A Systematic Analysis of the Juniper Dual EC Incident

APR 26TH, 2016

论文下载

2015年12月Juniper公司表示自己的VPN路由器系统ScreenOS被植入未经授权的代码，作者分析了该系统的随机数生成和IKE协议中的问题和带来的危害，可无须任何流量直接解密一次IKE握手和VPN流量

Background

Juniper使用的随机数生成器是Dual EC PRNG。Dual EC使用椭圆曲线上的两个点P、Q来生成随机数，在2007 Crypto，Shumow 和Ferguson指出 得到 logPQ = e 可以恢复出生成器的内部状态，从而预测之后的输出。因此2013年NIST不建议使用Dual EC，Juniper表示Dual EC仍在使用，但是Q点与NIST标准不同，是由Juniper选择的点，而且Dual EC的结果不直接输出，再经过X9.31 PRNG后输出。

- CVE-2015-77554 (“Administrative Access”)：ssh 后门密码 <<< %s(un=‘%s’) = %u
- CVE-2015-77565 (“VPN Decryption”)：There is no way to detect that this vulnerability was exploited.

对vulnerable和patch后的binary比较分析，发现攻击者修改了原始版本中的Q，patch 把Q改回原始的值。但是根据Juniper之前的申明，即使可以修改Q也无法解密，输出还要经过X9.31 PRNG。这就产生了对Juniper之前声明的准确性的疑问：

1. 为什么Q点的改变会导致VPN被解密的漏洞?
2. 为什么X9.31没能保护Q点被改变带来的问题?
3. ScreenOS中PRNG代码的历史是什么?
4. Juniper的Q点是如何产生的?
5. Juniper产生的Q点能否被攻击?

ScreenOS RNG

$s_1 = x(s_0P)$

$r_1 = x(s_1Q)$ $s_2 = x(s_1P)$

$r_2 = x(s_2Q)$ output = $lsb_{30}(r_1) \,||\, msb_2(\, lsb_{30}(r_2)\,)$

$eP = Q$, $dQ = P$, $d = e^{-1} \bmod order$ $x(d*s_1Q) = x(s_1Q) = s2$

Listing 1: The core ScreenOS 6.2 PRNG subroutines.

```
1  void prng_reseed(void) {
2    blocks_generated_since_reseed = 0;
3    if (dualec_generate(prng_temporary, 32) != 32)
4      error_handler("FIPS ERROR: PRNG failure, unable to reseed\n", 11);
5    memcpy(prng_seed, prng_temporary, 8);
6    prng_output_index = 8;
7    memcpy(prng_key, &prng_temporary[prng_output_index], 24);
8    prng_output_index = 32;
9  }
10 void prng_generate(void) {
11   int time[2];
12
13   time[0] = 0;
14   time[1] = get_cycles();
15   prng_output_index = 0;
16   ++blocks_generated_since_reseed;
17   if (!one_stage_rng())
18     prng_reseed();
19   for (; prng_output_index <= 0x1F; prng_output_index += 8) {
20     // FIPS checks removed for clarity
21     x9_31_generate_block(time, prng_seed, prng_key, prng_block);
22     // FIPS checks removed for clarity
23     memcpy(&prng_temporary[prng_output_index], prng_block, 8);
24   }
25 }
```

- prng_reseed()：用Dual EC产生32字节随机数（prng_temorary)，并作为prng_seed和prng_key
- prng_generate(): 有可能调用prng_reseed,prng_seed和prng_key输入到X9.31产生prng_block,拼接成32字节prng_temorary

代码问题：

1 prng_reseed()和prng_generate()共享全局变量prng_temorary和prng_output_index,在一次reseed之后prng_output_index值为32，导致prng_generate中调用X9.31的for循环不被执行。随机数直接来自DualEC

2 默认配置中one_state_rng总为true(?),因此每次生成随机数都会调用prng_reseed

Listing 2: The core ScreenOS 6.1 PRNG subroutine.

```
1  void prng_generate(char *output) {
2    unsigned int index;
3    int time[2];
4
5    index = 0;
6    // FIPS checks removed for clarity
7    if ( blocks_generated_since_reseed++ > 9999 )
8      prng_reseed();
9    // FIPS checks removed for clarity
10   time[0] = 0;
11   time[1] = get_cycles();
12   do {
13     // FIPS checks removed for clarity
14     prng_generate_block(time, prng_seed, prng_key, prng_block);
15     // FIPS checks removed for clarity
16     memcpy(&output[index], prng_output_block, min(20-index, 8));
17     index += min(20-index, 8);
18   } while ( index <= 19 );
19 }
```

- X9.31 PRNG
- index在ScreenOS6.1.0中是局部变量
- 不总reseed
- 输出20字节

Interaction with IKE

Internet Key Exchange：IPsec的密钥建立协议

- Phase 1 (IKEv1)/Initial Exchange (IKEv2):建立IKE Security Association
- Phase 2 (IKEv1)/CREATE_CHILD_SA (IKEv2)：建立IKE SA来保护上层数据，可以有多个SA
- Phase 1：nounce，DH密钥交换，产生会话密钥。通过签名或预共享密钥来认证
- Phase 2：nounce，从会话密钥中产生子SA的密钥

攻击IKE的整体思路：

1 利用Phase 1中的随机数，恢复Dual EC的内部状态；
2 预测DH私钥，计算DH协商后的密钥

3　解密Phase2的流量，获得nonce和public key

4　计算子SA的共享密钥，解密VPN流量

问题：

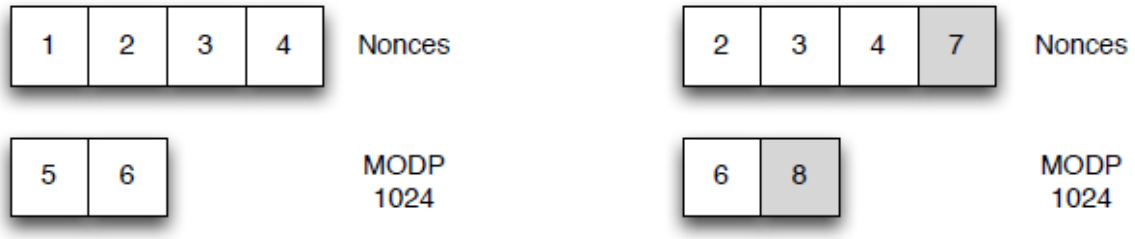
1　nonce 大小：

- 获得Dual EC完整的输出才能恢复内部状态。标准中取输出的低30字节，需要枚举高2字节，216 IKE中的nonce是可变长的，8~256bytes。 ScreenOS中nonce是30byte

2　nonce 和DH key的生成：

- 如果key先于nonce生成，就无法从nonce预测key。
- **随机数池**：出于性能的考虑，会先生成好nonce和key。随机数减少后再向其中添加
- ScreenOS对nonce，DH密钥各自维护一个队列，nonce队列的长度是所有DH队列长度之和的两倍

生成的总体优先级是nonce、低强度的密钥、高强度的密钥。



(a) At system startup　　(b) After a DH exchange

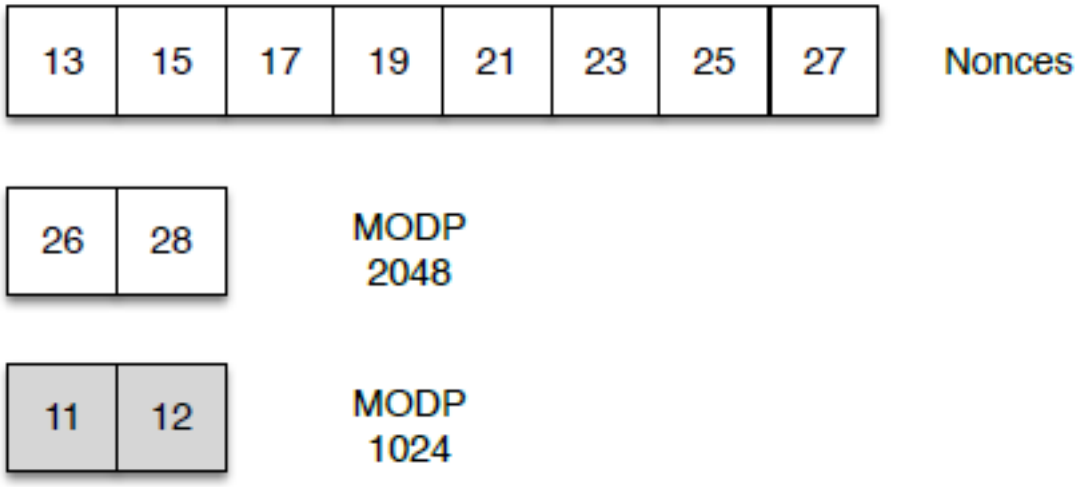


Figure 2: Queue state after 8 MODP 2048 exchanges. Numbers indicate generation order, and stale values are highlighted. If several connections have been made to the same DH group, the other DH group can grow stale as all nonces that were generated before those keys are used up.

1 Non-DH Phase 2 Exchanges：
   - phase2中没有DH交换，只交换nonce

Exploiting the Vulnerability against IKE

利用的几个关键条件

1 Dual EC的实现
2 Dual EC那两个函数的连锁错误
3 默认配置中总是reseed，和条件2一起导致随机数都直接来自DualEC
4 6.2中每次生成32byte随机数
5 nonce先于 key 生成

modify a firmware of VPN device

Passively detecting Juniper ScreenOS

- 区分真随机数和Dual EC的输出？
- ScreenOS中老版本OpenSSL 的bug： Juniper Dual EC can’t generate a zero byte in nonce
- shodan 发现约26,000个

Discussion

1. 为什么Q点的改变会导致VPN被解密的漏洞?:从nonce预测DH key
2. 为什么X9.31没能保护Q点被改变带来的问题？ X9.31 PRNG没有运行，Dual EC的结果被直接输出
3. ScreenOS中PRNG代码的历史是什么？ 6.1中X9.31，6.2中Dual E及若干问题
4. Juniper的Q点是如何产生的？ unable to answer
5. Juniper产生的Q点能否被攻击？unable to answer