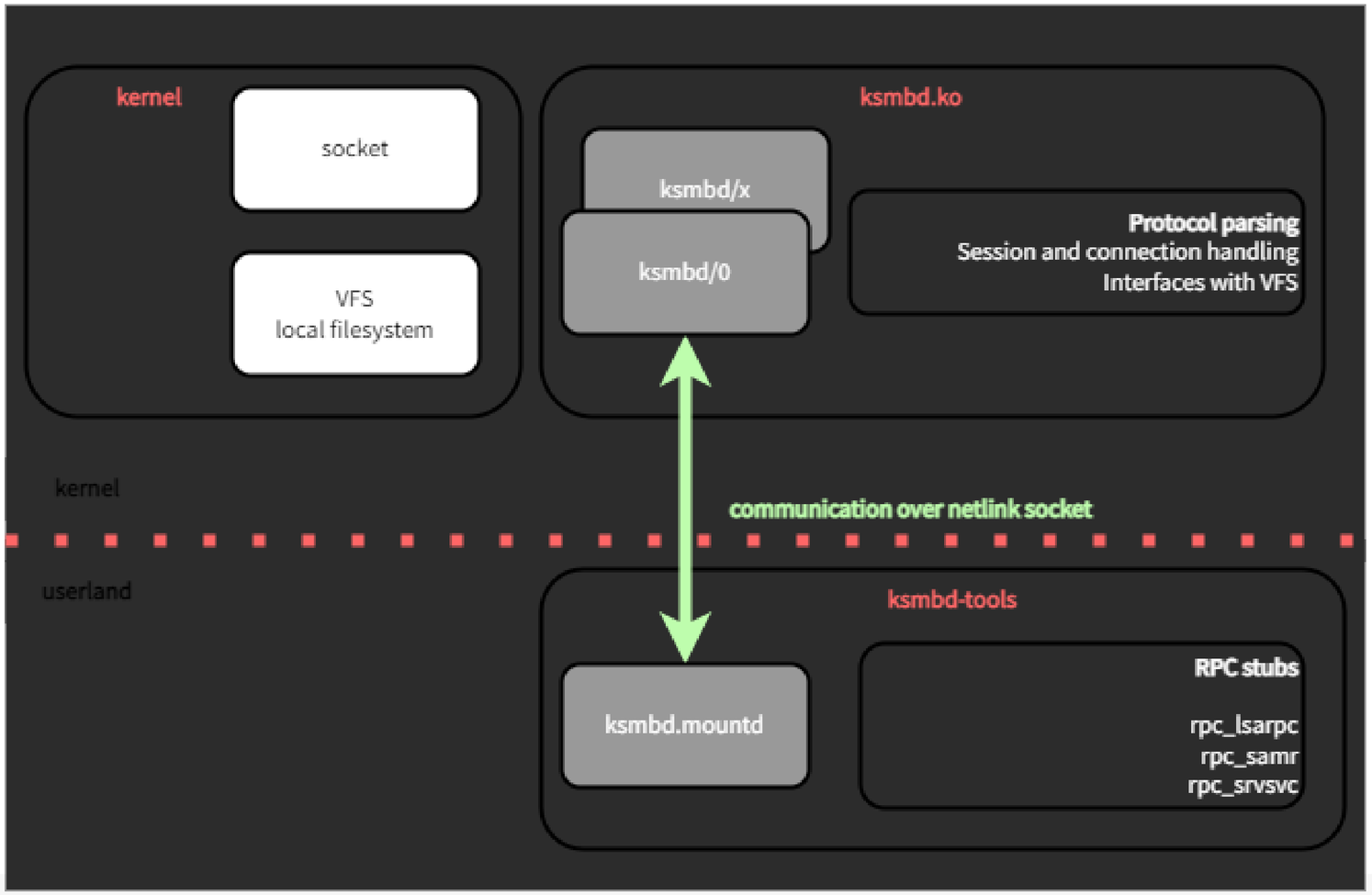
251218

ksmbd 1-day 분석과 syzkaller 실습

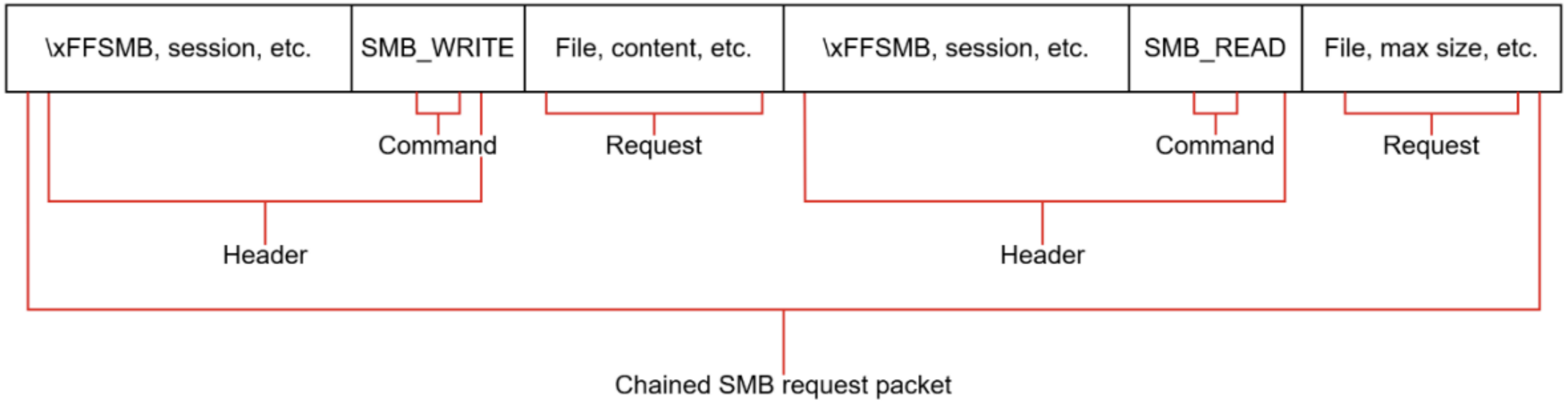
CONTENTS

KSMBD란?

리눅스 커널 내에서 SMB3 프로토콜을 구현-> 네트워크 파일 공유 서버



SMB3 프로토콜

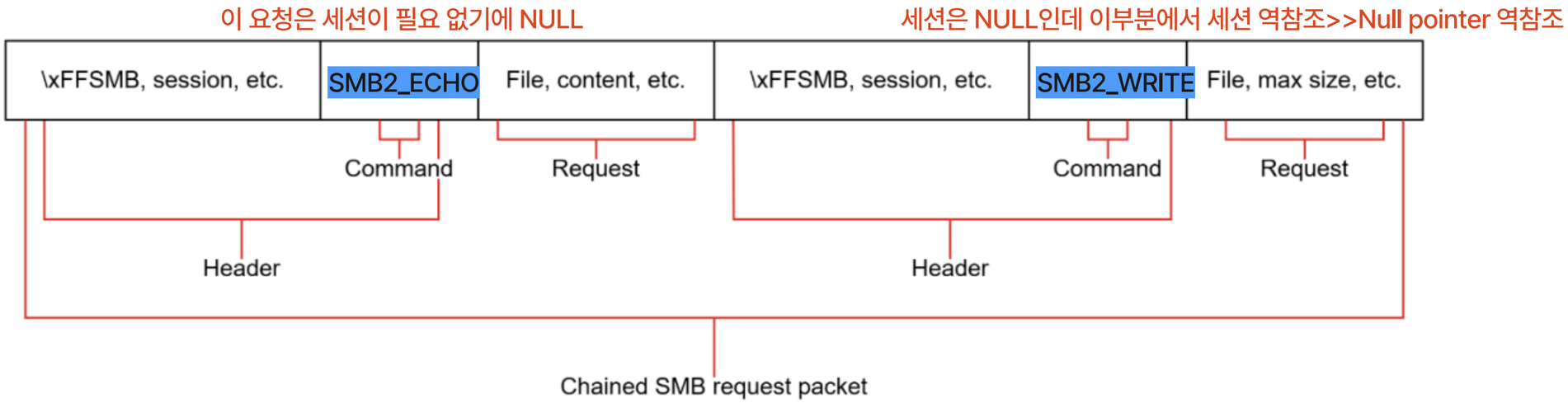

\xFFSMB, session, etc.
SMB_WRITE
File, content, etc.
\xFFSMB, session, etc.
SMB_READ
File, max size, etc.
Command
Request
Command
Request
Header
Header
Chained SMB request packet

ZDI-23-979

CVE ID	CVE-2023-3866
CVSS	5.9, AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H
AFFECTED VENDORS	Linux
AFFECTED PRODUCTS	Kernel

SMB request를 연속적으로 처리하는 세션 과정에서 발생

- static void __handle_ksmbd_work에서는 세션 검사 한 번만 진행
- command가 여러개 있는 chained packet에 대해서도 검사는 한 번만



ZDI-23-979

CVE ID	CVE-2023-3866
CVSS	5.9, AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H
AFFECTED VENDORS	Linux
AFFECTED PRODUCTS	Kernel

SMB request를 연속적으로 처리하는 세션 과정에서 발생

- static void __handle_ksmbd_work에서는 세션 검사 한 번만 진행
- command가 여러개 있는 chained packet에 대해서도 검사는 한 번만

ZDI-23-979

```
if (conn->ops->check_user_session) {
        rc = conn->ops->check_user_session(work);

        // if rc != 0 goto send (auth failed)
        if (rc < 0) {
                command = conn->ops->get_cmd_val(work);
                conn->ops->set_rsp_status(work,
                                STATUS_USER_SESSION_DELETED);
                goto send;
        } else if (rc > 0) {
                rc = conn->ops->get_ksmbd_tcon(work);
                if (rc < 0) {
                        conn->ops->set_rsp_status(work,
                                STATUS_NETWORK_NAME_DELETED);
                        goto send;
                }
        }
}
```

```
do {
        rc = __process_request(work, conn, &command);
        if (rc == SERVER_HANDLER_ABORT)
                break;

    // [snip] (set SMB credits)
} while (is_chained_smb2_message(work));

if (work->send_no_response)
        return;
```

ZDI-23-980

CVE ID	CVE-2023-3865
CVSS	7.1, AV:N/AC:H/PR:L/UI:N/S:C/C:L/I:N/A:H
AFFECTED VENDORS	Linux
AFFECTED PRODUCTS	Kernel

```
int ksmbd_smb2_check_message(struct ksmbd_work *work)
{
        struct smb2_pdu *pdu = ksmbd_req_buf_next(work);
        struct smb2_hdr *hdr = &pdu->hdr;
        int command;
        __u32 clc_len;   /* calculated length */
        __u32 len = get_rfc1002_len(work->request_buf); //원래는 이렇게 len 설정
        if (le32_to_cpu(hdr->NextCommand) > 0)
                len = le32_to_cpu(hdr->NextCommand); //여기서 overwrite
        //그리고 이에 대한 유효성 검사가 없음
        else if (work->next_smb2_rcv_hdr_off)
                len -= work->next_smb2_rcv_hdr_off;

        // [snip] check flag in header

        if (hdr->StructureSize != SMB2_HEADER_STRUCTURE_SIZE) {
                // [snip] return error
        }
```

```
hdr->StructureSize == 64
pdu->StructureSize2 == smb2_req_struct_sizes[command]   // SMB2_WRITE: 49,
SMB2_ECHO: 4
hdr->NextCommand == pdu->StructureSize2 + hdr->StructureSize   // SMB_ECHO
hdr->NextCommand == hdr->DataOffset + hdr->Length   // SMB_WRITE
```

ZDI-23-980

CVE ID	CVE-2023-3865
CVSS	7.1, AV:N/AC:H/PR:L/UI:N/S:C/C:L/I:N/A:H
AFFECTED VENDORS	Linux
AFFECTED PRODUCTS	Kernel

```
int ksmbd_smb2_check_message(struct ksmbd_work *work)
{
        struct smb2_pdu *pdu = ksmbd_req_buf_next(work);
        struct smb2_hdr *hdr = &pdu->hdr;
        int command;
        __u32 clc_len;   /* calculated length */
        __u32 len = get_rfc1002_len(work->request_buf); //원래는 이렇게 len 설정
        if (le32_to_cpu(hdr->NextCommand) > 0)
                len = le32_to_cpu(hdr->NextCommand); //여기서 overwrite
        //그리고 이에 대한 유효성 검사가 없음
        else if (work->next_smb2_rcv_hdr_off)
                len -= work->next_smb2_rcv_hdr_off;

        // [snip] check flag in header

        if (hdr->StructureSize != SMB2_HEADER_STRUCTURE_SIZE) {
                // [snip] return error
        }
```

```
struct smb2_echo_req {
        struct smb2_hdr hdr;
        __le16 StructureSize;     /* Must be 4 */
        __u16   Reserved;
} __packed;
```

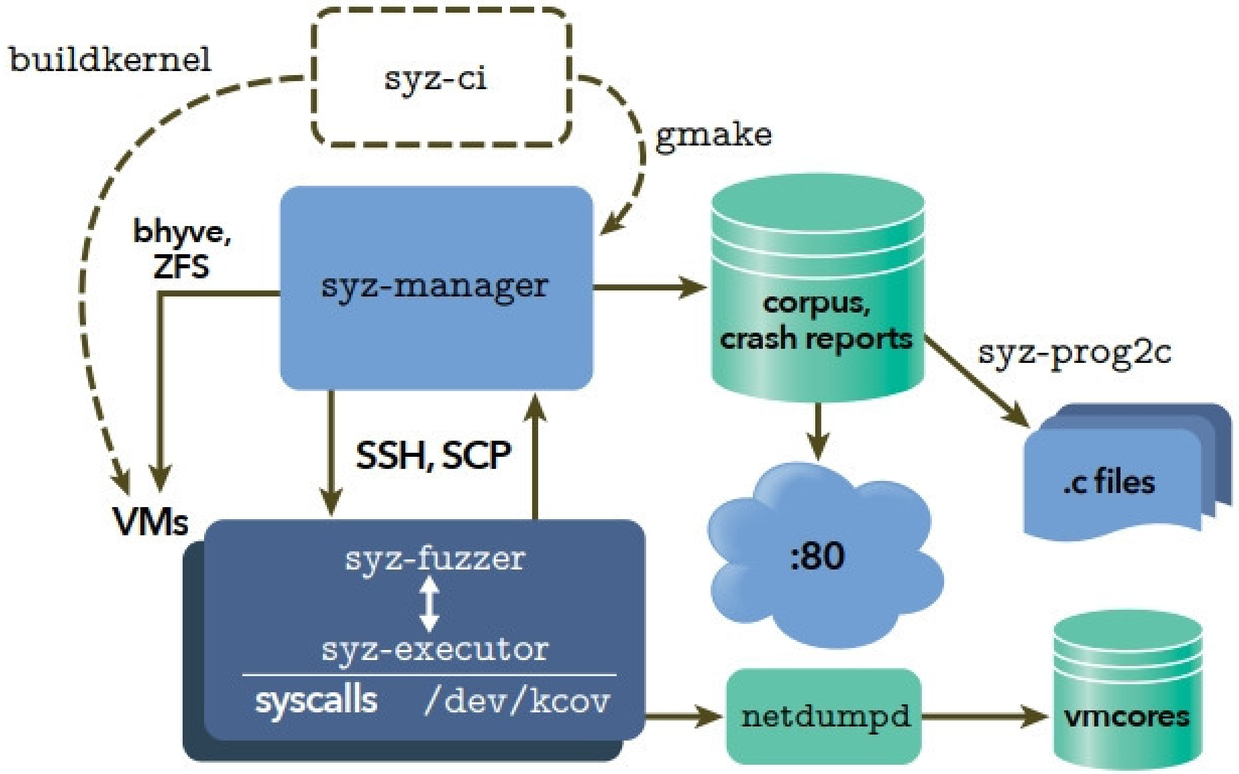
ZDI-23-980

CVE ID	CVE-2023-3865
CVSS	7.1, AV:N/AC:H/PR:L/UI:N/S:C/C:L/I:N/A:H
AFFECTED VENDORS	Linux
AFFECTED PRODUCTS	Kernel

```
int ksmbd_smb2_check_message(struct ksmbd_work *work)
{
        struct smb2_pdu *pdu = ksmbd_req_buf_next(work);
        struct smb2_hdr *hdr = &pdu->hdr;
        int command;
        __u32 clc_len;   /* calculated length */
        __u32 len = get_rfc1002_len(work->request_buf); //원래는 이렇게 len 설정
        if (le32_to_cpu(hdr->NextCommand) > 0)
                len = le32_to_cpu(hdr->NextCommand); //여기서 overwrite
        //그리고 이에 대한 유효성 검사가 없음
        else if (work->next_smb2_rcv_hdr_off)
                len -= work->next_smb2_rcv_hdr_off;

        // [snip] check flag in header

        if (hdr->StructureSize != SMB2_HEADER_STRUCTURE_SIZE) {
                // [snip] return error
        }
```

```
struct smb2_write_req {
        struct smb2_hdr hdr;
        __le16 StructureSize; /* Must be 49 */
        __le16 DataOffset; /* offset from start of SMB2 header to write data */
        __le32 Length;
        __le64 Offset;
        __u64  PersistentFileId; /* opaque endianness */
        __u64  VolatileFileId; /* opaque endianness */
        __le32 Channel; /* MBZ unless SMB3.02 or later */
        __le32 RemainingBytes;
        __le16 WriteChannelInfoOffset;
        __le16 WriteChannelInfoLength;
        __le32 Flags;
        __u8   Buffer[];
} __packed;
```

Syzkaller란?

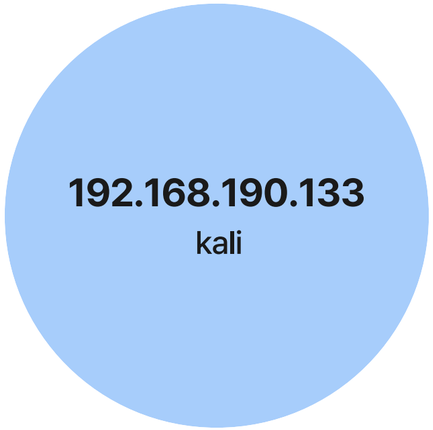


Syzkaller

구글에서 만든 리눅스 커널용 system call fuzzing 도구
Host-Guest 사이 ssh 연결로 원격 퍼징 진행

Syzkaller 실습 구축

syzkaller 실행
원격으로 kali의 ksmbd 퍼징



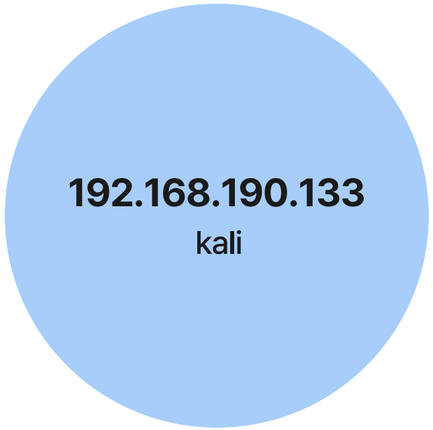
ksmbd 설치하여 빌드

커널 소스 다운 ~ 빌드

sudo apt update
sudo apt install git build-essential flex bison libssl-dev libelf-dev

git clone https://github.com/torvalds/linux.git
cd linux

Syzkaller 실습 구축



syzkaller 실행
원격으로 kali의 ksmbd 퍼징

ksmbd 설치하여 빌드

커널 설정

make defconfig
make kvm_guest.config

./scripts/config --enable CONFIG_SMB_SERVER
./scripts/config --enable CONFIG_SMB_SERVER_SMBDIRECT
./scripts/config --enable CONFIG_SMB_INSECURE_SERVER

./scripts/config --enable CONFIG_KCOV
./scripts/config --enable CONFIG_KCOV_INSTRUMENT_ALL
./scripts/config --enable CONFIG_KASAN
./scripts/config --enable CONFIG_KASAN_INLINE
./scripts/config --enable CONFIG_DEBUG_INFO

03 Syzkaller 실습 구축

192.168.190.131
ubuntu

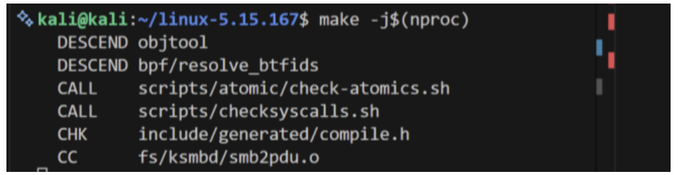
192.168.190.133
kali

syzkaller 실행
원격으로 kali의 ksmbd 퍼징

ksmbd 설치하여 빌드

커널 빌드

make olddefconfig
make -j$(nproc)



Syzkaller 실습 구축

일반 전체

전체 커널의 syscall을 모두 넣어
무작위 난사

ksmbd syscall 관련

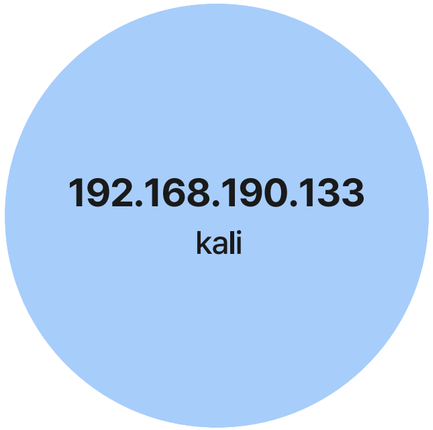
KSMBD 관련 syscall만 넣어서
핵심 기능에 관한 퍼징만 진행

syzlang 사용

smb3 프로토콜 내부 로직을 기반으로
시드 생성하여 퍼징 진행

Syzkaller 실습 구축

syzkaller 실행
원격으로 kali의 ksmbd 퍼징



ksmbd 설치하여 빌드

퍼저 설치

git clone
https://github.com/google/syzkaller.git
cd syzkaller

02 Syzkaller 실습 구축

ksmbd-fuzz > my.cfg

```
{
    "name": "ksmbd-fuzzer",
    "target": "linux/amd64",
    "http": "0.0.0.0:56741",
    "workdir": "/home/ubuubuloadlalala/ksmbd-fuzz/workdir",
    "kernel_obj": "/home/ubuubuloadlalala/ksmbd-fuzz/linux",
    "image": "/home/ubuubuloadlalala/ksmbd-fuzz/dummy.img",
    "sshkey": "/home/ubuubuloadlalala/.ssh/id_rsa",
    "syzkaller": "/home/ubuubuloadlalala/syzkaller",
    "procs": 4,
    "type": "none",
    "rpc": ":34905",
    "reproduce": false,
    "vm": {
        "targets": [ "192.168.190.133:22" ],
        "kernel_dir": "/home/kali/ksmbd-fuzz/linux",
        "user": "kali"
    }
}

```

```
"enable_syscalls": [
    "socket", "socketpair", "bind", "listen", "accept", "accept4",
    "connect", "sendto", "sendmsg", "sendmmsg",
    "recvfrom", "recvmsg", "recvmmsg",
    "shutdown", "close",

    "ioctl", "fcntl",

    "read", "write", "readv", "writev", "pread64", "pwrite64",

    "open", "openat", "creat",
    "stat", "fstat", "lstat",
    "lseek", "truncate", "ftruncate",

    "getxattr", "setxattr", "fgetxattr", "fsetxattr", "listxattr", "flistxattr",

    "epoll_create", "epoll_ctl", "epoll_wait",
    "poll", "ppoll", "select", "pselect6"
],
```

02 Syzkaller 실습 구축

```
ubuubuloadlalala@YOUK-es6:~/syzkaller$ ./bin/syz-manager -config ~/ksmbd-fuzz/my.cfg
2025/12/11 01:24:13 serving rpc on tcp://38175
2025/12/11 01:24:13 no VMs started (type=none)
2025/12/11 01:24:13 you are supposed to start syz-executor manually as:
2025/12/11 01:24:13 syz-executor runner local manager.ip 38175
2025/12/11 01:24:14 skipped 40 seeds
```

```
2025/12/11 01:35:32 corpus                 : 288 (288 seeds), 0 to be reminimized, 0 to be r
esmashed
2025/12/11 01:35:38 candidates=288 corpus=0 coverage=0 exec total=724 (154/min)
2025/12/11 01:35:48 candidates=260 corpus=0 coverage=0 exec total=755 (155/min)
2025/12/11 01:35:58 candidates=45 corpus=1 coverage=3575 exec total=975 (194/min)
2025/12/11 01:36:08 candidates=0 corpus=67 coverage=16864 exec total=1753 (338/min)
2025/12/11 01:36:18 candidates=0 corpus=72 coverage=17224 exec total=1789 (334/min)
2025/12/11 01:36:28 candidates=0 corpus=77 coverage=17251 exec total=1845 (334/min)
2025/12/11 01:36:38 candidates=0 corpus=94 coverage=19042 exec total=2358 (414/min)
2025/12/11 01:36:48 candidates=0 corpus=129 coverage=19497 exec total=2937 (502/min)
2025/12/11 01:36:58 candidates=0 corpus=131 coverage=19498 exec total=2984 (495/min)
2025/12/11 01:37:08 candidates=0 corpus=150 coverage=19604 exec total=3345 (540/min)
2025/12/11 01:37:18 candidates=0 corpus=185 coverage=19797 exec total=4202 (11/sec)
```

```
[ 1322.748207] cgroup: Unknown subsys name 'net'
[ 1322.771653] cgroup: Unknown subsys name 'rlimit'
[ 1322.848273] Adding 124996k swap on ./swap-file.  Priority:0 extent
s:1 across:124996k FS
[ 1323.811163] uffd: Set unprivileged_userfaultfd sysctl knob to 1 if
 kernel faults must be handled without obtaining CAP_SYS_PTRACE capab
ility
[ 1327.519281] mmap: syz.0.512 (6114) uses deprecated remap_file_page
s() syscall. See Documentation/vm/remap_file_pages.rst.
```

02 Syzkaller 실습 구축

```
"disable_syscalls": [
    "keyctl",
    "add_key",
    "request_key",
    "bpf",
    "perf_event_open"
],

"enable_syscalls": [
    "socket$inet_tcp","socket$inet6_tcp", "setsockopt$inet_tcp_TCP_CONGESTION",
    "setsockopt$sock_int","setsockopt$SO_TIMESTAMPING","getsockopt","bind$inet","connect$inet
    "listen","accept","accept4", "sendto", "sendmsg$inet", "sendmmsg$inet", "recvfrom$inet",
    "recvmmsg", "shutdown","close","dup", "dup2","dup3", "ioctl$sock_inet_SIOCSIFFLAGS",
    "ioctl$sock_inet_SIOCGIFCONF", "ioctl$FIONREAD","ioctl$FIONBIO","fcntl$dupfd",
    "fcntl$getflags", "fcntl$setflags",  "fcntl$setstatus",
    "read","write", "readv", "writev", "pread64","pwrite64","preadv","pwritev", "sendfile","
    "openat", "openat2", "creat", "stat", "fstat", "lstat","newfstatat", "lseek",
    "truncate","ftruncate","fsync","fdatasync", "sync_file_range","getxattr", "setxattr",
    "fgetxattr","fsetxattr", "listxattr", "flistxattr","removexattr", "fremovexattr","mkdir",
    "mkdirat","rmdir", "unlink", "unlinkat", "rename", "renameat","renameat2", "link",
    "linkat",
```

03 Syzkaller란?

syzlang 정의

common_linux.h 수정

```
smb2_header {
    protocol_id     array[int8, 4]      # "\xfeSMB"
    header_length   int16be
    credit_charge   int16be
    status          int32be
    command         int16be             # SMB2 commands
    credits         int16be
    flags           int32be
    next_command    int32be
    message_id      int64be
    process_id      int32be
    tree_id         int32be
    session_id      int64be
    signature       array[int8, 16]
}
```

```
smb2_cmd_negotiate = 0x0
smb2_cmd_session_setup = 0x1
smb2_cmd_logoff = 0x2
smb2_cmd_tree_connect = 0x3
smb2_cmd_tree_disconnect = 0x4
smb2_cmd_create = 0x5
smb2_cmd_close = 0x6
smb2_cmd_flush = 0x7
smb2_cmd_read = 0x8
smb2_cmd_write = 0x9
smb2_cmd_lock = 0xa
smb2_cmd_ioctl = 0xb
smb2_cmd_cancel = 0xc
smb2_cmd_echo = 0xd
smb2_cmd_query_directory = 0xe
smb2_cmd_change_notify = 0xf
smb2_cmd_query_info = 0x10
smb2_cmd_set_info = 0x11
smb2_cmd_oplock_break = 0x12
```

03 Syzkaller란?

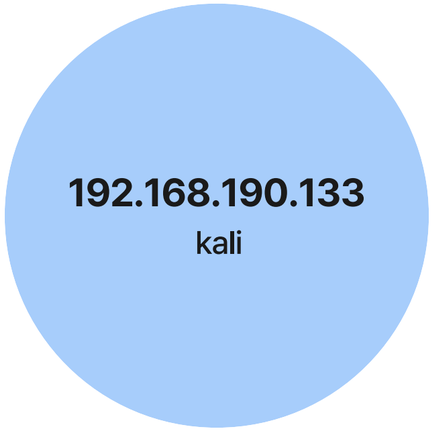
syzlang 정의

common_linux.h 수정

```c
#if SYZ_EXECUTOR_USES_SHMEM
static long syz_ksmbd_send_req(volatile long a0, volatile long a1, volatile long a2)
{
    int sockfd = (int)a0;
    char* buf = (char*)a1;
    size_t len = (size_t)a2;

    pid_t pid = getpid();

    size_t total_len = sizeof(pid) + len;
    char* packet = (char*)malloc(total_len);
    if (!packet)
        return -1;

    memcpy(packet, &pid, sizeof(pid));
    memcpy(packet + sizeof(pid), buf, len);

    struct sockaddr_in addr;
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_port = htons(445);
    inet_pton(AF_INET, "127.0.0.1", &addr.sin_addr);
    if (sockfd < 0) {
        sockfd = socket(AF_INET, SOCK_STREAM, 0);
        if (sockfd < 0) {
            free(packet);
            return -1;
        }

        if (connect(sockfd, (struct sockaddr*)&addr, sizeof(addr)) < 0) {
            close(sockfd);
            free(packet);
            return -1;
        }
    }
    ssize_t sent = send(sockfd, packet, total_len, 0);
    char response[4096];
    ssize_t received = recv(sockfd, response, sizeof(response), 0);

    free(packet);

    return (received > 0) ? received : -1;
}
#endif
```

Syzkaller 실습 구축

syzkaller 실행
원격으로 kali의 ksmbd 퍼징



ksmbd 설치하여 빌드

커널 빌드

cd ~/syzkaller
make generate
make executor
make manager
./bin/syz-manager -config <cfg 파일>

미디어 시장 현황

syzkaller: ksmbd-fuzzer

docs | mailing list | source 48b27acc

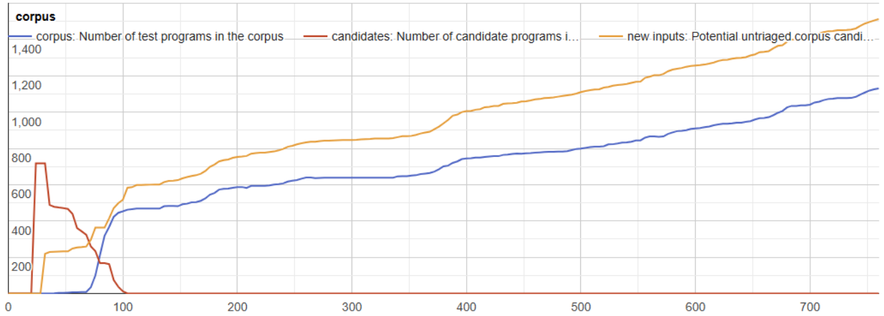
stats | coverage | syscalls | corpus | VMs | config

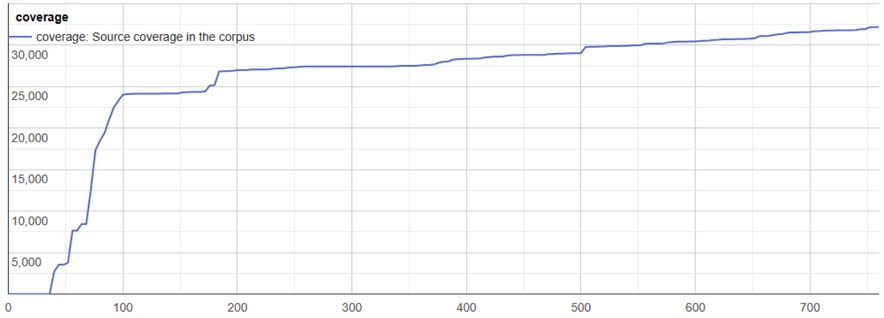
exec total	0 (0/hour)
crash types	0
crashes	0
suppressed	0
uptime	0 sec

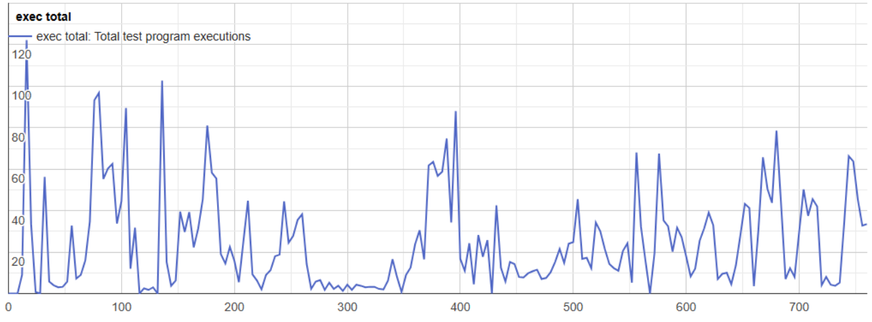
Log:

```
2025/12/11 02:25:32 serving rpc on tcp://34905
2025/12/11 02:25:32 no VMs started (type=none)
2025/12/11 02:25:32 you are supposed to start syz-executor manually as:
2025/12/11 02:25:32 syz-executor runner local manager.ip 34905
2025/12/11 02:25:32 serving http on http://0.0.0.0:56741
2025/12/11 02:25:33 skipped 40 seeds
```

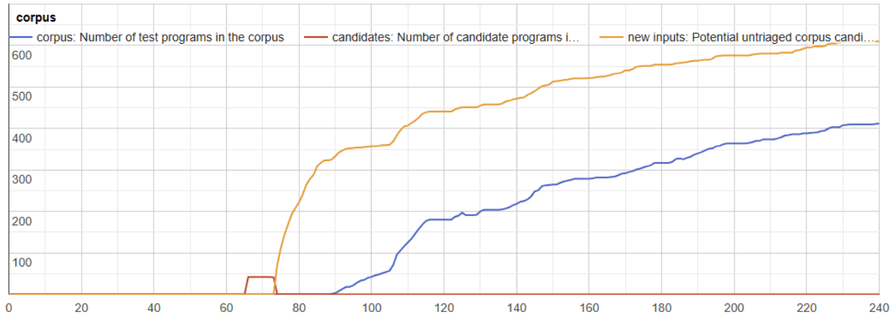
미디어 시장 현황

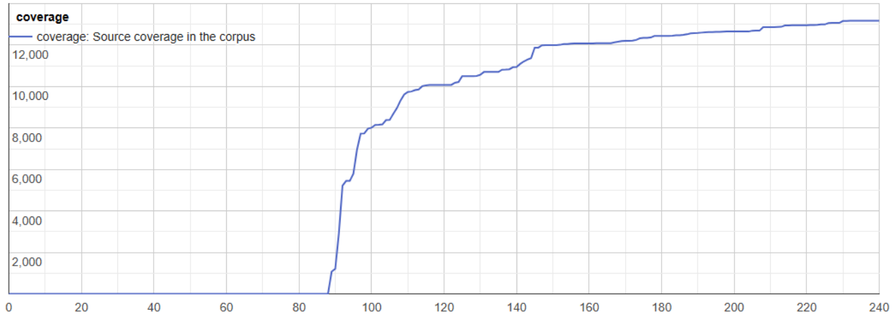


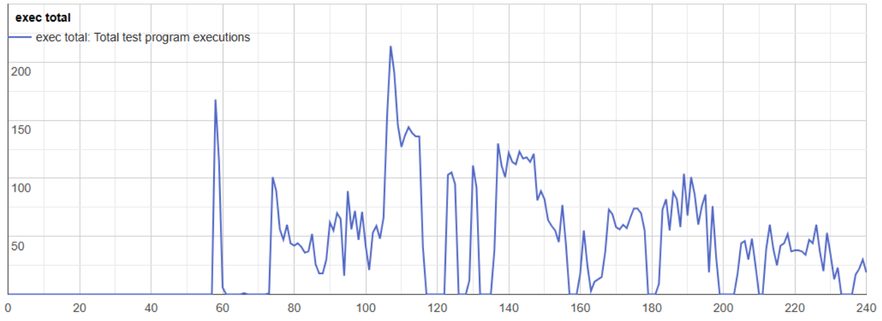




일반 전체







ksmbd syscall 관련