CS 136 Group Project
Security Review of iptables

Junhong Wang

Stewart Dulaney

Brett Woltz

Elyse Yao

Joshua McInerney

1. Summary

We performed code review on `iptables-1.8.7` to look for vulnerabilities. We chose to code review by hand based on static code analysis tools and documents found on the web.

With some static code analysis tools, we found hundreds of usage of easily misused (i.e. expose vulnerabilities) C functions. We identified that more than half of them are properly used. Although we didn't find any vulnerabilities, we did find two weaknesses in the code. First, we found a usage of `fopen` that could allow attackers to override the content of sensitive files, but only if the program was run as root (see 3.3.3 for details). Second, we found an insecure use of `strlen` that could cause an out-of-bounds read that could crash `iptables-translate` (see 3.3.5 for details).

Next, the team did a web search for previous vulnerabilities. Then, the study performed live testing on these previously documented vulnerabilities. The results of this testing showed that all of the previously vulnerabilities mentioned in this report have been patched. One of the vulnerabilities was fixed with a Linux kernel update, while most of the other ones seemed to have been patched through updates to the source code.

Next, we performed threat modeling in order to evaluate the system design security of `iptables`.This involved defining security goals and security policies, and using the attack tree methodology. Our modeling results confirmed that we had not missed any attack surfaces in our investigations. Regarding penetration testing of `iptables`, we didn't have the bandwidth to perform the tests ourselves but we did include our research that should be used in future security evaluations.

Although we didn't have enough time to look into all the potential vulnerabilities, based on the ones we've already investigated, we believe that `iptables-1.8.7` is sufficiently secure.

2. Plan

During our first meeting, we discussed how we were going to approach the problem and find vulnerabilities in `iptables`. We originally planned to try a code review with automated tools, a web search for common or likely vulnerabilities, and code review by hand. We planned to meet regularly on Zoom to discuss our progress and any ideas, then work independently using different methods between meetings.

Progress Schedule:

2/25:	Discuss the project, review the spec, and split work
3/1:	Check in, review progress, and discuss next steps
3/8:	Discuss evaluation results, plan report
3/10:	Polish report and make last edits before submission

We discuss our planned approaches and deviations in the subsections below.

2.1. Code review using automated tools

We used these tools to analyze the code statically and look for potential security problems:

- Sonarcloud
- Flawfinder
- PVS-Studio Analyzer

We looked at the OWASP list of source code analysis tools and started with trying to use Sonarcloud. However, even after building `iptables`, Sonarcloud could not detect the build files and did not work properly, so we tried Flawfinder instead. We used results from Flawfinder for our initial investigation of possible vulnerabilities.

Although we originally planned on using a single tool, we also decided to try a second static code analysis tool, PVS-Studio. Flawfinder output a long list of possible issues, and we tried to see if a different tool would find a smaller set of more obvious issues we could prioritize.

2.2. A web search for known or likely security problems

We planned to research common vulnerabilities in C to see if the same mistakes were being made in the code base. We also wanted to check previously known vulnerabilities of `iptables` and verify if the vulnerabilities were actually resolved or if the fix introduced new vulnerabilities.

2.3. Code review by hand

We originally planned to manually review the code and look for common security vulnerabilities we learned from 2.1. Although we quickly realized that there were too many lines of code to fully review in a reasonable amount of time, we identified and reviewed code sections that we felt were likely to have errors by looking at results from 2.2.

2.4. Live testing

In order to test `iptables`, we used a virtual machine of `Ubuntu 20.04.2 LTS` (see section 3.4.1 for more details on the testing environment). The plan for testing `iptables` was to test all the previous vulnerabilities just to be sure they were patched.

2.5 Threat Modeling

In order to evaluate the system design security of `iptables`, we also wanted to perform threat modeling. The reason we wanted to use threat modeling was to ensure we had not missed any attack surfaces in our static analysis, web search, manual code review, and live testing. First, we planned to define the security goals of the system. Next, we planned to define the security policy requirements of `iptables`. Finally, we planned to conduct threat modeling for `iptables` using the attack tree methodology.

2.6 Penetration Testing

To supplement our live testing, we planned to investigate how to perform penetration tests against `iptables` and execute them against specific configurations. Although our team didn't have the bandwidth to complete this part of the evaluation, our results include our research into methods for how they would be carried out.

3. Results

We executed our plans described in the previous section, and here are the results.

3.1. Code review using automated tools

In this section, we present the information we got from the automated code analysis tools run against the source code of `iptables`.

3.1.1. Sonarcloud

Sonarcloud requires you to actually build the program to analyze C code (Sonarcloud does not support C code static analysis). We built `iptables` following the `INSTALL` file, but Sonarcloud couldn't detect the build files. Thus, we were unable to analyze the code with Sonarcloud.

3.1.2. Flawfinder

Flawfinder worked and produced output as follows:

```
$ flawfinder . > flawfinder_output.txt # current directory is iptables-1.8.7
```

It turns out there are 683 potential vulnerabilities in the code. See 7.1 for the full output. However, as the output says at the end, they could be false positives. So we needed to check each hit by hand to see if they are real vulnerabilities. We were able to identify 435 of them as false positives by statically analyzing the code (e.g. check if the destination buffer size is larger or equal to the source buffer size for `strcpy`). The rest of 248 cases require us to dynamically analyze the code (i.e. need to have a deep understanding of how the code works semantically), which we will discuss in 3.3. See 7.2 for the 248 possible vulnerabilities we narrowed down to.

3.1.3. PVS Studio

We also used PVS Studio to run `pvs-studio-analyzer` and `plog-converter`:

```
pvs-studio-analyzer trace -- make
pvs-studio-analyzer analyze -o ~/documents/iptables.log -l
~/.config/PVS-Studio/PVS-Studio.lic
plog-converter -a GA:1,2 -t tasklist -o ~/documents/iptables2.tasks
~/documents/iptables.log
```

Although it required a license, we were able to use it for free using the student key. We removed results relating to copyleft licenses, which were irrelevant to our security analysis. The full cleaned output of our analysis is included in 7.3.

While PVS found fewer potential vulnerabilities than flawfinder, the results had a fair amount of overlap. Both static code analysis tools found many potential buffer overflow or format string vulnerabilities.

3.1.4. Summary of Potential Vulnerabilities

Between Flawfinder and PVS Studio, these are the categories of potential issues we found:

1. Buffer overflow
    a. `memcpy()`
    b. `sprintf()`
    c. `strlen()`
    d. `strncpy()`
    e. `strcat()`
    f. `getopt()/getopt_long()`
2. Memory Leak:
    a. `realloc()`
3. Pointers
    a. Unsafe use of pointer after comparison to null pointer
    b. Potential null pointer passed to functions
4. Array Overrun
5. Unsafe use of memset()
6. Unsafe use of `printf()`, `vfprintf()`
7. `fopen(), open()` symlink
8. `atoi()`

3.2. A web search for known or likely security problems

In this section, we present what we learned from looking for common and previously known vulnerabilities on Google.

3.2.1. Common security vulnerabilities in C

A quick web search suggests that these are common security vulnerabilities in C:

- `gets`
- `strcpy`
- `sprintf`
- `printf`
- `open`
- `fopen`
- `strcat`
- `strcmp`

3.2.2. Previously known vulnerabilities for iptables

The best thing we can learn from these previously patched vulnerabilities is that our goal shouldn't necessarily be to break the code, or insert malicious code, but to make it work in unintended ways. Details of previously patched vulnerabilities are shown below.

3.2.2.1. CVE-2019-11360

> A buffer overflow in iptables-restore in netfilter iptables 1.8.2 allows an attacker to (at least) crash the program or potentially gain code execution via a specially crafted iptables-save file. This is related to add_param_to_argv in xshared.c.

> One could argue that the severity of the issue is not high, because one already needs root privileges to run iptables-restore in a standard debian setup. So if an attacker is capable of running iptables-restore, then she already can do more harm to your system anyway. If an administrator can be convinced to apply unknown iptables rules files, then her system is most likely already compromised.
>
> Furthermore, I had to disable all well-known security features like NX, Canaries and PIE to build this PoC. This means, that building a real-world exploit probably takes much more effort and/or is quite hard.

Sources: [1] [2]

3.2.2.2. CVE-2012-2663

In the man page for `iptables`, having '--syn' set is supposed to only match tcp packets with the "SYN bit set and the ACK, RST and FIN bits cleared." Usually, these packets are used for TCP initialization, so dropping them would mean refusing an incoming connection without affecting outgoing connections. However the Linux stack considers packets with the SYN bit set and the ACK and RST flags cleared. Thus, packets with both the SYN and FIN bits set would still result in an initialization. This could lead to SYN+ACK flooding resulting in DoS and if there is any rule set in `iptables` with the flags "-p tcp ... --syn", those rules can be bypassed. [1] [2]

As a concrete example, if I have the following rule set up in my firewall to block tcp initialization packets from a particular ip,

"`iptables -A INPUT -s xxx.xxx.xxx.xxx -p tcp -m tcp --dport 22 --syn -j DROP`"

‘-s xxx.xxx.xxx.xxx’ checks the ip of the packet, ‘-p tcp’ indicates the protocol is tcp, ‘-m tcp’ tells `iptables` to match the given information to the packet. ‘--syn’ is the flag that causes the issue.

Then the attacker at the ip address could simply “craft” their packets to have both the SYN and FIN fields set, and my firewall would no longer be able to filter their tcp initialization packets. Essentially ignoring this rule completely.

This was patched in the fix for CVE-2012-6638. [3] Which made the linux kernel drop tcp packets with the SYN+FIN bits set to avoid SYN+FIN floods/Dos. [4]

Sources: [1] [2] [3] [4]

3.2.2.3. CVE-2001-1388

> iptables before 1.2.4 does not accurately convert rate limits that are specified on the command line, which could allow attackers or users to generate more or less traffic than intended by the administrator.

3.2.2.4. CVE-2001-1387

> iptables-save does not save the rules defined correctly. Specifically, so far as I can tell, every instance of
> a rule that contains reject-with icmp-host-prohibited is always changed to --reject-with tcp-reset.
>
> This is true even on rules which do not contain -p tcp.

Sources: [1]

3.2.2.5. FTP Port Exploit

The module which manages connections in the RELATED state deals with FTP DATA connections. Specifically, the PORT command, which manages active FTP DATA connections, does not validate the specified IP/port and adds a RELATED rule set allowing connections between the FTP server and the specified IP/port. Since very few firewalls have strict rules regarding these RELATED connections, a potential attacker could connect to many entities.

An attacker does not need to be authenticated by the FTP server, instead just using the PORT command which requires no validation. From here an attacker can use the FTP server to connect to any port on the same firewall or even another device protected by the firewall. In order for an attacker to exploit this vulnerability, the attacker first needs to be behind the firewall in question, either by compromising an FTP server behind the firewall or a client (which can be used to connect to a remote FTP server to do the same thing).

The actual exploit begins by connecting to the FTP server, and then issuing the PORT command specifying any arbitrary IP address and port, since it won't be validated. This will create an entry and put it into the connection table. This can be dangerous, for example, if the attacker specifies the SSH port. While the FTP server will reject it, there will be a short period of time during which the firewall has the entry before it expires and the attacker can potentially SSH into a device behind the firewall.

3.3. Code review by hand

Based on 3.1 and 3.2, we started to investigate the potential vulnerabilities in detail. For most of the problems, we needed to figure out the values of the variables during the run-time, which requires very thorough knowledge about how the application works. Thus, we researched only a few of the problems due to the time constraint.

3.3.1. getopt/getopt_long

The `getopt` and `getopt_long` functions parses the command line arguments. The function signature is as follows:

```
int getopt(int argc, char * const argv[], const char *optstring)
int getopt_long(int argc, char * const argv[], const char *optstring)
```

According to CVE-1999-0966, `getopt` allows an attacker to gain root access if `argv[0]` is too long. `argv[0]` is the name of an executable, and we don't have any files whose names are too long. The names of the executables are constant, and the attackers can't change them. Thus, we do not need to worry about the use of `getopt` and `getopt_long`.

3.3.2. memset

According to SornerSouce, we should not use memset to clear sensitive data. memset can be optimized out if the compiler thinks the operation is not necessary. For example:

```
int unsafe_memset()
{
    char pwd[64];
    fgets(pwd, 64, stdin);
    ...
    /* use pwd for something here */
    ...
    memset(pwd, 0, 64);
    return 0;
}
```

From the compiler's point of view, `pwd` is not used after `memset`, so it could decide to optimize the code by removing that entire line. If `memset` is optimized away during compilation, then the code above has a huge problem. If another program calls `malloc` and happens to get the memory segment of `pwd`, then it will be able to see the content of `pwd`. To avoid this, we should use `memset_s`, which is guaranteed to be there after compilation. We checked every usage of `memset`, but they aren't clearing any sensitive data. Thus, we do not need to worry about this problem.

3.3.3. open/fopen

open and fopen are susceptible to symbolic link attacks. There are two usages of `fopen` in the code base, which are vulnerable to potential symbolic link attacks (`iptables-save.c` and `xtables-save.c`):

```
case 'f':
    file = fopen(optarg, "w");
    if (file == NULL) {
        fprintf(stderr, "Failed to open file, error: %s\n",
            strerror(errno));
        exit(1);
    }
    ret = dup2(fileno(file), STDOUT_FILENO);
```

Both files contain the exact same pieces of code shown above. For the context, `f` means the file option, which specifies the filename to redirect the program output. According to `fopen` man page, `w` corresponds to `O_WRONLY|O_CREAT|O_TRUNC` flags. Thus,

attackers can pass, for example, `-f=bad_file.txt`, where `bad_file.txt` is a symbolic link to `/etc/passwd`, as an option to perform a symbolic link attack. `fopen` will follow the symlink. If they happen to be able to run this program as root, then they can overwrite the password file. To mitigate this problem, as recommended by Oracle Developer's Guide, we should use open with `O_NOFOLLOW` flag to prevent the program from following the symlink. With the `O_NOFOLLOW` flag, the program will fail to open the file if the file is a symbolic link.

Similarly, there is one usage of `open` in the code base, which is vulnerable to potential symbolic link attacks (`xshared.c`):

```
lock_file = getenv("XTABLES_LOCKFILE");
   if (lock_file == NULL || lock_file[0] == '\0')
       lock_file = XT_LOCK_NAME;
fd = open(lock_file, O_CREAT, 0600);
```

Again, we can mitigate this problem by also passing `O_NOFOLLOW` as a flag. These vulnerabilities are not very serious unless attackers can run the program as root.

3.3.4. printf/vfprintf

In our automated results from PVS-Studio, there were four warnings for a potential format string vulnerability using printf, all in `ip6tables.c`. We found that all four occurrences were duplicates of the same code, shown below.

ip6tables.c `(842, 845, 951, 955)`:

```
    printf(unsupported_rev);
```

Although `unsupported_rev` is a variable, we found that it's actually a constant string and not externally-controlled. Therefore, these cases of `printf` are not vulnerabilities.

While flawfinder had more results, including warnings about usage of both `printf` and `vprintf`, they all seem to be false flags as well. Some use only string literals as the format argument, and others use variables that aren't externally controlled.

3.3.5. strlen

We found a discussion on Hacker News about potential vulnerabilities caused by improper string handling in C. One recommendation was to always use size_t to store string lengths. Researching more on the actual potential consequences of storing the result of `strlen()` in an `int`, we found that it's likely due to the fact that strlen() itself returns an unsigned size_t, which could cause an unsigned to signed conversion error when stored in an `int`. Combined with a large input string or an improperly null-terminated string, this could cause a negative value to be stored instead of the large positive value.

We found two significant incidents of the results of `strlen()` being stored in a signed integer:

xtables-translate.c (37-41)

```
void xlate_ifname(struct xt_xlate *xl, const char *nftmeta, const char
*ifname, invert)
{
   int ifaclen = strlen(ifname), i, j;
   char iface[IFNAMSIZ * 2];
   if (ifaclen < 1 || ifaclen >= IFNAMSIZ)
       return;
```

Even if an interface name is provided such that storing `strlen(ifname)` in a signed integer will cause the integer to incorrectly be treated as a negative length, the conditional directly after it ensures that the function will return without anything being done.

nft-shared.c

```
void add_iniface(struct nftnl_rule *r, char *iface, uint32_t op)
{
   int iface_len;

   iface_len = strlen(iface);

   add_meta(r, NFT_META_IIFNAME);
   if (iface[iface_len - 1] == '+') {
```

Similar code exists in the same file in the function `add_outiface`, as well as in `add_logical_iniface` and `add_logical_outiface` in `nft-bridge.c`.

If a large enough interface is provided such that `strlen(iface)` causes `iface_len` to treat it as a negative signed integer, `iface[iface_len - 1]` could attempt to access a negative index, resulting in an out of bounds read. According to CWE-25, this could result in either a crash or reading other data unintended for access.

Tracing through the code, we found that these functions are used in `iptables-translate` and `iptables-restore-translate`. The former translates a command line from `iptables` to nftables syntax, and the latter translates a ruleset from `iptables` to nftables. Neither of these commands actually modifies any rules, only converting text. As a result, it's unlikely that this can be used for any serious exploit.

To patch this vulnerability, the code should be modified to include a range check on the variables used to store the results of strlen before doing any further operations with them.

3.3.6 strcat/strcpy

Section of nft_arp_print_rule_details in nft-arp.c

```
if (fw->arp.iniface[0] != '\0') {
        strcat(iface, fw->arp.iniface);
        print_iface = 1;
    }
    else if (format & FMT_VIA) {
        print_iface = 1;
        if (format & FMT_NUMERIC) strcat(iface, "*");
        else strcat(iface, "any");
    }
```

Section of for_each_chain6 in xtables.c

```
if (!cs.target && (strlen(cs.jumpto) == 0 || ip6tc_is_chain(cs.jumpto,
*handle))) {
      size_t size;
      cs.target = xtables_find_target(XT_STANDARD_TARGET,
XTF_LOAD_MUST_SUCCEED);
```

```
        size = sizeof(struct xt_entry_target) + cs.target->size;
        cs.target->t = xtables_calloc(1, size);
        cs.target->t->u.target_size = size;
        strcpy(cs.target->t->u.user.name, cs.jumpto);
        xs_init_target(cs.target);
}
```

As is the case with strlen, a large enough incoming ARP interface into the strcat function in the first code segment above could cause a buffer overflow if the iface buffer is not large enough to hold it. This could potentially result in a crash or even stack/heap corruption. strcpy is similar in that cs.jumpto is larger than the `u.user.name` buffer that it's being copied into, a buffer overflow will occur.

Going through the codebase and as seen in these code segments, these functions don't directly modify `iptables` rules making it an unlikely candidate as an exploit for a serious `iptables` compromise. Although these functions don't seem to pose any serious threats to the security of `iptables`, it would be useful to ensure that they can't possibly cause availability or integrity issues. Patching these vulnerabilities could involve using the safe versions of these functions instead: `strcpy_s` and `strcat_s`. If not, then doing range checking on the size of the buffers against the size of the strings being concatenated/copied to the buffer would suffice.

3.4. Live testing

3.4.1. Environment Details

```
ubuntu@ubuntu-VirtualBox:~$ lsb_release -a

No LSB modules are available.

Distributor ID:     Ubuntu

Description:  Ubuntu 20.04.2 LTS

Release:      20.04

Codename:     focal
```

```
ubuntu@ubuntu-VirtualBox:~$ uname -a

Linux ubuntu-VirtualBox 5.8.0-44-generic #50~20.04.1-Ubuntu SMP Wed Feb 10 21:07:30
UTC 2021 x86_64 x86_64 x86_64 GNU/Linux

ubuntu@ubuntu-VirtualBox:~$ iptables --version

iptables v1.8.7 (legacy)

ubuntu@ubuntu-VirtualBox:~$ ifconfig

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500

        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255

        inet6 fe80::8fc2:e9e8:4ef7:891c  prefixlen 64  scopeid 0x20<link>

        ether 08:00:27:88:11:0b  txqueuelen 1000  (Ethernet)

        RX packets 20206  bytes 30109419 (30.1 MB)

        RX errors 0  dropped 0  overruns 0  frame 0

        TX packets 5988  bytes 396471 (396.4 KB)

        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536

        inet 127.0.0.1  netmask 255.0.0.0

        inet6 ::1  prefixlen 128  scopeid 0x10<host>

        loop  txqueuelen 1000  (Local Loopback)

        RX packets 260  bytes 22915 (22.9 KB)

        RX errors 0  dropped 0  overruns 0  frame 0

        TX packets 260  bytes 22915 (22.9 KB)

        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

(See section 6.4 for more details on the system)

3.4.2. Testing previous Vulnerabilities

3.4.2.1 CVE-2012-2663

In this vulnerability, an `iptables` rule with the flag '--syn' used can still be bypassed if the attacker crafts their packet to have both the SYN and FIN bits of their tcp packet set.

```
ubuntu@ubuntu-VirtualBox:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
ubuntu@ubuntu-VirtualBox:~$ sudo hping3 -c 2 -SF -p 31000 localhost
HPING localhost (lo 127.0.0.1): SF set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=31000 flags=RA seq=0 win=0 rtt=7.7 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=31000 flags=RA seq=1 win=0 rtt=4.9 ms

--- localhost hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 4.9/6.3/7.7 ms

#On another terminal,
ubuntu@ubuntu-VirtualBox:~$ nc -l 31000

#Back to the current terminal:

ubuntu@ubuntu-VirtualBox:~$ sudo hping3 -c 2 -SF -p 31000 localhost
HPING localhost (lo 127.0.0.1): SF set, 40 headers + 0 data bytes

--- localhost hping statistic ---
2 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

As we can see, packets crafted to have both the SYN and FIN flags set are automatically dropped as a result of the fix to the Linux kernel from CVE-2012-6638. Thus, we cannot take advantage of the vulnerability described in CVE-2012-2663.

Just for fun, I added a rule involving '--syn' to `iptables` and tested the ping again.

```
ubuntu@ubuntu-VirtualBox:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 3100 --syn
-j DROP
ubuntu@ubuntu-VirtualBox:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 3100 --tcp-flags FIN,SYN,RST,ACK SYN -j DROP

#On another terminal,
ubuntu@ubuntu-VirtualBox:~$ nc -l 31000
```

```
#Back to the current terminal:

ubuntu@ubuntu-VirtualBox:~$ sudo hping3 -c 2 -n -FS -p 80 10.0.2.15
HPING 10.0.2.15 (enp0s3 10.0.2.15): SF set, 40 headers + 0 data bytes

--- 10.0.2.15 hping statistic ---
2 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

The results were the same as not having the rule at all, as is intended.

One thing to note however, is that if your goal is to prevent outgoing tcp connection requests, the '--syn' field is not enough. You can still use crafted packets with SYN and FIN and you may get a response if the receiver does not drop SYN+FIN packets. For example,

```
ubuntu@ubuntu-VirtualBox:~$ sudo iptables -A OUTPUT -p tcp -m tcp --syn -j DROP
ubuntu@ubuntu-VirtualBox:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A OUTPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -j DROP
ubuntu@ubuntu-VirtualBox:~$ sudo hping3 -c 2 -n -FS -p 80 netfilter.org
HPING netfilter.org (enp0s3 92.243.18.11): SF set, 40 headers + 0 data bytes
len=46 ip=92.243.18.11 ttl=255 id=20331 sport=80 flags=RA seq=0 win=0 rtt=5.9 ms
len=46 ip=92.243.18.11 ttl=255 id=20332 sport=80 flags=RA seq=1 win=0 rtt=8.5 ms

--- netfilter.org hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 5.9/7.2/8.5 ms
```

Though this is technically intended behavior, since '--syn' is only supposed to drop packets with just the SYN bit active.

3.5 Threat Modeling

3.5.1 Security Goals

- Confidentiality
    - In `iptables`, we can think of the system administrator as the user. The information that is supposed to be secret are the firewall rules defined by the system administrator. This is sensitive information that should not be leaked because an attacker could use it to hack the system. For example, if an attacker knew a rule restricted packets sent to an FTP service

running on port 21 to certain IP addresses, they could attempt to spoof those IP addresses.

- Integrity
    - In `iptables`, the primary information stored are the firewall rules defined by the system administrator. As `iptables` is a program specifically intended to be used to improve the security of systems, it should not allow attackers to change the rules because that would undermine the security of the system.
- Availability
    - The `iptables` program is used to configure and enforce firewall rules for a variety of services running on a system. As such, `iptables` should not allow attackers to stop others from using the very services it is meant to protect.

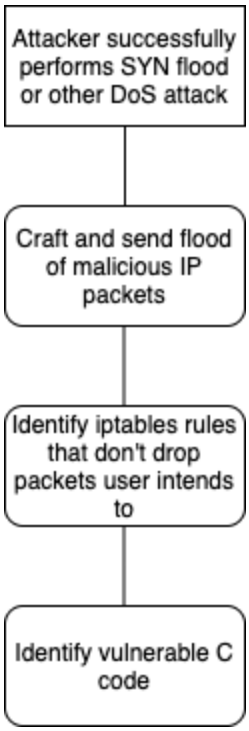
3.5.2 Security Policies/Requirements

- Confidentiality
    - No attacker should be able to access the information stored in the firewall rules.
- Integrity
    - No attacker should be able to change the information stored in the firewall rules.
- Availability
    - No attacker should be able to crash the `iptables` program.
    - No attacker should be able to overwhelm the `iptables` program, causing services to be unavailable.

3.5.3 Attack Tree

Note that because our security review is only concerned with security problems directly related to `iptables` itself, we do not consider threats in which an attack originates by an attacker executing the `iptables` command. This is because the there are only four ways to run `iptables`: (1) as the root user, (2) with sudo access, (3) by setting the SUID bit on the `iptables` executable, and (4) by setting file capabilities on a copy of the `iptables` executable. This implies an attacker gained privileged access through some vulnerability or attack surface external to `iptables`, and is therefore not relevant to our security evaluation. For similar reasons, we omit threats involving stealing user credentials or using a password cracker.


Attacker accesses the information stored in the iptables rules
Gain root or privileged access to the system
Exploit Buffer Overflow
Craft and send malicious IP packet
Identify vulnerable C code
Attacker modifies the iptables rules
Gain root or privileged access to the system
Exploit Buffer Overflow
Craft and send malicious IP packet
Identify vulnerable C code
Attacker crashes the iptables program
Exploit Segmentation Fault
Exploit Buffer Overflow
Exploit Memory Leak
Exploit C Exception
Craft and send flood of malicious IP packets
Craft and send malicious IP packet
Identify vulnerable C code
Identify vulnerable firewall rules



3.6 Penetration Testing

Penetration testing involves ethical hackers scaling simulated cyberattacks on a computer system to hunt down security vulnerabilities that need to be patched up. [1] In general, there are 13 steps to firewall penetration testing [2]:

Step 1. Locating The Firewall
Step 2. Conducting Traceroute
Step 3. Port Scanning
Step 4. Banner Grabbing
Step 5. Access Control Enumeration
Step 6. Identifying Firewall Architecture
Step 7. Testing The Firewall Policy
Step 8. Firewalking
Step 9. Port Redirection
Step 10. Internal And External Testing
Step 11. Test For Covert Channels
Step 12. HTTP tunneling

Step 13. Identify Firewall Specific Vulnerabilities

As we are only concerned with vulnerabilities within `iptables` itself which is a local software firewall, steps 3, 5, 6, 7, 10, and 13 are most relevant to our evaluation. We will focus on how the nmap network scanning tool can be used to determine what information an attacker can gather about a Linux server running `iptables` and verify that `iptables` is applying rules as expected. There are several different types of nmap scans we can perform [3][4]:

Method	Description	nmap Command
nmap Null Scan	Does not set any bits (TCP flag header is 0)	nmap -sN
nmap FIN Scan	Sets just the TCP FIN bit	nmap -sF
nmap Xmas Scan	Sets the FIN, PSH, and URG flags (lights the TCP packet up like a Christmas tree)	nmap -sX
nmap ACK Scan	Sets just the TCP ACK bit	nmap -sA
nmap SYN Scan	Sends TCP SYN packet	nmap -sS
nmap UDP Scan	Sends UDP packet to every targeted port	nmap -sU

These nmap scans all send packets in order to probe targeted ports. Then, nmap interprets the response and assigns one of 6 states to each port:

State	Description
open	A service is actively accepting TCP connections, UDP datagrams or SCTP associations on this port
closed	The port is accessible but there is no service listening on it
filtered	nmap cannot determine whether the port is open because a firewall (e.g. iptables) prevents its probes from reaching the port

unfiltered	The port is accessible, but nmap is unable to determine if it is open or closed
opened|filtered	nmap is unable to determine if the port is open or filtered
closed|filtered	nmap is unable to determine if the port is closed or filtered

Finding ports in an open state is often the goal of port scanning because they are an avenue for attack. Attackers use a combination of these scans in order to categorize ports into a state representing the most useful information. Although we ran out of time to run these penetration tests ourselves, our goals were to identify and test C code for iptables rules that may not be applied correctly or rules that produce responses with more information than necessary and test our hypotheses with nmap scans.

Sources: [1] [2] [3] [4]

4. Lessons Learned

We learned that C programming language is a pretty fragile language in terms of security. It is very easy to misuse standard C functions to expose security vulnerabilities. We can easily mitigate those problems by using secure versions of the functions (e.g. _s functions). However, as discussed in Why didn't gcc (or glibc) implement _s functions, those mitigations usually introduce extra instructions that slow down the execution of the program. This defeats the purpose of using C programming language to run the code as fast as possible. Therefore, many C programs out there still use easily misused C functions and carefully make sure there are no vulnerabilities whenever they invoke a potentially vulnerable C function. At the end of the day, it's all about the tradeoff. You have either a slow secure C program or a fast possibly insecure C program.

While looking for previous vulnerabilities, we discovered how bugs are reported and patched in an open source Linux project. From these previous vulnerabilities, we also learned that something as small as a faulty conversion rate can allow attackers to flood a network with traffic (see 3.2.2.3). We also learned that TCP packets with both the SYN and FIN bits set is unintended behavior and that attackers can craft their own packets to get by certain firewall rules (see 3.2.2.2).

From the live testing, we learned how to test an environment through a virtual machine. We also learned how to use specific tools like `hping3` to craft custom packets to test firewalls.

We also learned that although a security design review is often completed before there's any code, threat modeling can also be a helpful guide in a security evaluation. By creating an attack tree for `iptables`, we were able to increase our confidence that we had not overlooked any avenues of attack for the `iptables` program. During our investigation of `iptables` penetration testing, we learned that the nmap network scanning tool is used for one method of penetration testing `iptables`. After using tools such as OWASP Zed Attack Proxy (ZAP) for penetration testing web applications, we were surprised that no such automated tool existed for penetration testing firewalls such as `iptables`.

5. Recommendations for Future Evaluations

As already mentioned, it's important to recognize that the C language often sacrifices security for speed/efficiency. This presents itself in the form of code that often has vulnerabilities associated with native C functions. Specifically, code that processes some kind of input (from a user or file) is one of the most common sources of security flaws in a C program and would probably be worthwhile to investigate.

For large code bases, such as this one, code review by hand is a tedious task. Using static analysis tools, such as Flawfinder or Sonarcloud, can and should be used to narrow in on the exact sections of code to review for security vulnerabilities. If time permits, we could've looked into the rest of the potential vulnerabilities we didn't have time to work on.

As far as live testing, it may be worthwhile to write a script that tests all combinations of inputs. Specifically, a tester could write a script that creates a rule in `iptables` for a given combination of input flags, try to send and receive different crafted packets for various protocols, and then see if `iptables` handles the packet accordingly. Due to the limited amount of time for this study and all the possible combinations of input flags and crafted packets, this test could not be done in the given amount of time.

In terms of threat modeling and design review, future evaluations would benefit from constructing a component diagram and data flow diagram with trust boundaries since there is no official design document for `iptables`. This would provide opportunities for analyzing how changing `iptables` could lead to security flaws. In addition, a design

comparison with the successor of `iptables`, nftables, may yield design flaws that are relevant to security. Since we did not have time to perform penetration testing of `iptables`, that should definitely be included in a future security evaluation. In addition, another valuable evaluation approach would be to perform fuzz testing on `iptables`. Future evaluations should account for the fact that the process can take multiple weeks to run. Finally, another source for guiding future manual code reviews would be the bug tracker for the `iptables` project.

6. Work Breakdown

This section shows which elements of the report are done by whom.

Name	Sections
Junhong Wang	1, 2.1, 2.2, 2.3, 3.1.1, 3.1.2, 3.2.1, 3.3.1, 3.3.2, 3.3.3, 4, 7.1, 7.2
Stewart Dulaney	1, 2.5, 2.6, 3.5.1, 3.5.2, 3.5.3, 3.6, 4, 5
Brett Woltz	1, 2.2, 2.4, 3.2.2.2, 3.2.2.3, 3.2.2.4, 3.4.1, 3.4.2.1, 4, 5
Elyse Yao	1, 2, 2.1, 2.3, 3.2.2.1, 3.1.3, 3.1.4, 3.3.4, 3.3.5, 7.3
Joshua McInerney	1, 3.2.2.5, 3.3.6, 5

7. Supplementary Materials

We embedded all the files with large output in this section. You will find links/references to these files in the report (e.g. 7.1).

7.1. flawfinder_output.txt

```
Flawfinder version 2.0.15, (C) 2001-2019 David A. Wheeler.
Number of rules (primarily dangerous function names) in C/C++ ruleset: 222
Examining ./include/libiptc/ipt_kernel_headers.h
Examining ./include/libiptc/libxtc.h
Examining ./include/libiptc/xtcshared.h
```

```
Examining ./include/libiptc/libiptc.h
Examining ./include/libiptc/libip6tc.h
Examining ./include/xtables.h
Examining ./include/ip6tables.h
Examining ./include/iptables.h
Examining ./include/linux/netfilter_ipv6/ip6t_frag.h
Examining ./include/linux/netfilter_ipv6/ip6t_ah.h
Examining ./include/linux/netfilter_ipv6/ip6t_rt.h
Examining ./include/linux/netfilter_ipv6/ip6t_LOG.h
Examining ./include/linux/netfilter_ipv6/ip6t_mh.h
Examining ./include/linux/netfilter_ipv6/ip6t_NPT.h
Examining ./include/linux/netfilter_ipv6/ip6t_ipv6header.h
Examining ./include/linux/netfilter_ipv6/ip6t_opts.h
Examining ./include/linux/netfilter_ipv6/ip6t_hl.h
Examining ./include/linux/netfilter_ipv6/ip6t_srh.h
Examining ./include/linux/netfilter_ipv6/ip6_tables.h
Examining ./include/linux/netfilter_ipv6/ip6t_REJECT.h
Examining ./include/linux/netfilter_ipv6.h
Examining ./include/linux/netfilter_bridge.h
Examining ./include/linux/netfilter.h
Examining ./include/linux/types.h
Examining ./include/linux/netfilter/xt_RATEEST.h
Examining ./include/linux/netfilter/xt_connlimit.h
Examining ./include/linux/netfilter/xt_cgroup.h
Examining ./include/linux/netfilter/xt_state.h
Examining ./include/linux/netfilter/xt_tcpudp.h
Examining ./include/linux/netfilter/xt_limit.h
Examining ./include/linux/netfilter/xt_cluster.h
Examining ./include/linux/netfilter/xt_physdev.h
Examining ./include/linux/netfilter/ipset/ip_set.h
Examining ./include/linux/netfilter/xt_statistic.h
Examining ./include/linux/netfilter/xt_nfacct.h
Examining ./include/linux/netfilter/xt_ipvs.h
Examining ./include/linux/netfilter/nf_conntrack_common.h
Examining ./include/linux/netfilter/xt_bpf.h
Examining ./include/linux/netfilter/xt_owner.h
Examining ./include/linux/netfilter/x_tables.h
Examining ./include/linux/netfilter/xt_ipcomp.h
Examining ./include/linux/netfilter/xt_osf.h
Examining ./include/linux/netfilter/xt_devgroup.h
Examining ./include/linux/netfilter/xt_TCPOPTSTRIP.h
Examining ./include/linux/netfilter/xt_policy.h
Examining ./include/linux/netfilter/xt_CHECKSUM.h
Examining ./include/linux/netfilter/xt_comment.h
Examining ./include/linux/netfilter/xt_realm.h
Examining ./include/linux/netfilter/xt_conntrack.h
Examining ./include/linux/netfilter/xt_recent.h
Examining ./include/linux/netfilter/xt_SECMARK.h
```

```
Examining ./include/linux/netfilter/xt_sctp.h
Examining ./include/linux/netfilter/xt_addrtype.h
Examining ./include/linux/netfilter/nf_tables.h
Examining ./include/linux/netfilter/xt_connbytes.h
Examining ./include/linux/netfilter/xt_CLASSIFY.h
Examining ./include/linux/netfilter/xt_helper.h
Examining ./include/linux/netfilter/nf_tables_compat.h
Examining ./include/linux/netfilter/xt_socket.h
Examining ./include/linux/netfilter/xt_hashlimit.h
Examining ./include/linux/netfilter/nf_conntrack_tuple_common.h
Examining ./include/linux/netfilter/xt_u32.h
Examining ./include/linux/netfilter/xt_CONNSECMARK.h
Examining ./include/linux/netfilter/xt_IDLETIMER.h
Examining ./include/linux/netfilter/xt_string.h
Examining ./include/linux/netfilter/xt_set.h
Examining ./include/linux/netfilter/xt_time.h
Examining ./include/linux/netfilter/xt_mark.h
Examining ./include/linux/netfilter/xt_TPROXY.h
Examining ./include/linux/netfilter/xt_rpfilter.h
Examining ./include/linux/netfilter/xt_connmark.h
Examining ./include/linux/netfilter/xt_NFQUEUE.h
Examining ./include/linux/netfilter/xt_mac.h
Examining ./include/linux/netfilter/xt_SYNPROXY.h
Examining ./include/linux/netfilter/xt_connlabel.h
Examining ./include/linux/netfilter/xt_CT.h
Examining ./include/linux/netfilter/xt_dccp.h
Examining ./include/linux/netfilter/xt_cpu.h
Examining ./include/linux/netfilter/xt_TCPMSS.h
Examining ./include/linux/netfilter/xt_HMARK.h
Examining ./include/linux/netfilter/xt_NFLOG.h
Examining ./include/linux/netfilter/xt_length.h
Examining ./include/linux/netfilter/xt_esp.h
Examining ./include/linux/netfilter/xt_TEE.h
Examining ./include/linux/netfilter/xt_AUDIT.h
Examining ./include/linux/netfilter/xt_quota.h
Examining ./include/linux/netfilter/xt_dscp.h
Examining ./include/linux/netfilter/xt_multiport.h
Examining ./include/linux/netfilter/nfnetlink.h
Examining ./include/linux/netfilter/xt_LED.h
Examining ./include/linux/netfilter/xt_pkttype.h
Examining ./include/linux/netfilter/nf_nat.h
Examining ./include/linux/netfilter/xt_ecn.h
Examining ./include/linux/netfilter/xt_iprange.h
Examining ./include/linux/netfilter_ipv4/ip_queue.h
Examining ./include/linux/netfilter_ipv4/ipt_CLUSTERIP.h
Examining ./include/linux/netfilter_ipv4/ipt_ah.h
Examining ./include/linux/netfilter_ipv4/ipt_LOG.h
Examining ./include/linux/netfilter_ipv4/ipt_ECN.h
```

```
Examining ./include/linux/netfilter_ipv4/ipt_ttl.h
Examining ./include/linux/netfilter_ipv4/ip_tables.h
Examining ./include/linux/netfilter_ipv4/ipt_addrtype.h
Examining ./include/linux/netfilter_ipv4/ipt_ULOG.h
Examining ./include/linux/netfilter_ipv4/ipt_realm.h
Examining ./include/linux/netfilter_ipv4/ipt_REJECT.h
Examining ./include/linux/netfilter_bridge/ebt_mark_t.h
Examining ./include/linux/netfilter_bridge/ebt_ip.h
Examining ./include/linux/netfilter_bridge/ebt_802_3.h
Examining ./include/linux/netfilter_bridge/ebt_mark_m.h
Examining ./include/linux/netfilter_ipv4.h
Examining ./include/linux/kernel.h
Examining ./include/linux/filter.h
Examining ./include/linux/netfilter_arp.h
Examining ./include/linux/netfilter_arp/arp_tables.h
Examining ./include/linux/netfilter_arp/arpt_mangle.h
Examining ./include/libipulog/libipulog.h
Examining ./include/iptables/internal.h
Examining ./include/libipq/libipq.h
Examining ./libiptc/linux_stddef.h
Examining ./libiptc/libiptc.c
Examining ./libiptc/libip6tc.c
Examining ./libiptc/libip4tc.c
Examining ./libiptc/linux_list.h
Examining ./utils/nfsynproxy.c
Examining ./utils/nfbpf_compile.c
Examining ./utils/nfnl_osf.c
Examining ./extensions/libxt_CT.c
Examining ./extensions/libip6t_eui64.c
Examining ./extensions/libipt_REJECT.c
Examining ./extensions/libxt_connbytes.c
Examining ./extensions/libxt_addrtype.c
Examining ./extensions/libipt_SNAT.c
Examining ./extensions/libebt_ip6.c
Examining ./extensions/libxt_helper.c
Examining ./extensions/libxt_owner.c
Examining ./extensions/libxt_pkttype.c
Examining ./extensions/libip6t_icmp6.c
Examining ./extensions/libebt_nflog.c
Examining ./extensions/libipt_NETMAP.c
Examining ./extensions/libxt_CLASSIFY.c
Examining ./extensions/libxt_iprange.c
Examining ./extensions/libxt_ecn.c
Examining ./extensions/libebt_arp.c
Examining ./extensions/libxt_dccp.c
Examining ./extensions/libip6t_ipv6header.c
Examining ./extensions/libxt_LED.c
Examining ./extensions/libipt_MASQUERADE.c
```

```
Examining ./extensions/libxt_CHECKSUM.c
Examining ./extensions/libxt_conntrack.c
Parsing failed to find end of parameter list; semicolon terminated it in
((r), (l), offsetof(typeof(*(l)), sizeof(*info));

      struct xt_conntrack_mtinfo2 *info = cb->data;
      struct xt_conntrack_mtinfo3 up;

      memset(&up, 0, sizeof(up));
      memcpy(&up, info, sizeof(*info));
      up.
Examining ./extensions/libebt_dnat.c
Examining ./extensions/libxt_cpu.c
Examining ./extensions/libip6t_DNPT.c
Examining ./extensions/libebt_mark_m.c
Examining ./extensions/libxt_recent.c
Examining ./extensions/libebt_arpreply.c
Examining ./extensions/libxt_dscp.c
Examining ./extensions/libxt_limit.c
Examining ./extensions/tos_values.c
Examining ./extensions/libxt_policy.c
Examining ./extensions/libxt_TEE.c
Examining ./extensions/libipt_REDIRECT.c
Examining ./extensions/libxt_esp.c
Examining ./extensions/libebt_log.c
Examining ./extensions/libxt_ipcomp.c
Examining ./extensions/dscp_helper.c
Examining ./extensions/libxt_TCPOPTSTRIP.c
Examining ./extensions/libxt_tos.c
Examining ./extensions/libxt_time.c
Examining ./extensions/libipt_ttl.c
Examining ./extensions/libip6t_SNAT.c
Examining ./extensions/libxt_mac.c
Examining ./extensions/libxt_NFQUEUE.c
Examining ./extensions/libip6t_mh.c
Examining ./extensions/libxt_devgroup.c
Examining ./extensions/libxt_nfacct.c
Examining ./extensions/libxt_u32.c
Examining ./extensions/libip6t_HL.c
Examining ./extensions/libxt_standard.c
Examining ./extensions/libxt_statistic.c
Examining ./extensions/libip6t_frag.c
Examining ./extensions/libebt_mark.c
Examining ./extensions/libxt_connlimit.c
Examining ./extensions/libxt_set.c
Examining ./extensions/libxt_udp.c
Examining ./extensions/libxt_TRACE.c
Examining ./extensions/libxt_cgroup.c
```

```
Examining ./extensions/libxt_mark.c
Examining ./extensions/libip6t_REJECT.c
Examining ./extensions/libip6t_srh.c
Examining ./extensions/libipt_LOG.c
Examining ./extensions/libip6t_dst.c
Examining ./extensions/libipt_icmp.c
Examining ./extensions/libebt_ip.c
Examining ./extensions/libxt_tcp.c
Examining ./extensions/libipt_ECN.c
Examining ./extensions/libip6t_REDIRECT.c
Examining ./extensions/libarpt_mangle.c
Examining ./extensions/libip6t_DNAT.c
Examining ./extensions/libipt_CLUSTERIP.c
Examining ./extensions/libip6t_NETMAP.c
Examining ./extensions/libipt_realm.c
Examining ./extensions/libip6t_rt.c
Examining ./extensions/libxt_multiport.c
Examining ./extensions/libxt_comment.c
Examining ./extensions/libxt_length.c
Examining ./extensions/libipt_ah.c
Examining ./extensions/libxt_connlabel.c
Examining ./extensions/libxt_sctp.c
Examining ./extensions/libxt_TCPMSS.c
Examining ./extensions/libip6t_LOG.c
Examining ./extensions/libxt_SECMARK.c
Examining ./extensions/libebt_pkttype.c
Examining ./extensions/libxt_bpf.c
Examining ./extensions/libip6t_MASQUERADE.c
Examining ./extensions/libipt_DNAT.c
Examining ./extensions/libxt_rpfilter.c
Examining ./extensions/libxt_connmark.c
Examining ./extensions/libxt_osf.c
Examining ./extensions/libxt_SYNPROXY.c
Examining ./extensions/libxt_TPROXY.c
Examining ./extensions/libip6t_ah.c
Examining ./extensions/libebt_snat.c
Examining ./extensions/libxt_string.c
Examining ./extensions/libxt_IDLETIMER.c
Examining ./extensions/libxt_AUDIT.c
Examining ./extensions/libxt_RATEEST.c
Examining ./extensions/libip6t_SNPT.c
Examining ./extensions/libxt_quota.c
Examining ./extensions/libxt_CONNSECMARK.c
Examining ./extensions/libip6t_hbh.c
Examining ./extensions/libxt_set.h
Examining ./extensions/libxt_NFLOG.c
Examining ./extensions/libebt_802_3.c
Examining ./extensions/libebt_stp.c
```

```
Examining ./extensions/libxt_HMARK.c
Examining ./extensions/libxt_hashlimit.c
Examining ./extensions/libxt_cluster.c
Examining ./extensions/libxt_socket.c
Examining ./extensions/libipt_ULOG.c
Examining ./extensions/libebt_among.c
Examining ./extensions/libebt_vlan.c
Examining ./extensions/libxt_physdev.c
Examining ./extensions/libxt_icmp.h
Examining ./extensions/libxt_ipvs.c
Examining ./extensions/libebt_redirect.c
Examining ./iptables/iptables-save.c
Examining ./iptables/xtables-eb.c
Examining ./iptables/iptables.c
Examining ./iptables/xshared.h
Examining ./iptables/nft-arp.c
Examining ./iptables/xtables.c
Examining ./iptables/ip6tables.c
Examining ./iptables/xtables-eb-translate.c
Examining ./iptables/ip6tables-standalone.c
Examining ./iptables/iptables-xml.c
Examining ./iptables/xtables-multi.h
Examining ./iptables/nft-cache.c
Examining ./iptables/xtables-monitor.c
Examining ./iptables/xtables-arp-standalone.c
Examining ./iptables/nft-bridge.c
Examining ./iptables/xtables-eb-standalone.c
Examining ./iptables/nft-cmd.c
Examining ./iptables/nft.c
Examining ./iptables/xtables-save.c
Examining ./iptables/xtables-standalone.c
Examining ./iptables/nft-chain.c
Examining ./iptables/nft-ipv6.c
Examining ./iptables/nft-shared.c
Examining ./iptables/iptables-standalone.c
Examining ./iptables/xshared.c
Examining ./iptables/nft-arp.h
Examining ./iptables/xtables-restore.c
Examining ./iptables/xtables-translate.c
Examining ./iptables/nft-cache.h
Examining ./iptables/xtables-arp.c
Examining ./iptables/nft.h
Examining ./iptables/ip6tables-multi.h
Examining ./iptables/nft-cmd.h
Examining ./iptables/xtables-legacy-multi.c
Examining ./iptables/nft-bridge.h
Examining ./iptables/iptables-multi.h
Examining ./iptables/iptables-restore.c
```

```
Examining ./iptables/nft-shared.h
Examining ./iptables/nft-chain.h
Examining ./iptables/nft-ipv4.c
Examining ./iptables/xtables-nft-multi.c
Examining ./libipq/libipq.c
Examining ./libxtables/xtables.c
Examining ./libxtables/xtoptions.c
Examining ./libxtables/getethertype.c

FINAL RESULTS:

./extensions/libebt_ip.c:363:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libebt_ip.c:378:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libebt_ip6.c:204:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libebt_ip6.c:219:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libebt_log.c:122:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./extensions/libip6t_NETMAP.c:31:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libip6t_icmp6.c:103:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./extensions/libipt_NETMAP.c:26:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libipt_icmp.c:123:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./extensions/libxt_CT.c:148:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
```

```
  easily misused).
./extensions/libxt_dccp.c:20:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libxt_sctp.c:24:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./include/libipq/libipq.h:38:  [4] (format) fprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./include/xtables.h:425:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./include/xtables.h:634:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:90:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:220:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:324:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:327:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:328:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:330:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:331:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:335:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:340:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:341:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:343:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
```

```
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:344:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:396:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:404:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:410:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:412:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:664:  [4] (buffer) strcpy: 👀
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/ip6tables.c:843:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:953:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/ip6tables.c:1581:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/iptables.c:87:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:218:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:314:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:317:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:318:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:320:  [4] (format) printf:
```

```
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:321:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:325:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:330:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:331:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:333:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:334:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:388:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:396:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:402:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:404:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:655:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/iptables.c:840:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:949:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/iptables.c:1569:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
```

```
./iptables/nft-arp.c:420:  [4] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused).
./iptables/nft-arp.c:438:  [4] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused).
./iptables/nft-arp.c:456:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
./iptables/nft-arp.c:458:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
./iptables/nft-arp.c:482:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
./iptables/nft-arp.c:484:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
./iptables/nft-bridge.c:453:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-ipv4.c:158:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
./iptables/nft-shared.c:330:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:366:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:428:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:584:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
```

```
./iptables/nft-shared.c:674:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:694:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:735:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:738:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:739:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:741:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:742:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:746:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:751:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:752:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:754:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:755:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:765:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:773:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:780:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
```

```
  (CWE-134). Use a constant for the format specification.
./iptables/nft-shared.c:782:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/nft.c:2505:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:156:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:577:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:581:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:609:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:614:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:635:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:641:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xshared.c:659:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:661:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:719:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:722:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
```

```
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.h:15:  [4] (format) fprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xtables-arp.c:99:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xtables-eb.c:223:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xtables-eb.c:487:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xtables.c:87:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./iptables/xtables.c:217:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libiptc/libiptc.c:46:  [4] (format) fprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libiptc/libiptc.c:47:  [4] (format) fprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libiptc/libiptc.c:54:  [4] (format) fprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libiptc/libiptc.c:1111:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1142:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1163:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1254:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
```

```
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1277:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1283:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1324:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:1671:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:2562:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libiptc/libiptc.c:2596:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libxtables/xtables.c:79:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:89:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:574:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libxtables/xtables.c:1456:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
./libxtables/xtables.c:1801:  [4] (buffer) strcat:
```

```
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused).
./libxtables/xtables.c:2114:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:2118:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:2123:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:2128:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:2133:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:2137:  [4] (format) printf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:2369:  [4] (format) vsnprintf:
  If format strings can be influenced by an attacker, they can be exploited,
  and note that sprintf variations do not always \0-terminate (CWE-134). Use
  a constant for the format specification.
./utils/nfnl_osf.c:105:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./utils/nfnl_osf.c:128:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./extensions/libxt_statistic.c:69:  [3] (random) random:
  This function is not sufficiently random for security-related functions
  such as key and nonce creation (CWE-327). Use a more secure technique for
  acquiring random values.
./extensions/libxt_statistic.c:107:  [3] (random) random:
  This function is not sufficiently random for security-related functions
  such as key and nonce creation (CWE-327). Use a more secure technique for
  acquiring random values.
./include/linux/netfilter/xt_statistic.h:26:  [3] (random) random:
  This function is not sufficiently random for security-related functions
  such as key and nonce creation (CWE-327). Use a more secure technique for
  acquiring random values.
./iptables/ip6tables.c:1106:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
```

```
./iptables/iptables-restore.c:108:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/iptables-save.c:141:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/iptables-xml.c:552:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/iptables.c:1100:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xshared.c:258:  [3] (buffer) getenv:
  Environment variables are untrustable input if they can be set by an
  attacker. They can have any content and length, and the same variable can
  be set more than once (CWE-807, CWE-20). Check environment variables
  carefully before using them.
./iptables/xtables-arp.c:491:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-eb-translate.c:220:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-eb.c:751:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-monitor.c:642:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-restore.c:304:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
```

```
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-restore.c:433:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-save.c:149:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-save.c:213:  [3] (buffer) getenv:
  Environment variables are untrustable input if they can be set by an
  attacker. They can have any content and length, and the same variable can
  be set more than once (CWE-807, CWE-20). Check environment variables
  carefully before using them.
./iptables/xtables-translate.c:538:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables.c:494:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./libxtables/xtables.c:247:  [3] (buffer) getenv:
  Environment variables are untrustable input if they can be set by an
  attacker. They can have any content and length, and the same variable can
  be set more than once (CWE-807, CWE-20). Check environment variables
  carefully before using them.
./libxtables/xtables.c:250:  [3] (buffer) getenv:
  Environment variables are untrustable input if they can be set by an
  attacker. They can have any content and length, and the same variable can
  be set more than once (CWE-807, CWE-20). Check environment variables
  carefully before using them.
./libxtables/xtables.c:263:  [3] (buffer) getenv:
  Environment variables are untrustable input if they can be set by an
  attacker. They can have any content and length, and the same variable can
  be set more than once (CWE-807, CWE-20). Check environment variables
  carefully before using them.
./utils/nfnl_osf.c:437:  [3] (buffer) getopt:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./utils/nfsynproxy.c:197:  [3] (buffer) getopt_long:
```

```
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./extensions/libarpt_mangle.c:92:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libarpt_mangle.c:108:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_among.c:77:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_among.c:134:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./extensions/libebt_among.c:175:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_arp.c:95:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libebt_arp.c:132:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_arp.c:135:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_arpreply.c:60:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_dnat.c:57:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_ip.c:201:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libebt_ip.c:238:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libebt_ip.c:241:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./extensions/libebt_ip6.c:292:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libebt_ip6.c:310:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./extensions/libebt_snat.c:61:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libip6t_DNAT.c:93:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libip6t_DNAT.c:111:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libip6t_DNAT.c:127:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libip6t_DNAT.c:206:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libip6t_DNAT.c:208:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libip6t_DNAT.c:272:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libip6t_DNAT.c:297:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libip6t_DNAT.c:361:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libip6t_SNAT.c:87:  [2] (integer) atoi:
```

```
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libip6t_SNAT.c:105:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libip6t_icmp6.c:100:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libip6t_rt.c:50:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libip6t_rt.c:84:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libipt_CLUSTERIP.c:133:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libipt_CLUSTERIP.c:135:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libipt_CLUSTERIP.c:136:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./extensions/libipt_DNAT.c:97:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libipt_DNAT.c:115:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
```

```
  consider saving to an unsigned value if that is intended).
./extensions/libipt_DNAT.c:319:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libipt_DNAT.c:338:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libipt_DNAT.c:353:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libipt_SNAT.c:91:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libipt_SNAT.c:109:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libipt_icmp.c:120:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_CT.c:145:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_comment.c:55:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_conntrack.c:510:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./extensions/libxt_conntrack.c:541:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_conntrack.c:547:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_conntrack.c:560:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_hashlimit.c:272:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_hashlimit.c:422:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libxt_iprange.c:47:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_ipvs.c:73:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_ipvs.c:74:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_limit.c:70:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/libxt_owner.c:34:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_owner.c:43:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_policy.c:133:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./extensions/libxt_policy.c:134:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_policy.c:142:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_policy.c:143:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_set.c:98:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_set.c:179:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_string.c:96:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_time.c:215:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./extensions/libxt_time.c:232:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/libiptc/xtcshared.h:4:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/kernel.h:26:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/ipset/ip_set.h:255:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
```

```
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:16:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:30:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:39:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:53:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:71:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:77:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/x_tables.h:116:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_CT.h:19:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_CT.h:30:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_CT.h:31:  [2] (buffer) char:
```

```
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_IDLETIMER.h:40:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_IDLETIMER.h:49:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_LED.h:7:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_NFLOG.h:21:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_RATEEST.h:7:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_SECMARK.h:19:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_TEE.h:6:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_bpf.h:33:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_cgroup.h:18:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
```

```
  maximum possible length.
./include/linux/netfilter/xt_cgroup.h:33:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_comment.h:7:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_hashlimit.h:42:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_hashlimit.h:97:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_hashlimit.h:105:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_hashlimit.h:113:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_helper.h:6:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_mac.h:5:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_nfacct.h:13:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_nfacct.h:18:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
```

```
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_osf.h:41:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_osf.h:74:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_osf.h:75:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_osf.h:76:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_physdev.h:15:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_physdev.h:16:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_physdev.h:17:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_physdev.h:18:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_recent.h:31:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
```

```
./include/linux/netfilter/xt_recent.h:40:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_sctp.h:65:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./include/linux/netfilter/xt_string.h:17:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter/xt_string.h:18:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:35:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:36:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:62:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:63:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:108:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:133:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
```

```
./include/linux/netfilter_arp/arp_tables.h:155:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arp_tables.h:186:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arpt_mangle.h:8:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_arp/arpt_mangle.h:9:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_queue.h:27:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_queue.h:28:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_queue.h:32:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_queue.h:34:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_queue.h:47:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_tables.h:72:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
```

```
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_tables.h:73:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_tables.h:120:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_tables.h:155:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_tables.h:177:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ip_tables.h:208:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_LOG.h:16:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_ULOG.h:31:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_ULOG.h:40:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_ULOG.h:41:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_ULOG.h:43:  [2] (buffer) char:
```

```
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_ULOG.h:45:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv4/ipt_ULOG.h:46:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6_tables.h:65:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6_tables.h:66:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6_tables.h:124:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6_tables.h:195:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6_tables.h:217:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6_tables.h:248:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./include/linux/netfilter_ipv6/ip6t_LOG.h:16:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
```

```
  maximum possible length.
./iptables/ip6tables.c:859:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/ip6tables.c:1049:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/ip6tables.c:1052:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/iptables-restore.c:98:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-restore.c:100:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-restore.c:155:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/iptables-save.c:50:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-save.c:52:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/iptables-save.c:158:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/iptables-xml.c:54:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-xml.c:55:  [2] (buffer) char:
```

```
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-xml.c:56:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-xml.c:57:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-xml.c:545:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/iptables-xml.c:568:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/iptables.c:1045:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/iptables.c:1048:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-arp.c:48:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-arp.c:52:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./iptables/nft-arp.c:95:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-arp.c:109:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./iptables/nft-arp.c:285:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./iptables/nft-arp.c:406:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-arp.c:407:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-arp.c:426:  [2] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused). Risk is low because the
  source is a constant string.
./iptables/nft-arp.c:444:  [2] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused). Risk is low because the
  source is a constant string.
./iptables/nft-bridge.c:172:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.c:207:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.c:221:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-bridge.c:236:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-bridge.c:809:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:18:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:19:  [2] (buffer) char:
```

```
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:20:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:21:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:22:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:23:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:24:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:25:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:72:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-bridge.h:177:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-cache.c:98:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-cache.c:141:  [2] (buffer) char:
```

```
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-cache.c:349:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-cache.c:373:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-cache.c:428:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-cache.c:496:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-ipv4.c:139:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-ipv4.c:147:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./iptables/nft-ipv4.c:156:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./iptables/nft-ipv4.c:349:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv4.c:353:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv4.c:405:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-ipv6.c:132:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./iptables/nft-ipv6.c:146:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv6.c:160:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv6.c:231:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-ipv6.c:297:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv6.c:301:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv6.c:340:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-ipv6.c:364:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-shared.c:241:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:327:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:363:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:394:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:476:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:478:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:525:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:995:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-shared.c:999:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/nft-shared.c:1002:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/nft-shared.h:194:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft-shared.h:195:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:69:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:214:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:303:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:304:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:917:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
```

```
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:1002:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft.c:1250:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft.c:1356:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:1602:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:2775:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/nft.c:3184:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.c:262:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/xshared.c:470:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.c:557:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.c:589:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
```

```
  maximum possible length.
./iptables/xshared.c:626:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.c:791:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:108:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:110:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:112:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:114:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:115:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:116:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:117:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:118:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
```

```
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xshared.h:195:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-eb-translate.c:149:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-eb.c:284:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-eb.c:460:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-eb.c:573:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-eb.c:601:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-eb.c:607:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-monitor.c:53:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-monitor.c:128:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-monitor.c:222:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-monitor.c:275:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
```

```
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-monitor.c:331:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-monitor.c:372:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-monitor.c:611:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-restore.c:260:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-restore.c:347:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/xtables-save.c:166:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/xtables-translate.c:36:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./iptables/xtables-translate.c:61:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./iptables/xtables-translate.c:201:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-translate.c:203:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
```

```
./iptables/xtables-translate.c:206:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-translate.c:209:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables-translate.c:558:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/xtables.c:293:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:295:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:298:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:300:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:333:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:334:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:335:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:336:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:366:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:368:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:371:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:373:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:406:  [2] (buffer) memcpy:
```

```
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:408:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:411:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/xtables.c:413:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libip6tc.c:135:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libiptc/libip6tc.c:255:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libiptc/libiptc.c:110:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libiptc/libiptc.c:959:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libiptc/libiptc.c:988:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1123:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1255:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./libiptc/libiptc.c:1353:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libiptc/libiptc.c:1776:  [2] (buffer) memcpy:
```

```
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1826:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1865:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1972:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:2212:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:2432:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:2502:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:2581:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libiptc/libiptc.c:2657:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libxtables/getethertype.c:50:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/getethertype.c:52:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/getethertype.c:58:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libxtables/getethertype.c:81:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
```

```
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libxtables/xtables.c:133:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:139:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:146:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:377:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libxtables/xtables.c:407:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:605:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:710:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:826:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1381:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1384:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./libxtables/xtables.c:1390:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1450:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
```

```
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1463:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./libxtables/xtables.c:1472:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1558:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1573:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1613:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1647:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1655:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./libxtables/xtables.c:1666:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1670:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1674:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1700:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1710:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1714:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
```

```
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./libxtables/xtables.c:1727:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1738:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1744:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1749:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1796:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1808:  [2] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf. Risk is low because the source has a constant maximum length.
./libxtables/xtables.c:1846:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1869:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1914:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:1948:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1952:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./libxtables/xtables.c:1963:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
```

```
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1967:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1971:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:1991:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2001:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2005:  [2] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant string.
./libxtables/xtables.c:2020:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2142:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2143:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2144:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2145:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2147:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
```

```
./libxtables/xtables.c:2148:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2149:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2150:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2160:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2161:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2165:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2166:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2170:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2171:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2175:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2176:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2183:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2188:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2190:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
```

```
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2190:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2333:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtoptions.c:36:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtoptions.c:104:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:120:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:516:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:533:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:649:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:665:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:738:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:785:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:1094:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtoptions.c:1099:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
```

```
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./utils/nfnl_osf.c:113:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./utils/nfnl_osf.c:272:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./utils/nfnl_osf.c:275:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./utils/nfnl_osf.c:396:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./utils/nfnl_osf.c:398:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./utils/nfsynproxy.c:95:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./utils/nfsynproxy.c:206:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/dscp_helper.c:56:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_arp.c:97:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libebt_ip.c:203:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
```

```
./extensions/libebt_ip.c:294:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_ip6.c:138:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_ip6.c:297:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libebt_log.c:115:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_nflog.c:76:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_nflog.c:79:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libip6t_REJECT.c:121:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_REJECT.c:123:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_icmp6.c:81:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_icmp6.c:100:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_mh.c:80:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_mh.c:87:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:133:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:135:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:141:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:142:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
```

```
./extensions/libipt_icmp.c:101:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_icmp.c:120:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_CT.c:145:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_HMARK.c:199:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_HMARK.c:200:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_LED.c:76:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_LED.c:99:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_addrtype.c:98:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_addrtype.c:98:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:224:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:224:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:263:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:263:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:298:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:298:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:333:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
```

```
./extensions/libxt_conntrack.c:333:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:1063:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:1065:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:386:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:389:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:391:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:393:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:395:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:56:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:59:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:61:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:63:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:65:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_osf.c:73:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_recent.c:121:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120). Risk is low because the source is a
  constant string.
./extensions/libxt_sctp.c:203:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
```

```
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:77:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:158:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:257:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:435:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:604:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.h:65:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libxt_set.h:77:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libxt_set.h:108:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libxt_string.c:83:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_string.c:84:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libxt_string.c:98:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/ip6tables.c:256:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/ip6tables.c:1570:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-restore.c:218:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/iptables-restore.c:258:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-save.c:62:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
```

```
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-save.c:66:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-xml.c:159:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/iptables-xml.c:253:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/iptables-xml.c:520:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120). Risk is low because the source is a
  constant string.
./iptables/iptables-xml.c:534:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120). Risk is low because the source is a
  constant string.
./iptables/iptables.c:247:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables.c:1558:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-arp.c:235:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-arp.c:412:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-arp.c:425:  [1] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused). Risk is low because the
  source is a constant character.
./iptables/nft-arp.c:443:  [1] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused). Risk is low because the
  source is a constant character.
./iptables/nft-arp.c:459:  [1] (buffer) strncat:
  Easily used incorrectly (e.g., incorrectly computing the correct maximum
  size to add) [MS-banned] (CWE-120). Consider strcat_s, strlcat, snprintf,
  or automatically resizing strings.
./iptables/nft-arp.c:460:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-arp.c:485:  [1] (buffer) strncat:
```

```
  Easily used incorrectly (e.g., incorrectly computing the correct maximum
  size to add) [MS-banned] (CWE-120). Consider strcat_s, strlcat, snprintf,
  or automatically resizing strings.
./iptables/nft-arp.c:486:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-bridge.c:72:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-bridge.c:85:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-bridge.c:790:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-cache.c:171:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-cmd.c:43:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-ipv4.c:348:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-ipv4.c:352:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-ipv6.c:296:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-ipv6.c:300:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-shared.c:140:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:154:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:276:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:285:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:671:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
```

```
./iptables/nft-shared.c:882:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft.c:995:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft.c:1243:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft.c:1343:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:563:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/xshared.c:565:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/xshared.c:568:  [1] (buffer) strncat:
  Easily used incorrectly (e.g., incorrectly computing the correct maximum
  size to add) [MS-banned] (CWE-120). Consider strcat_s, strlcat, snprintf,
  or automatically resizing strings.
./iptables/xshared.c:569:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:595:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/xshared.c:597:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/xshared.c:600:  [1] (buffer) strncat:
  Easily used incorrectly (e.g., incorrectly computing the correct maximum
  size to add) [MS-banned] (CWE-120). Consider strcat_s, strlcat, snprintf,
  or automatically resizing strings.
./iptables/xshared.c:601:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:686:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:690:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-arp.c:854:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-eb-translate.c:329:  [1] (buffer) strlen:
```

```
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-eb.c:800:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-eb.c:917:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-restore.c:158:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-translate.c:35:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-translate.c:97:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables.c:916:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:163:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libiptc/libiptc.c:1145:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libiptc/libiptc.c:1307:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:1724:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:1726:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:2243:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:2379:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:2388:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:388:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
./libxtables/xtables.c:563:  [1] (buffer) strlen:
```

```
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:610:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:672:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:917:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:998:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1004:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1182:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1188:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1476:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1634:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1639:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1702:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate ofr check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1800:  [1] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused). Risk is low because the source is a constant character.
./libxtables/xtables.c:1935:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1940:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1993:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
```

```
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:2380:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtoptions.c:338:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtoptions.c:354:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./utils/nfnl_osf.c:412:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).

ANALYSIS SUMMARY:

Hits = 683
Lines analyzed = 67441 in approximately 1.06 seconds (63344 lines/second)
Physical Source Lines of Code (SLOC) = 53466
Hits@level = [0] 2178 [1] 147 [2] 384 [3]  24 [4] 128 [5]   0
Hits@level+ = [0+] 2861 [1+] 683 [2+] 536 [3+] 152 [4+] 128 [5+]   0
Hits/KSLOC@level+ = [0+] 53.5106 [1+] 12.7745 [2+] 10.0251 [3+] 2.84293 [4+]
2.39404 [5+]   0
Dot directories skipped = 8 (--followdotdir overrides)
Minimum risk level = 1

Not every hit is necessarily a security vulnerability.
You can inhibit a report by adding a comment in this form:
// flawfinder: ignore
Make *sure* it's a false positive!
You can use the option --neverignore to show these.

There may be other security vulnerabilities; review your code!
See 'Secure Programming HOWTO'
(https://dwheeler.com/secure-programs) for more information.
```

7.2. flawfinder_output_clean.txt

```
…
./iptables/ip6tables.c:220:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
...
./iptables/ip6tables.c:664:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
```

```
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/ip6tables.c:1581:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/iptables.c:218:  [4] (format) vfprintf:
...
./iptables/iptables.c:655:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/iptables.c:1569:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-arp.c:420:  [4] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused).
./iptables/nft-arp.c:438:  [4] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  [MS-banned] (CWE-120). Consider using strcat_s, strncat, strlcat, or
  snprintf (warning: strncat is easily misused).
...
./iptables/nft-bridge.c:453:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/nft-shared.c:330:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:366:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:428:  [4] (buffer) strcpy:
```

```
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:584:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:674:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/nft-shared.c:694:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/xshared.c:156:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/xshared.c:659:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:661:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:719:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./iptables/xshared.c:722:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/xtables-eb.c:487:  [4] (buffer) strcpy:
```

```
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./iptables/xtables.c:217:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
...
./libiptc/libiptc.c:1324:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
...
./libxtables/xtables.c:89:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./libxtables/xtables.c:574:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination
[MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
./libxtables/xtables.c:1456:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
...
./libxtables/xtables.c:2369:  [4] (format) vsnprintf:
  If format strings can be influenced by an attacker, they can be exploited,
  and note that sprintf variations do not always \0-terminate (CWE-134). Use
  a constant for the format specification.
./utils/nfnl_osf.c:105:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
./utils/nfnl_osf.c:128:  [4] (format) vfprintf:
  If format strings can be influenced by an attacker, they can be exploited
  (CWE-134). Use a constant for the format specification.
...
./iptables/ip6tables.c:1106:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/iptables-restore.c:108:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
```

```
./iptables/iptables-save.c:141:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/iptables-xml.c:552:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/iptables.c:1100:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
...
./iptables/xtables-arp.c:491:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-eb-translate.c:220:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-eb.c:751:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-monitor.c:642:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-restore.c:304:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-restore.c:433:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables-save.c:149:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
```

```
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
...
./iptables/xtables-translate.c:538:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./iptables/xtables.c:494:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
...
./utils/nfnl_osf.c:437:  [3] (buffer) getopt:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./utils/nfsynproxy.c:197:  [3] (buffer) getopt_long:
  Some older implementations do not protect against internal buffer
overflows
  (CWE-120, CWE-20). Check implementation on installation, or limit the size
  of all string inputs.
./extensions/libarpt_mangle.c:92:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libarpt_mangle.c:108:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./extensions/libebt_among.c:134:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./extensions/libxt_conntrack.c:510:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_conntrack.c:541:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_conntrack.c:547:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_conntrack.c:560:  [2] (buffer) memcpy:
```

```
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./extensions/libxt_limit.c:70:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
...
./extensions/libxt_policy.c:133:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_policy.c:134:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_policy.c:142:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./extensions/libxt_policy.c:143:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./iptables/ip6tables.c:1049:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/ip6tables.c:1052:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./iptables/iptables-restore.c:155:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./iptables/iptables-save.c:52:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/iptables-save.c:158:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./iptables/iptables-xml.c:568:  [2] (misc) fopen:
```

```
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/iptables.c:1045:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/iptables.c:1048:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./iptables/nft-ipv6.c:146:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-ipv6.c:160:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./iptables/nft-shared.c:241:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:327:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:363:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:394:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:476:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:478:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./iptables/nft-shared.c:525:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./iptables/nft-shared.c:999:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/nft-shared.c:1002:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
```

```
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./iptables/xshared.c:262:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./iptables/xshared.c:626:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
...
./iptables/xtables-eb.c:284:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./iptables/xtables-restore.c:347:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./iptables/xtables-save.c:166:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./iptables/xtables-translate.c:558:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./libiptc/libiptc.c:988:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1123:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./libiptc/libiptc.c:1353:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
```

```
./libiptc/libiptc.c:1776:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1826:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1865:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libiptc/libiptc.c:1972:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./libiptc/libiptc.c:2581:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libiptc/libiptc.c:2657:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./libxtables/getethertype.c:58:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libxtables/getethertype.c:81:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./libxtables/xtables.c:133:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:139:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:146:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:377:  [2] (misc) open:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
```

```
...
./libxtables/xtables.c:1914:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
...
./libxtables/xtables.c:2160:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2161:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2165:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2166:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2170:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2171:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2175:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2176:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2183:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2188:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtables.c:2190:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./libxtables/xtables.c:2190:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
...
```

```
./libxtables/xtoptions.c:104:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:120:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:516:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:533:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:649:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:665:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:738:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
./libxtables/xtoptions.c:785:  [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.
...
./libxtables/xtoptions.c:1099:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
...
./utils/nfnl_osf.c:398:  [2] (misc) fopen:
  Check when opening files - can an attacker redirect it (via symlinks),
  force the opening of special file type (e.g., device files), move things
  around to create a race condition, control its ancestors, or change its
  contents? (CWE-362).
./utils/nfsynproxy.c:95:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
./utils/nfsynproxy.c:206:  [2] (integer) atoi:
  Unless checked, the resulting number can exceed the expected range
  (CWE-190). If source untrusted, check both minimum and maximum, even if
the
  input had no minus sign (large numbers can roll over into negative number;
  consider saving to an unsigned value if that is intended).
./extensions/dscp_helper.c:56:  [1] (buffer) strlen:
```

```
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_arp.c:97:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libebt_ip.c:203:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libebt_ip.c:294:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_ip6.c:138:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
...
./extensions/libebt_log.c:115:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_nflog.c:76:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libebt_nflog.c:79:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libip6t_REJECT.c:121:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_REJECT.c:123:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_icmp6.c:81:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_icmp6.c:100:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_mh.c:80:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libip6t_mh.c:87:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:133:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:135:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
```

```
./extensions/libipt_REJECT.c:141:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_REJECT.c:142:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_icmp.c:101:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libipt_icmp.c:120:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_CT.c:145:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_HMARK.c:199:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_HMARK.c:200:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_LED.c:76:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_LED.c:99:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_addrtype.c:98:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_addrtype.c:98:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:224:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:224:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:263:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:263:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:298:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
```

```
./extensions/libxt_conntrack.c:298:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:333:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:333:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:1063:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_conntrack.c:1065:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:386:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:389:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:391:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:393:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_hashlimit.c:395:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:56:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:59:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:61:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:63:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_limit.c:65:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_osf.c:73:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
```

```
...
./extensions/libxt_sctp.c:203:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:77:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:158:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:257:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:435:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_set.c:604:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
...
./extensions/libxt_string.c:83:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./extensions/libxt_string.c:84:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./extensions/libxt_string.c:98:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/ip6tables.c:256:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/ip6tables.c:1570:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-restore.c:218:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/iptables-restore.c:258:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-save.c:62:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-save.c:66:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables-xml.c:159:  [1] (buffer) strncpy:
```

```
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/iptables-xml.c:253:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
...
./iptables/iptables.c:247:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/iptables.c:1558:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-arp.c:235:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-arp.c:412:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
...
./iptables/nft-arp.c:460:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
...
./iptables/nft-arp.c:486:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-bridge.c:72:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-bridge.c:85:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-bridge.c:790:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-cache.c:171:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-cmd.c:43:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-ipv4.c:348:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-ipv4.c:352:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-ipv6.c:296:  [1] (buffer) strncpy:
```

```
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-ipv6.c:300:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-shared.c:140:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:154:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:276:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:285:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft-shared.c:671:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/nft-shared.c:882:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft.c:995:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft.c:1243:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/nft.c:1343:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:563:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/xshared.c:565:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
...
./iptables/xshared.c:569:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:595:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./iptables/xshared.c:597:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
```

```
...
./iptables/xshared.c:601:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:686:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xshared.c:690:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-arp.c:854:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-eb-translate.c:329:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-eb.c:800:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-eb.c:917:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-restore.c:158:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-translate.c:35:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables-translate.c:97:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./iptables/xtables.c:916:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:163:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libiptc/libiptc.c:1145:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libiptc/libiptc.c:1307:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:1724:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:1726:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
```

```
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:2243:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:2379:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libiptc/libiptc.c:2388:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
...
./libxtables/xtables.c:563:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:610:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:672:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:917:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:998:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1004:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1182:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1188:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1476:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1634:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1639:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1702:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
...
```

```
./libxtables/xtables.c:1935:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtables.c:1940:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:1993:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtables.c:2380:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./libxtables/xtoptions.c:338:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).
./libxtables/xtoptions.c:354:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers [MS-banned] (CWE-120).
./utils/nfnl_osf.c:412:  [1] (buffer) strlen:
  Does not handle strings that are not \0-terminated; if given one it may
  perform an over-read (it could cause a crash if unprotected) (CWE-126).

ANALYSIS SUMMARY:

Hits = 683
Lines analyzed = 67441 in approximately 1.06 seconds (63344 lines/second)

...
```

7.3. pvs_output

```
/home/test/Downloads/iptables-1.8.7/extensions/libip6t_DNAT.c    208    warn
V512 A call of the 'memcpy' function will lead to underflow of the buffer '&
range'.
/home/test/Downloads/iptables-1.8.7/extensions/libip6t_DNAT.c    274    warn
V512 A call of the 'memcpy' function will lead to underflow of the buffer '&
range'.
/home/test/Downloads/iptables-1.8.7/extensions/libip6t_DNAT.c    299    warn
V512 A call of the 'memcpy' function will lead to underflow of the buffer '&
range'.
/home/test/Downloads/iptables-1.8.7/extensions/libip6t_DNAT.c    363    warn
V512 A call of the 'memcpy' function will lead to underflow of the buffer '&
range'.
/home/test/Downloads/iptables-1.8.7/extensions/libip6t_mh.c    113    warn
V575 The potential null pointer is passed into 'strchr' function. Inspect
```

```
the first argument. Check lines: 113, 112.
/home/test/Downloads/iptables-1.8.7/extensions/libipt_DNAT.c    65    warn
V701 realloc() possible leak: when realloc() fails in allocating memory,
original pointer 'info' is lost. Consider assigning realloc() to a temporary
pointer.
/home/test/Downloads/iptables-1.8.7/extensions/libipt_SNAT.c    59    warn
V701 realloc() possible leak: when realloc() fails in allocating memory,
original pointer 'info' is lost. Consider assigning realloc() to a temporary
pointer.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_MARK.c    290    warn
V576 Incorrect format. Consider checking the third actual argument of the
'sscanf' function. A pointer to the unsigned int type is expected.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_MARK.c    302    warn
V576 Incorrect format. Consider checking the third actual argument of the
'sscanf' function. A pointer to the unsigned int type is expected.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_MARK.c    315    warn
V576 Incorrect format. Consider checking the third actual argument of the
'sscanf' function. A pointer to the unsigned int type is expected.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    365
warn    V547 Expression 'tmp == 0' is always false.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    392
warn    V1048 The 'ud->mult' variable was assigned the same value.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    720
err    V1051 Consider checking for misprints. It's possible that the 'burst'
should be checked here.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    751
err    V1051 Consider checking for misprints. It's possible that the 'burst'
should be checked here.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    781
err    V1051 Consider checking for misprints. It's possible that the 'burst'
should be checked here.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    969
warn    V581 The conditional expressions of the 'if' statements situated
alongside each other are identical. Check lines: 966, 969.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_hashlimit.c    1121
warn    V581 The conditional expressions of the 'if' statements situated
alongside each other are identical. Check lines: 1118, 1121.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_iprange.c    62    warn
V1004 The 'ia6' pointer was used unsafely after it was verified against
nullptr. Check lines: 59, 62.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_iprange.c    70    warn
V1004 The 'ia4' pointer was used unsafely after it was verified against
nullptr. Check lines: 67, 70.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_multiport.c    133
warn    V557 Array overrun is possible. The value of '++ i' index could
reach 15.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_multiport.c    134
warn    V557 Array overrun is possible. The value of 'i' index could reach
```

```
15.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_sctp.c    76    warn
V575 The potential null pointer is passed into 'strchr' function. Inspect
the first argument. Check lines: 76, 74.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_sctp.c    173    warn
V575 The potential null pointer is passed into 'strcasecmp' function.
Inspect the first argument. Check lines: 173, 168.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_string.c    155    warn
V576 Incorrect format. Consider checking the third actual argument of the
'sscanf' function. A pointer to the unsigned int type is expected.
/home/test/Downloads/iptables-1.8.7/extensions/libxt_tcp.c    49    warn
V575 The potential null pointer is passed into 'strchr' function. Inspect
the first argument. Check lines: 49, 48.
/home/test/Downloads/iptables-1.8.7/iptables/ip6tables.c    845    warn
V618 It's dangerous to call the 'printf' function in such a manner, as the
line being passed could contain format specification. The example of the
safe code: printf("%s", str);
/home/test/Downloads/iptables-1.8.7/iptables/ip6tables.c    955    warn
V618 It's dangerous to call the 'printf' function in such a manner, as the
line being passed could contain format specification. The example of the
safe code: printf("%s", str);
/home/test/Downloads/iptables-1.8.7/iptables/iptables-xml.c    150    warn
V547 Expression 'curChain[0]' is always true.
/home/test/Downloads/iptables-1.8.7/iptables/iptables.c    842    warn
V618 It's dangerous to call the 'printf' function in such a manner, as the
line being passed could contain format specification. The example of the
safe code: printf("%s", str);
/home/test/Downloads/iptables-1.8.7/iptables/iptables.c    951    warn
V618 It's dangerous to call the 'printf' function in such a manner, as the
line being passed could contain format specification. The example of the
safe code: printf("%s", str);
/home/test/Downloads/iptables-1.8.7/libipq/libipq.c    109    warn    V641
The size of the '& h->peer' buffer is not a multiple of the element size of
the type 'struct sockaddr'.
/home/test/Downloads/iptables-1.8.7/libipq/libipq.c    170    warn    V641
The size of the '& h->peer' buffer is not a multiple of the element size of
the type 'struct sockaddr'.
/home/test/Downloads/iptables-1.8.7/libipq/libipq.c    243    warn    V641
The size of the '& h->local' buffer is not a multiple of the element size of
the type 'struct sockaddr'.
/home/test/Downloads/iptables-1.8.7/libiptc/libiptc.c    501    warn    V575
The potential null pointer is passed into 'memset' function. Inspect the
first argument.
/home/test/Downloads/iptables-1.8.7/libiptc/libiptc.c    820    warn    V619
The array 'pr->entry' is being utilized as a pointer to single object.
/home/test/Downloads/iptables-1.8.7/libiptc/libiptc.c    2245    err    V547
Expression is always false.
/home/test/Downloads/iptables-1.8.7/libiptc/libiptc.c    2381    err    V547
```

```
Expression is always false.
/home/test/Downloads/iptables-1.8.7/libiptc/libiptc.c    2646    warn
V619 The array 'r->entry' is being utilized as a pointer to single object.
/home/test/Downloads/iptables-1.8.7/libiptc/libiptc.c    2651    warn
V619 The array 'r->entry' is being utilized as a pointer to single object.
/home/test/Downloads/iptables-1.8.7/libxtables/xtables.c    1458    warn
V512 A call of the 'sprintf' function will lead to overflow of the buffer
'buf'.
/home/test/Downloads/iptables-1.8.7/libxtables/xtables.c    1869    warn
V1048 The 'addrptmp' variable was assigned the same value.
/home/test/Downloads/iptables-1.8.7/libxtables/xtables.c    2145    warn
V1009 Check the array initialization. Only the first element is initialized
explicitly. The rest elements are initialized with zeros.
/home/test/Downloads/iptables-1.8.7/libxtables/xtables.c    2146    warn
V1009 Check the array initialization. Only the first element is initialized
explicitly. The rest elements are initialized with zeros.
/home/test/Downloads/iptables-1.8.7/libxtables/xtables.c    2147    warn
V1009 Check the array initialization. Only the first element is initialized
explicitly. The rest elements are initialized with zeros.
/home/test/Downloads/iptables-1.8.7/libxtables/xtoptions.c    870    err
V590 Consider inspecting this expression. The expression is excessive or
contains a misprint.
/home/test/Downloads/iptables-1.8.7/utils/nfnl_osf.c    209    warn    V1048
The 'wc' variable was assigned the same value.
/home/test/Downloads/iptables-1.8.7/utils/nfnl_osf.c    278    warn    V1032
The pointer 'buf' is cast to a more strictly aligned pointer type.
/home/test/Downloads/iptables-1.8.7/utils/nfnl_osf.c    380    err    V597
The compiler could delete the 'memset' function call, which is used to flush
'buf' buffer. The memset_s() function should be used to erase the private
data.
```

7.4. lshw_output

```
ubuntu@ubuntu-VirtualBox:~$ sudo lshw
[sudo] password for ubuntu:
ubuntu-virtualbox
    description: Computer
    product: VirtualBox
    vendor: innotek GmbH
    version: 1.2
    serial: 0
    width: 64 bits
    capabilities: smbios-2.5 dmi-2.5 vsyscall32
    configuration: family=Virtual Machine
```

```
uuid=ABD2F363-47C6-864A-8594-16DCE2409068
  *-core
       description: Motherboard
       product: VirtualBox
       vendor: Oracle Corporation
       physical id: 0
       version: 1.2
       serial: 0
     *-firmware
          description: BIOS
          vendor: innotek GmbH
          physical id: 0
          version: VirtualBox
          date: 12/01/2006
          size: 128KiB
          capacity: 128KiB
          capabilities: isa pci cdboot bootselect int9keyboard int10video
acpi
     *-memory
          description: System memory
          physical id: 1
          size: 1GiB
     *-cpu
          product: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
          vendor: Intel Corp.
          physical id: 2
          bus info: cpu@0
          width: 64 bits
          capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8
apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx
rdtscp x86-64 constant_tsc rep_good nopl xtopology nonstop_tsc cpuid
tsc_known_freq pni pclmulqdq monitor ssse3 cx16 pcid sse4_1 sse4_2 x2apic
movbe popcnt aes xsave avx rdrand hypervisor lahf_lm abm 3dnowprefetch
invpcid_single pti fsgsbase avx2 invpcid rdseed clflushopt flush_l1d
     *-pci
          description: Host bridge
          product: 440FX - 82441FX PMC [Natoma]
          vendor: Intel Corporation
          physical id: 100
          bus info: pci@0000:00:00.0
          version: 02
          width: 32 bits
          clock: 33MHz
        *-isa
             description: ISA bridge
             product: 82371SB PIIX3 ISA [Natoma/Triton II]
             vendor: Intel Corporation
             physical id: 1
```

```
             bus info: pci@0000:00:01.0
             version: 00
             width: 32 bits
             clock: 33MHz
             capabilities: isa bus_master
             configuration: latency=0
        *-ide
             description: IDE interface
             product: 82371AB/EB/MB PIIX4 IDE
             vendor: Intel Corporation
             physical id: 1.1
             bus info: pci@0000:00:01.1
             version: 01
             width: 32 bits
             clock: 33MHz
             capabilities: ide isa_compat_mode pci_native_mode bus_master
             configuration: driver=ata_piix latency=64
             resources: irq:0 ioport:1f0(size=8) ioport:3f6
ioport:170(size=8) ioport:376 ioport:d000(size=16)
        *-display
             description: VGA compatible controller
             product: SVGA II Adapter
             vendor: VMware
             physical id: 2
             bus info: pci@0000:00:02.0
             version: 00
             width: 32 bits
             clock: 33MHz
             capabilities: vga_controller bus_master rom
             configuration: driver=vmwgfx latency=64
             resources: irq:18 ioport:d010(size=16) memory:e0000000-e0ffffff
memory:f0000000-f01fffff memory:c0000-dffff
        *-network
             description: Ethernet interface
             product: 82540EM Gigabit Ethernet Controller
             vendor: Intel Corporation
             physical id: 3
             bus info: pci@0000:00:03.0
             logical name: enp0s3
             version: 02
             serial: 08:00:27:88:11:0b
             size: 1Gbit/s
             capacity: 1Gbit/s
             width: 32 bits
             clock: 66MHz
             capabilities: pm pcix bus_master cap_list ethernet physical tp
10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
             configuration: autonegotiation=on broadcast=yes driver=e1000
```

```
driverversion=7.3.21-k8-NAPI duplex=full ip=10.0.2.15 latency=64 link=yes
mingnt=255 multicast=yes port=twisted pair speed=1Gbit/s
             resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
        *-generic
             description: System peripheral
             product: VirtualBox Guest Service
             vendor: InnoTek Systemberatung GmbH
             physical id: 4
             bus info: pci@0000:00:04.0
             version: 00
             width: 32 bits
             clock: 33MHz
             configuration: driver=vboxguest latency=0
             resources: irq:20 ioport:d040(size=32) memory:f0400000-f07fffff
memory:f0800000-f0803fff
        *-multimedia
             description: Multimedia audio controller
             product: 82801AA AC'97 Audio Controller
             vendor: Intel Corporation
             physical id: 5
             bus info: pci@0000:00:05.0
             version: 01
             width: 32 bits
             clock: 33MHz
             capabilities: bus_master
             configuration: driver=snd_intel8x0 latency=64
             resources: irq:21 ioport:d100(size=256) ioport:d200(size=64)
        *-usb
             description: USB controller
             product: KeyLargo/Intrepid USB
             vendor: Apple Inc.
             physical id: 6
             bus info: pci@0000:00:06.0
             version: 00
             width: 32 bits
             clock: 33MHz
             capabilities: ohci bus_master cap_list
             configuration: driver=ohci-pci latency=64
             resources: irq:22 memory:f0804000-f0804fff
           *-usbhost
                product: OHCI PCI host controller
                vendor: Linux 5.8.0-44-generic ohci_hcd
                physical id: 1
                bus info: usb@1
                logical name: usb1
                version: 5.08
                capabilities: usb-1.10
                configuration: driver=hub slots=12 speed=12Mbit/s
```

```
              *-usb
                   description: Human interface device
                   product: USB Tablet
                   vendor: VirtualBox
                   physical id: 1
                   bus info: usb@1:1
                   version: 1.00
                   capabilities: usb-1.10
                   configuration: driver=usbhid maxpower=100mA
speed=12Mbit/s
        *-bridge
             description: Bridge
             product: 82371AB/EB/MB PIIX4 ACPI
             vendor: Intel Corporation
             physical id: 7
             bus info: pci@0000:00:07.0
             version: 08
             width: 32 bits
             clock: 33MHz
             capabilities: bridge
             configuration: driver=piix4_smbus latency=0
             resources: irq:9
        *-sata
             description: SATA controller
             product: 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI
mode]
             vendor: Intel Corporation
             physical id: d
             bus info: pci@0000:00:0d.0
             version: 02
             width: 32 bits
             clock: 33MHz
             capabilities: sata pm ahci_1.0 bus_master cap_list
             configuration: driver=ahci latency=64
             resources: irq:21 ioport:d240(size=8) ioport:d248(size=4)
ioport:d250(size=8) ioport:d258(size=4) ioport:d260(size=16)
memory:f0806000-f0807fff
     *-pnp00:00
          product: PnP device PNP0303
          physical id: 3
          capabilities: pnp
          configuration: driver=i8042 kbd
     *-pnp00:01
          product: PnP device PNP0f03
          physical id: 4
          capabilities: pnp
          configuration: driver=i8042 aux
     *-scsi:0
```

```
          physical id: 5
          logical name: scsi1
          capabilities: emulated
        *-cdrom
             description: DVD reader
             product: CD-ROM
             vendor: VBOX
             physical id: 0.0.0
             bus info: scsi@1:0.0.0
             logical name: /dev/cdrom
             logical name: /dev/dvd
             logical name: /dev/sr0
             version: 1.0
             capabilities: removable audio dvd
             configuration: ansiversion=5 status=nodisc
     *-scsi:1
          physical id: 6
          logical name: scsi2
          capabilities: emulated
        *-disk
             description: ATA Disk
             product: VBOX HARDDISK
             vendor: VirtualBox
             physical id: 0.0.0
             bus info: scsi@2:0.0.0
             logical name: /dev/sda
             version: 1.0
             serial: VBe2d107d8-0d3bbea8
             size: 10GiB (10GB)
             capabilities: partitioned partitioned:dos
             configuration: ansiversion=5 logicalsectorsize=512
sectorsize=512 signature=a9946843
           *-volume:0
                description: Windows FAT volume
                vendor: mkfs.fat
                physical id: 1
                bus info: scsi@2:0.0.0,1
                logical name: /dev/sda1
                logical name: /boot/efi
                version: FAT32
                serial: 4bc9-524b
                size: 510MiB
                capacity: 512MiB
                capabilities: primary bootable fat initialized
                configuration: FATs=2 filesystem=fat mount.fstype=vfat
mount.options=rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=iso88
59-1,shortname=mixed,errors=remount-ro state=mounted
           *-volume:1
```

```
               description: Extended partition
               physical id: 2
               bus info: scsi@2:0.0.0,2
               logical name: /dev/sda2
               size: 9725MiB
               capacity: 9725MiB
               capabilities: primary extended partitioned
partitioned:extended
             *-logicalvolume
                  description: EXT4 volume
                  vendor: Linux
                  physical id: 5
                  logical name: /dev/sda5
                  logical name: /
                  version: 1.0
                  serial: 444dec06-ebb3-4a06-ae24-1913bdb54a5d
                  size: 9725MiB
                  capacity: 9725MiB
                  capabilities: journaled extended_attributes large_files
huge_files dir_nlink recover 64bit extents ext4 ext2 initialized
                  configuration: created=2021-02-23 10:09:00
filesystem=ext4 lastmountpoint=/ modified=2021-03-08 00:59:59
mount.fstype=ext4 mount.options=rw,relatime,errors=remount-ro
mounted=2021-03-08 01:00:00 state=mounted
```