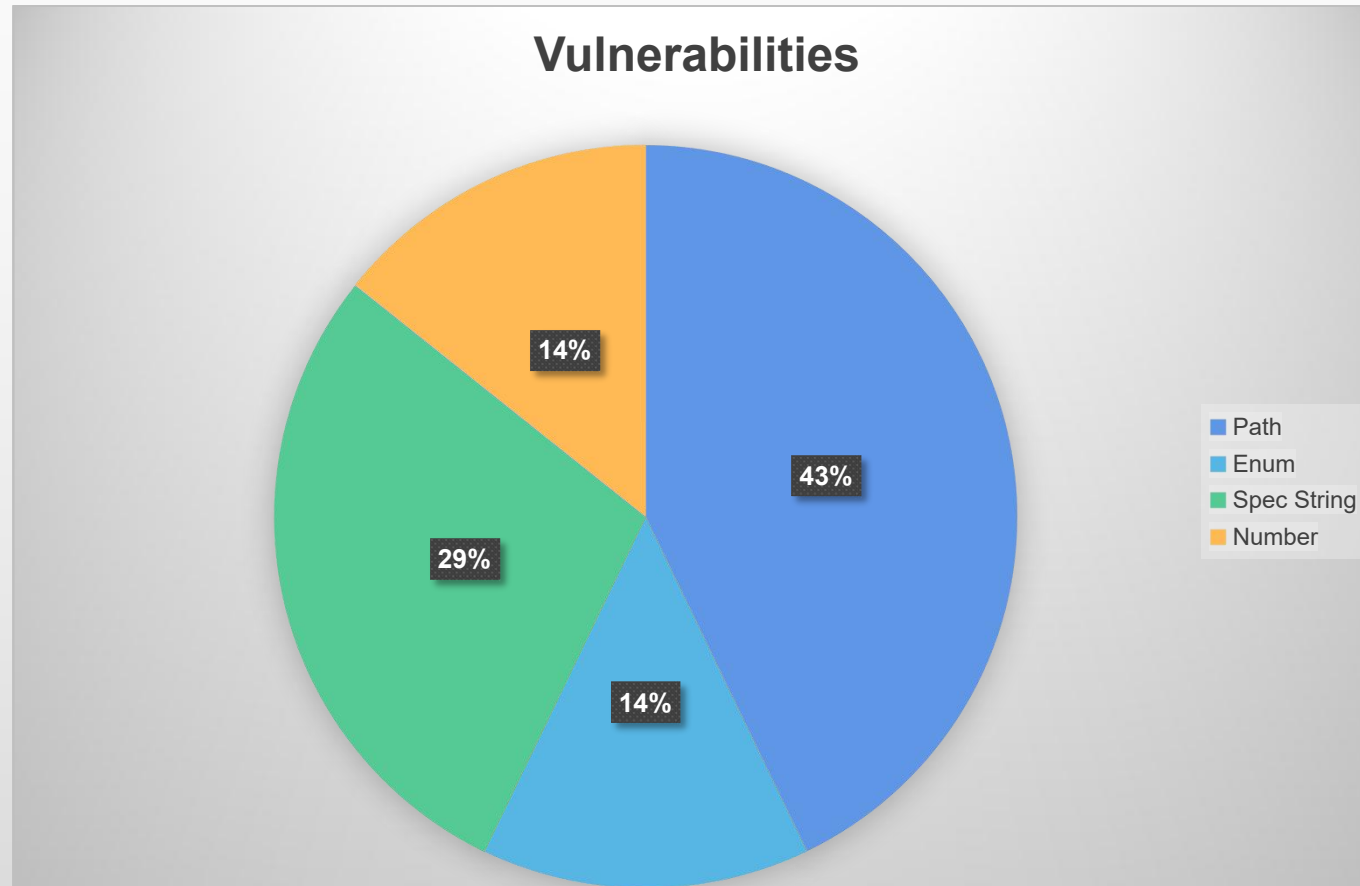


# RQ3: Remedies

- How to improve efficiency?
  - Reducing Redundant Tests
- 81% Tests in functional test suites are redundant for CEIT
- XX% Tests have no relevance with target parameter

# RQ3: Remedies

- How to improve efficiency?
  - Smarter CEIT methods

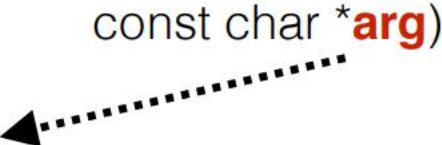


# RQ3: Remedies

- How to reduce ineffectiveness?
  - Software Design

```
1 check_autovacuum_work_mem(int *newval, ...){
2     if (*newval < 1024)
3         *newval = 1024;
4 }
```

```
static const char *authn_cache_timeout(cmd_parms *cmd, void
*CFG,
{
...
    const char *arg)
    int secs = atoi(arg);
    cfg->timeout = apr_time_from_sec(secs);
...
}
```



# RQ3: Remedies

- How to reduce ineffectiveness?
  - Less False Errors

Configuration Mutation Rule	Example	
	Before	After
Delete the existing value	format = XML	format =
Randomly select values of the same data type from a pre-defined pool	format = XML	format = TXT
Randomly select values of a different data type from a pre-defined pool	format = XML	format = 123
Randomly inject spelling mistakes	format = XML	format = XLL
Change the case of text	format = XML	format = xml

# RQ3: Remedies

- How to reduce ineffectiveness?
  - Adequate Tests (Future Work)



# **Towards Effective Test Cases Generation for Configuration Error Injection Testing**

# Challenges

- Where to test
- How to test

# Motivating Examples

User Request



Session



```
parseconf_uint_array[] =
{
  { "accept_timeout", &tunable_accept_timeout },
  { "connect_timeout", &tunable_connect_timeout },
  { "local_umask", &tunable_local_umask },
  { "anon_umask", &tunable_anon_umask },
  { "ftp_data_port", &tunable_ftp_data_port },
  { "idle_session_timeout", &tunable_idle_session_timeout },
  { "data_connection_timeout", &tunable_data_connection_timeout },
  { "pasv_min_port", &tunable_pasv_min_port },
  { "pasv_max_port", &tunable_pasv_max_port },
  { "anon_max_rate", &tunable_anon_max_rate },
  { "local_max_rate", &tunable_local_max_rate },
  { "listen_port", &tunable_listen_port },
  { "max_clients", &tunable_max_clients },
  { "file_open_mode", &tunable_file_open_mode },
  { "max_per_ip", &tunable_max_per_ip },
  { "trans_chunk_size", &tunable_trans_chunk_size },
  { "delay_failed_login", &tunable_delay_failed_login },
  { "delay_successful_login", &tunable_delay_successful_login },
  { "max_login_fails", &tunable_max_login_fails },
  { "chown_upload_mode", &tunable_chown_upload_mode },
  { 0, 0 }
};
```

```
void
process_post_login(struct vsf_session* p_sess)
{
  str_getcwd(&p_sess->home_str);
  if (p_sess->is_anonymous)
  {
    vsf_sysutil_set_umask(tunable_anon_umask);
    p_sess->bw_rate_max = tunable_anon_max_rate;
  }
  else
  {
    vsf_sysutil_set_umask(tunable_local_umask);
    p_sess->bw_rate_max = tunable_local_max_rate;
  }
  if (p_sess->is_http)
  {
    handle_http(p_sess);
    bug("should not be reached");
  }
}
```

p\_sess->is\_anonymous = False

# Motivating Examples

```
{ ngx_string("mp4_buffer_size"),
  NGX_HTTP_MAIN_CONF|NGX_HTTP_SRV_CONF|NGX_HTTP_LOC_CONF
  ngx_conf_set_size_slot,
  NGX_HTTP_LOC_CONF_OFFSET,
  offsetof(ngx_http_mp4_conf_t, buffer_size),
  NULL },
```

```
static ngx_int_t
ngx_http_mp4_process(ngx_http_mp4_file_t *mp4)
{
  ...
  ngx_http_mp4_conf_t *conf;
  ...
  mp4->buffer_size = conf->buffer_size;
  ...
}
```

```
static ngx_int_t
ngx_http_mp4_handler(ngx_http_request_t *r)
{
```

```
static ngx_int_t
ngx_http_mp4_read(ngx_http_mp4_file_t *mp4, size_t size)
{
  ...

  if (mp4->offset + (off_t) mp4->buffer_size > mp4->end) {
    mp4->buffer_size = (size_t) (mp4->end - mp4->offset);
  }

  if (mp4->buffer_size < size) {
    ngx_log_error(NGX_LOG_ERR, mp4->file.log, 0,
                  "%s" mp4 file truncated", mp4->file.name.data);
    return NGX_ERROR;
  }
}
```

# Motivating Examples

```
ngx_int_t
ngx_http_core_content_phase(ngx_http_request_t *r,
    ngx_http_phase_handler_t *ph)
{
    ...
    if (r->content_handler) {
        r->write_event_handler = ngx_http_request_empty_handler;
        ngx_http_finalize_request(r, r->content_handler(r));
        return NGX_OK;
    }
}
```

```
void
ngx_http_update_location_config(ngx_http_request_t *r)
{
    ngx_http_core_loc_conf_t *clcf;
    #define ngx_http_get_module_loc_conf(r, module) (r)->loc_conf[module.ctx_index]
    clcf = ngx_http_get_module_loc_conf(r, ngx_http_core_module);
    ...
    if (clcf->handler) {
        r->content_handler = clcf->handler;
    }
}
```

```
static ngx_int_t
ngx_http_mp4_handler(ngx_http_request_t *r)
{
    ...
    static ngx_int_t
    ngx_http_mp4_read(ngx_http_mp4_file_t *mp4, size_t size)
    {
        ...
        if (mp4->offset + (off_t) mp4->buffer_size > mp4->end) {
            mp4->buffer_size = (size_t) (mp4->end - mp4->offset);
        }
        if (mp4->buffer_size < size) {
            ngx_log_error(NGX_LOG_ERR, mp4->file.log, 0,
                "\"%s\" mp4 file truncated", mp4->file.name.data);
            return NGX_ERROR;
        }
    }
}
```

Function Pointer

# Possible Solutions

- Dynamic Taint Analysis & Static Program Analysis
  - Configuration File -> Variables -> Sentences -> Function Calls
- Fuzzing&Symbolic Execution
  - make sure the target sentences are covered
- Repairing the test cases



# Motivating Examples

Configuration File

## 1. Taint analysis ...

```
parseconf_uint_array[] =
{
  { "accept_timeout", &tunable_accept_timeout },
  { "connect_timeout", &tunable_connect_timeout },
  { "local_umask", &tunable_local_umask },
  { "anon_umask", &tunable_anon_umask },
  { "ftp_data_port", &tunable_ftp_data_port },
  { "idle_session_timeout", &tunable_idle_session_timeout },
  { "data_connection_timeout", &tunable_data_connection_timeout },
  { "pasv_min_port", &tunable_pasv_min_port },
  { "pasv_max_port", &tunable_pasv_max_port },
  { "anon_max_rate", &tunable_anon_max_rate },
  { "local_max_rate", &tunable_local_max_rate },
  { "listen_port", &tunable_listen_port },
  { "max_clients", &tunable_max_clients },
  { "file_open_mode", &tunable_file_open_mode },
  { "max_per_ip", &tunable_max_per_ip },
  { "trans_chunk_size", &tunable_trans_chunk_size },
  { "delay_failed_login", &tunable_delay_failed_login },
  { "delay_successful_login", &tunable_delay_successful_login },
  { "max_login_fails", &tunable_max_login_fails },
  { "chown_upload_mode", &tunable_chown_upload_mode },
  { 0, 0 }
};
```

```
void
process_post_login(struct vsf_session* p_sess)
{
  str_getcwd(&p_sess->home_str);
  if (p_sess->is_anonymous)
  {
    vsf_sysutil_set_umask(tunable_anon_umask);
    p_sess->bw_rate_max = tunable_anon_max_rate;
  }
  else
  {
    vsf_sysutil_set_umask(tunable_local_umask);
    p_sess->bw_rate_max = tunable_local_max_rate;
  }
  if (p_sess->is_http)
  {
    handle_http(p_sess);
    bug("should not be reached");
  }
}
```

# Motivating Examples

## 2. Fuzzing & Symbolic Execution

User Request



Session



```
parseconf_uint_array[] =
{
  { "accept_timeout", &tunable_accept_timeout },
  { "connect_timeout", &tunable_connect_timeout },
  { "local_umask", &tunable_local_umask },
  { "anon_umask", &tunable_anon_umask },
  { "ftp_data_port", &tunable_ftp_data_port },
  { "idle_session_timeout", &tunable_idle_session_timeout },
  { "data_connection_timeout", &tunable_data_connection_timeout },
  { "pasv_min_port", &tunable_pasv_min_port },
  { "pasv_max_port", &tunable_pasv_max_port },
  { "anon_max_rate", &tunable_anon_max_rate },
  { "local_max_rate", &tunable_local_max_rate },
  { "listen_port", &tunable_listen_port },
  { "max_clients", &tunable_max_clients },
  { "file_open_mode", &tunable_file_open_mode },
  { "max_per_ip", &tunable_max_per_ip },
  { "trans_chunk_size", &tunable_trans_chunk_size },
  { "delay_failed_login", &tunable_delay_failed_login },
  { "delay_successful_login", &tunable_delay_successful_login },
  { "max_login_fails", &tunable_max_login_fails },
  { "chown_upload_mode", &tunable_chown_upload_mode },
  { 0, 0 }
};
```

```
void
process_post_login(struct vsf_session* p_sess)
{
  str_getcwd(&p_sess->home_str);
  if (p_sess->is_anonymous)
  {
    vsf_sysutil_set_umask(tunable_anon_umask);
    p_sess->bw_rate_max = tunable_anon_max_rate;
  }
  else
  {
    vsf_sysutil_set_umask(tunable_local_umask);
    p_sess->bw_rate_max = tunable_local_max_rate;
  }
  if (p_sess->is_http)
  {
    handle_http(p_sess);
    bug("should not be reached");
  }
}
```

p\_sess->is\_anonymous = False

p\_sess->is\_anonymous = True



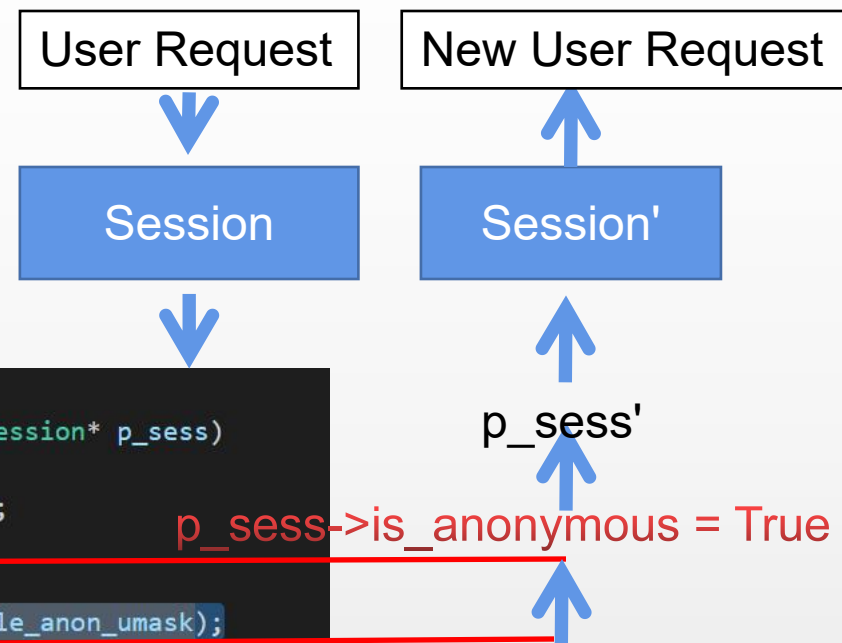


# Motivating Examples

## 3. Repairing

```
parseconf_uint_array[] =
{
  { "accept_timeout", &tunable_accept_timeout },
  { "connect_timeout", &tunable_connect_timeout },
  { "local_umask", &tunable_local_umask },
  { "anon_umask", &tunable_anon_umask },
  { "ftp_data_port", &tunable_ftp_data_port },
  { "idle_session_timeout", &tunable_idle_session_timeout },
  { "data_connection_timeout", &tunable_data_connection_timeout },
  { "pasv_min_port", &tunable_pasv_min_port },
  { "pasv_max_port", &tunable_pasv_max_port },
  { "anon_max_rate", &tunable_anon_max_rate },
  { "local_max_rate", &tunable_local_max_rate },
  { "listen_port", &tunable_listen_port },
  { "max_clients", &tunable_max_clients },
  { "file_open_mode", &tunable_file_open_mode },
  { "max_per_ip", &tunable_max_per_ip },
  { "trans_chunk_size", &tunable_trans_chunk_size },
  { "delay_failed_login", &tunable_delay_failed_login },
  { "delay_successful_login", &tunable_delay_successful_login },
  { "max_login_fails", &tunable_max_login_fails },
  { "chown_upload_mode", &tunable_chown_upload_mode },
  { 0, 0 }
};
```

```
void
process_post_login(struct vsf_session* p_sess)
{
  str_getcwd(&p_sess->home_str);
  if (p_sess->is_anonymous)
  {
    vsf_sysutil_set_umask(tunable_anon_umask);
    p_sess->bw_rate_max = tunable_anon_max_rate;
  }
  else
  {
    vsf_sysutil_set_umask(tunable_local_umask);
    p_sess->bw_rate_max = tunable_local_max_rate;
  }
  if (p_sess->is_http)
  {
    handle_http(p_sess);
    bug("should not be reached");
  }
}
```



p\_sess->is\_anonymous = True