VxWorks 高危漏洞 CVE-2019-12256 漏洞分析

H4lo / 2019-08-05 07:30:00 / 浏览数 5216 安全技术 漏洞分析 顶(0) 踩(0)

原文链接： https://go.armis.com/hubfs/White-papers/Urgent11%20Technical%20White%20Paper.pdf

漏洞介绍

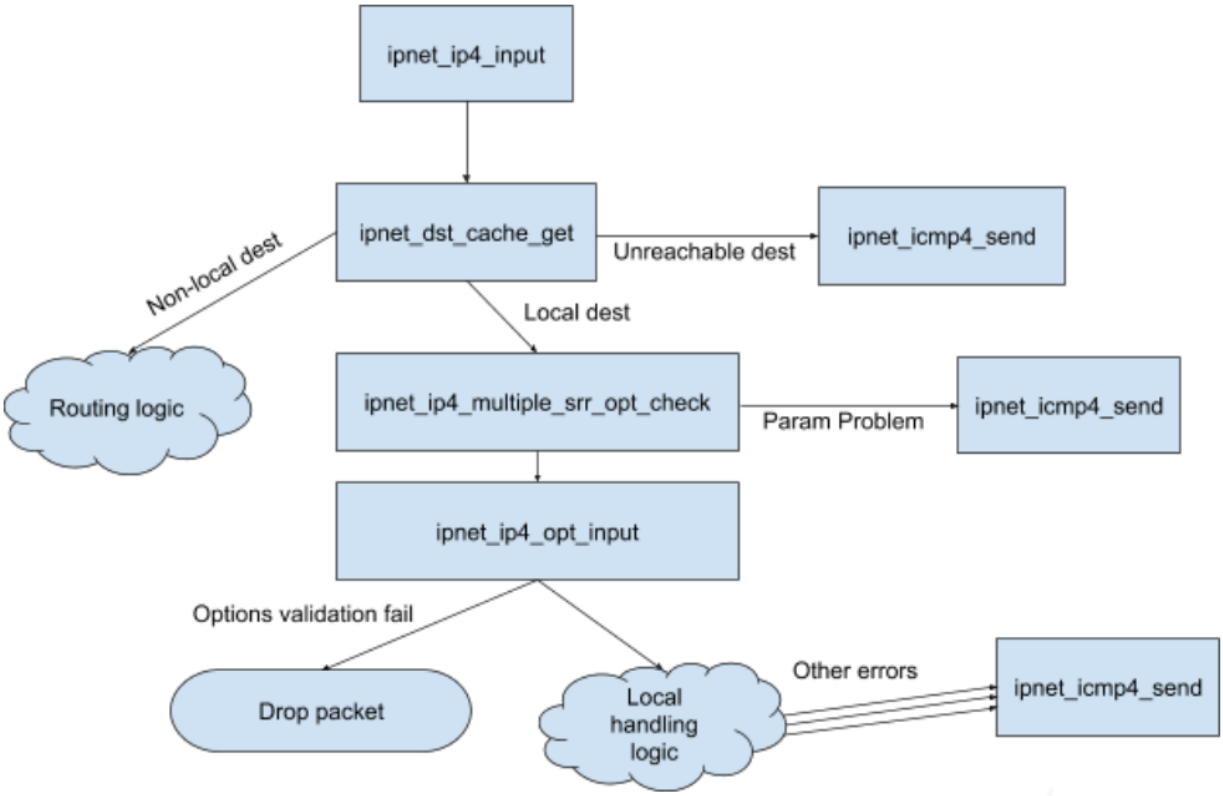
CVE-2019-12256 漏洞是 'URGENT/11' 中的一个，漏洞成因是在解析 IPv4 数据包 IP 选项时的栈溢出。

漏洞分析

在 VxWorks 的 IPnet 堆栈中，IPv4 协议在模块ipnet2/src/ipnet_ip4.c 中实现。该函数 ipnet_ip4_input 是传入IPv4
数据包的入口点。该功能首先执行基本功能验证头，然后调用 ipnet_dst_cache_get。这将根据报文的源IP地址和目的IP地址执行查找处理程序回调。

例如，如果目标地址与本地地址匹配，将调用 ipnet_ip4_local_rx 处理程序。除此以外可以转发（路由）分组，或者可能是目的地不可达条件触发。

某些错误条件（例如 Destination Unreachable）将触发要发送的 ICMP 错误数据包调用 ipnet_icmp4_send。
当由于IP格式错误而发生错误时，也会调用此函数选项和 ICMP 参数，问题数据包将被发送。 这可以通过多个阶段发生正在解析哪些 IP 选项：在
ipnet_ip4_multiple_srr_opt_check 中，仅验证该选项字段或任何各种选项中存在 LSRR 或 SSRR 选项的一个实例解析函数 - ipnet_ip4opt * _ rx。 在
ipnet_icmp4_send 中，构造了一个错误包，并且传入（格式错误的）数据包中的某些选项字段将被复制。 这是 ipnet_icmp4_copyopts 的功能。
以上流程如下图所示：



IPv4 packet handling flow chart, with calls to the ICMP error sending function

如上所示，在解析传入的IPv4分组时，各种代码流可以导致ICMP消息被发送以响应错误的（格式错误的）数据包。 将使用 ipnet_icmp4_send
函数发送响应ICMP数据包，它将尝试从传入中复制某些 IP 选项使用函数 ipnet_icmp4_copyopts 将数据包发送到传出数据包。

在至少两个代码流中，在完全解析传入的数据包和传入的 IP 之前，将发送传出的 ICMP
数据包选项完全验证是合法的，或者即使它们已经未通过验证。这个设计缺陷可能会导致此 ipnet_icmp4_send 上下文中的堆栈溢出。

自VxWorks版本6.9.3起，此漏洞就存在。两个不同的流可能导致此漏洞：

1. 发送到目标 MAC 地址但目标 IP 地址不是 a 的数据包目标的单播地址，并且无法访问它，这将导致调用
   ipnet_ip4_dst_unreachable。
2. 通过其 MAC 和 IP 地址定向到目标的普通单播数据包。 这个将发生在 ipnet_ip4_multiple_srr_opt_check
   函数中。当传入数据包的目标IP无法访问时，第一个流发生在 ipnet_ip4_input 中根据 ipnet_dst_cache_new 的返回值，它指示数据包应如何路由：

```
return_code = ipnet_dst_cache_new(&v37, ipnet_ip4_dst_cache_rx_ctor, &v45);
if ( return_code < 0 )
{
  return_code = -1 * return_code;
  // If dst is unreachable (EHOSTUNREACH, ENETUNREACH or EACCES)
  if ( return_code == 51 || return_code == 65 || return_code == 13 )
  {
    ipnet_ip4_dst_unreachable(packet, return_code);
```

Decompiled snippet from *ipnet_ip4_input*

然后函数 ipnet_ip4_dst_unreachable 将调用 ipnet_icmp4_send 并附加原始数据包到 icmp_param 结构。 在 ipnet_icmp4_send 中，来自失败（传入）数据包的 IP 选项将是由将传递给 ipnet_icmp4_copyopts 的结构引用：

```
struct Ipnet_copyopts_param
{
  void *options_ptr;
  int total_opt_size;
  int which_opts_to_copy;
};

int ipnet_icmp4_send(Ipnet_icmp_param *icmp_param, Ip_bool is_igmp)
{
  Ipnet_icmp_param *icmp_param;
  Ipcom_pkt *failing_pkt;
  struct Ipnet_copyopts_param options_to_copy;
  struct Ipnet_ip4_sock_opts opts;
  ...
  options_to_copy.options_ptr = NULL;
  options_to_copy.total_opt_size = 0;
  // By default, copy SSRR and LSRR options.
  options_to_copy.which_opts_to_copy = 0x208;
  ...
  if ( !is_igmp )
  {
    if ( icmp_param->type == IPNET_ICMP4_TYPE_DST_UNREACHABLE ||
         icmp_param->type == IPNET_ICMP4_TYPE_PARAMPROB ||
         ... ) {

      failing_pkt = icmp_param->recv_pkt;
      ...
      ip_hdr = failing_pkt->data[icmp_param->recv_pkt->ipstart];
      ...
      options_to_copy.total_opt_size = 4 * (*ip_hdr & 0xF) - 20;
      if ( options_to_copy.total_opt_size )
        options_to_copy.options_ptr = ip_hdr + 20;
      ...
    }
  }
  ...
```

```
  ipnet_icmp4_copyopts(icmp_param, &options_to_copy, &opts, &ip4_info);
  ...
}
```

缺省情况下，options_to_copy 结构将指示 ipnet_icmp4_copyopts 复制 SSRR 和LSRR 从失败的数据包到输出数据包的选项。
当失败的数据包尚未验证时，包含有效的 IP 选项（与上述流程一样），可以强制执行 ipnet_icmp4_copyopts 功能
将故障数据包中的多个 SSRR 或 LSRR 选项复制到opts结构上，在 ipnet_icmp4_send 的堆栈上分配。 此外，这些复制的选项可能包含非法否则不允许的结构。
例如，发送IP数据包时在 IP 选项字段中包含以下字节，将发生堆栈溢出：

Type (LSRR)	Length	LSRR-Pointer	Type (LSRR)	Length	LSRR-Pointer
\x83	\x03	\x27	\x83	\x03	\x27

在此示例中，IP选项字段中包含两个LSRR选项。
这些LSRR选项不包含任何路由条目（每个选项长度只有3个字节），SRR-Pointer字段指向结束选项。如前所述，接收SRR选项的主机需要记录所有记录路由条目，反转它们
可以在ipnet_icmp4_copyopts中查看此功能：

```
int ipnet_icmp4_copyopts(Ipnet_icmp_param *icmp_param,
                         struct Ipnet_copyopts_param *copyopts_param,
                         struct Ipnet_ip4_sock_opts *opts, void *ip4_info)
{
  ...
  while ( 1 ) {
    current_opt = ipnet_ip4_get_ip_opt_next(current_opt,
                                            copyopts_param->options_ptr,
                                            copyopts_param->total_opt_size);
    if ( !current_opt )
      break;
    opt_type = *current_opt;
    if ( copyopts_param->which_opts_to_copy & (1 << (opt_type & 31)) )
    {
      ...
      if ( opt_type == 0x83 || opt_type == 0x89 ) {
        // IP_IPOPT_LSRR or IP_IPOPT_SSRR
        srr_ptr_offset = 39;
        srr_opt = (srr_opt_t *)&opts->opts[opts->len];
        // Limits max ptr offset to 39,
        // (but doesn't validate this offset is within the current option)
```

```
        if ( (int)current_opt[2] <= 39 )
           srr_ptr_offset = current_opt[2];
        offset_to_current_route_entry = srr_ptr_offset - 5;
        ...
        srr_opt->type = opt_type;
        current_route_entry = &current_opt[offset_to_current_route_entry];
        srr_opt->length = 3;
        srr_opt->route_ptr = 4;
        while ( offset_to_current_route_entry > 0 ) {
          memcpy((char *)srr_opt + srr_opt->length, current_route_entry, 4);
          current_route_entry -= 4;
          offset_to_current_route_entry -= 4;
          srr_opt->length += 4;
        }
        memcpy((char *)srr_opt + srr_opt->length, &icmp_param->to, 4);
        srr_opt->length += 4;
        total_opts_len = opts->len + srr_opt->length;
      }
    }
  }
  ...
}
```


ipnet_icmp4_copyopts 中的代码将使用这些 SRR-Pointer 字段作为最终路由条目的偏移量在 SRR 选项中，将所有路由条目复制到传出数据包选项 opts，即在 ipnet_icmp4_send 的堆栈上分配。

输入缓冲区中的每个 LSRR
选项都是3个字节长，但它将生成一个43字节的复制输出选项（3个字节的标题，36个字节的路由条目，4个字节的a）。由于没有验证（在此上下文中）失败数据包不包含多
\
LSRR选项，将导致发送此类型的多个选项在opts的溢出中，这是在堆栈上分配的40字节数组。如上所述，当包含数据包的数据包时，此漏洞也可能由另一个流触发以上选项
*_
rx中的选项解析过程失败时函数，或在ipnet_ip4_multiple_srr_opt_check函数中，发送ICMP错误消息响应（通过ipnet_ip4_opt_icmp_param_prob）。发生这种情况时，

总结

幸运的是，因为漏洞依赖于发送带有无效 IP 选项的数据包，所以它不可能通过互联网利用。遇到数据包的第一个路由器将丢弃它。因此，漏洞只能由 LAN 上的攻击者利用。

由于此漏洞在解析 IP 标头本身时，也可以通过发送一个特殊的 crash 包来触发在广播数据包中使用无效IP选项。这可以允许攻击者同时定位多个易受攻击的设备。

目录

RSS 关于社区 友情链接 社区小黑板