DENIAL-OF-SERVICE ATTACK ON PROGRAMMABLE ROUTER

[1] YING HU, [2]LEI ZHUANG

[1]Department of Information and Engineering, Zhengzhou University, Zhengzhou 450000, Henan, China

[2]Department of Information and Engineering, Zhengzhou University, Zhengzhou 450000, Henan, China

ABSTRACT

Because of the limitation of routers which use ASICs, there are more and more programmable routers be adopted at present, it is a trend that the ASIC routers will be replaced. Programmable router brings out flexibility, at the same time, it brings out security problems. This paper introduced a type of network attack. The attack sends appropriate length of packet, while application code could not detect that the length of the packet is out of range, and the operation of inserting header leads to buffer overflow, thereby, this forms denial-of-service attack.

Keywords: *Programmable Router, Attack; Denial-of-Service, ASIC Router*

1. INTRODUCTION

Network security is an important concern in the Internet. Most efforts have focused on end-systems since it presented no practical attack target on the network infrastructure itself.

In the past, most network routers used ASICs (application-specific integrated circuits) to forward packets. Since the functionality of an ASIC cannot be changed once it has been designed, the use of general-purpose processor provides more flexibility to adjust a router's functionality after production [1]. Therefore, there is an ongoing shift toward developing routers based on programmable packet processors [4] rather than based on ASICs.

A side-effect of this shift is that it gives rise to a new class of vulnerabilities and corresponding attacks. Routers based on ASICs cannot be changed except by replacing actual hardware, so there are no practical attack targets. In contrast, routers based on general-purpose processors that run software to perform packet processing functions exhibit the same vulnerabilities in conventional end-systems and embedded systems since attackers can attempt to crash the systems, change its operation, extract information, etc.

In our work, we show a practical example of such an attack.

2. RELATED WORK

Routers that use software for packet processing include workstation-based routers [2, 6] which use operating system, and programmable routers [7]. Out of consideration of the performance speed, most router systems use multi-core packet processors (network processors) [9, 10]. Commercial examples of network processors are the Intel IXP2400 [11], the EZchip NP-3 [12], the LSI APP [13], the Cavium Octeon [14], and the Cisco QuantumFlow [15]. The number of packet processors in these chips ranges from 8 in the IXP2400 to more than 100 in the Cisco Silicon packet processor (SPP).

Stress the vulnerabilities in routers is also important in the next generation network. In these network researches, network virtualization [8, 5] and network service [3] are the main direction. Both of them use programmable packet processors. Thus, developing defense mechanisms to protect the packet processors in router systems is critical for the continued success of the Internet.

In the meantime, piracy becomes increasingly rampant as the customers can easily duplicate and redistribute the received multimedia content to a large audience [1].

3. VULNERABILITIES AND ATTACKS IN NETWORK INFRASTRUCTURE

3.1 Attack Classification

The classification can be seen on Table 1.

Table 1: Examples Of Network Attacks And Defenses

Attack target	Goal of attack	Attack examples	defenses
End-system	Data access and modification	Hacking, phishing, espionage	Virus scanner, firewall, network intrusion

	Denial-of-service	Denial-of-service attack via botnets	detection system
Control plane	Data access and modification	Malicious route announcement, DNS cache poisoning	Secure routing protocols, secure DNS(DNSSEC)
	Denial-of-service	DNS recursion attack	
Data plane	Data access and modification	Eavesdropping, man-in-the-middle attack	Secure network protocol (IPSec, TLS)
	Denial-of-service	Exploit of vulnerable packet processing code	Processing monitor

Focus of this paper

3.2 Security Model

In our work, we use a security model that reflects the operation of current Internet. Basically, we assume that the packet processing code on a router is benign and an attacker aims to exploit vulnerabilities.

3.2.1 Security requirements

1) The operation of the router does not change under attack;
2) Malicious traffic should be identified and be discarded.

3.2.2 Attacker capabilities

1) An attacker can send arbitrary data and control packets;
2) It can modify instruction and data memory through exploits;
3) It cannot modify the source code or binary of the protocol implementation before it is installed on the router;
4) It cannot physically access the router.

Based on this security model, we present a concrete attack.

4. A BUFFER OVERFLOW ATTACK

Routers can perform a variety of protocol processing operations. For our attack example, we assume that the protocol processing operation includes adding a header to a packet.

The premise of our attack is that the packet processing code is benign and does not contain intentionally malicious code. The attacker sends a carefully crafted packet to one of the router's network interface cards. The processing of this packet turns the 'good' protocol routine that runs on the network processor into 'bad' one. There is nothing inherently wrong with the packet or the application code, but the combination of the two can lead to the processor's malfunctioning.

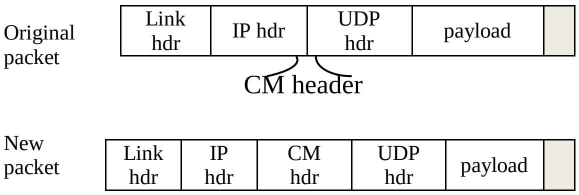

Figure 1: Protocol Header Insertion

```
#define MAX_PKT 1484
Int generate_CM_header(into rig_pkt[], unsigned
short len1,unsigned short len2)
{
    Int new_pkt_buf[MAX_PKT];
    Unsigned short sum;
    Sum=len1+len2;
    If (sum>MAX_PKT) { return -1;}
    Else{
         Memcpy(new_pkt_buf+len1), orig_pkt,
len2);
         …
}
    …
    Return 0;
}
```

Figure 2: Example application code

Figure 2 shows code for inserting the new CM header to the original packet, the code contains a vulnerability which is caused by an integer overflow.

As we know, integers can represent values within a given range. For example, the integer type 'unsigned short' ranges from 0 to 65535. When a variable exceeds the upper limit, value wraps around zero to stay within the allowed limits. For example:

```
Unsigned short sum;
Unsigned short one=65532;
Unsigned short two=8;
Sum = one + two;
```

The value of sum is not 65540 but 4 due to the limited amount of memory space that is assigned to it.

The code steps are:

Parse headers to identify header boundary between IP and UDP.

Shift the UDP to the right to make room for the CM header.

Insert CM header in packet.

The underlying danger is caused by step 1, and is lead to by step 2 and 3 which is realized by function Memcpy.

If an attacker sends a packet with a regular, oversized packet, the size check will fail. However, If an attacker sends a packet with malformed UDP length field, then the code performs incorrectly:

CM_hdr_size + UDP_length =12 + 65532 =8(incorrect due to integer overflow)

CM_hdr_size + UDP_length < MAX_PKT(even though it is not)

65532 bytes are copied into the new_pkt_buf, which can only accommodate 1486 bytes.

The result is buffer overflow attack, which will overwrite the processor's stack. To be exactly, the new packet buffer will overflow and start rewriting the local variables of the current frame, continue with the stack pointer and finally overwrite the return address of the current frame as well. When the return address is overwritten, the program will jump to whichever address the attacker has chosen! Thereby, the attacker can make the program jump to malicious code which is carried inside the packet payload.

5. CONCLUSION

In this paper, we describe a type of network attack. The attack exploits vulnerabilities in the packet processor of modern routers. We show how integer vulnerabilities can be used to execute arbitrary attack code. To our knowledge, this work can provides an important step toward understanding and correcting vulnerabilities in the modern and future network infrastructure.

REFERENCES:

[1] W. Eatherton, "The push of network processing to the top of the pyramid", *Keynote Presentation at ACM/IEEE Symposium on Architectures for Networking and Communication Systems*, October, 2005.

[2] E. Kohler, R. Morris, B. Chen, J. Jannotti, M. F. Kaashoek, "The Click modular router", *ACM Transactions on Computer Systems*, Vol. 18, No. 3, 2000, pp. 263-297.

[3] T. Wolf, "Service-centric end-to-end abstractions in next-generation networks", *Proceedings of Fifteenth IEEE International Conference on Computer Communications and Networks*, October, 2006, pp. 79-86.

[4] Q. Wu, D. Chasaki, T. Wolf, "Implementation of a simplified network processor", *Proceedings of IEEE International Conference on High Performance Switching and Routing*, June, 2010.

[5] J. S. Turner, D. E. Taylor, "Diversifying the Internet", *Proceedings of IEEE Global Communications Conference, November*, 2005.

[6] N. C. Hutchinson, L. L. Peterson, "The x-kernel: An architecture for implementing network protocols", *IEEE Transactions on Software Engineering*, Vol. 17, No. 1, 1991, pp. 64-76.

[7] L. Ruf, K. Farkas, H. Hug, B. Plattner, "Network services on service extensible routers", *Proceedings of Seventh Annual International Working Conference on Active Networking*, November, 2005.

[8] T. Anderson, L. Peterson, S. Shenker, J. Turner, "Overcoming the Internet impasse through virtualization", *Computer*, Vol. 38, No. 4, 2005, pp. 34-41.

[9] J. S. Turner, P. Crowley, J. DeHart, A. Freestone, B. Heller, F. Kuhns, S. Kumar, J. Lockwood, J. Lu, M. Wilson, C. Wiseman, D. Zar, "Supercharging PlanetLab: a high performance, multi-application, overlay network platform", *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, August, 2007, pp. 85-96.

[10] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford, "In VINI veritas: realistic and controlled network experimentation", *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, August, 2006, pp. 3-14.

[11] Intel Second Generation Network Processor, Intel Corporation, 2005. http://www.intel.com/design/network/products/npfamily/.

[12] NP-3 – 30-Gigabit Network Processor with Integrated Traffic Management, EZchip Technologies Ltd., Yokneam, Israel, May 2007, http: //www.ezchip.com/.

[13] APP3300 Family of Advanced Communication Processors, LSI Corporation, Aug. 2007, http://www.lsi.com/.

[14] OCTEON Plus CN58XX 4 to 16-Core MIPS64-Based SoCs, Cavium Networks, Mountain View, CA, 2008.

[15] The Cisco QuantumFlow Processor: Cisco's Next Generation Network Processor, Cisco Systems, Inc., San Jose, CA, Feb. 2008.