musalbas
Personal homepage of
Mustafa Al-Bassam

Equation Group's BENIGNCERTAIN tool - a remote exploit to extract Cisco VPN private keys

18 Aug 2016

In the Equation Group dump that contained NSA hacking tools, there was an overlooked tool called BENIGNCERTAIN.

Analysis of the tool shows that it appears to be a remote exploit for Cisco PIX devices that sends an Internet Key Exchange (IKE) packet to the victim machine, causing it to dump some of its memory. The memory dump can then be parsed to extract an RSA private key and other sensitive configuration information.

The tool references Cisco PIX versions 5.2(9) to 6.3(4), which was released in 2004. It is also worth noting that the Cisco PIX line of products are at their end-of-life.

The exploit consists of three binaries, each consisting of an individual step in the exploitation process.

The first step is executing `bc-genpkt`, which generates an IKE packet of arbitrary size and fills some of it with arbitrary data.

```
Usage: ./bc-genpkt [-h] [-o <file>] [-f <X>] [-r] [-s] [-v[vv]] size

    -h  help/usage
    -o file write data to named file
    -f X    fill remainder of large packets with character 'X'
    -r  randomize the initiator cookie
    -s  randomize the SPI
    -v[vv]  verbosity - show lengths, packet dumps, etc
    size    size of new packet, should be 96 <= size <= 65536 bytes

    Packets larger than 2528 bytes will be filled with random data
    unless the -f option is used.
```

This generates a packet file which can be used as input to the binary `bc-id`, which sends the packet to the victim host. Hector Martin notes that it sends a IKE packets with a large Group-Prime option, and speculates that if the victim host is replying using the request length but only filling in the requested 768 bit prime, then it returns a buffer of uninitialised data following it.

```
Usage:
   ./bc-id -t <dest IP> []
  Options:
      -t <dest IP>
      -l <local port>
      -p <remote port>
      -I <infile name>  [defaults to sendpacket.raw]
      -O <outfile name> [defaults to ".raw"]
      -f <packetfile name> Reads in packet from a file.
      -h              print this message
      -q              quiet mode.  Doesn't print hex of response pack
```

The strings in the `bc-id` binary shows that the program seems to patch some memory and look for a start string in the response. However Hector Martin notes that this appears to be dead unreferenced code.

```
Patched memory at location %d with %s
```

```
  *** Error: Start string never found.
    %.*s
```

The `bc-id` program then outputs a file which can be used as input to `bc-parser`, a program that parses the response.

```
BENIGNCERTAIN parser v1.0
Usage:  ./bc-parser [-c] [-h] [-u] [-v]

    -c  Generate PIX configuration commands
    -h  Usage
    -v  Verbose - print uninteresting/default stuff
    -u  Hex-dump unknown structures only
    -x  Hex-dump all structures (overrides -u)

      BENIGNCERTAIN .raw file
```

The strings in the `bc-parser` binary shows what the tool extracts, which appears to include VPN configuration details and RSA private keys.

```
RSA private key structure at offset 0x%04x, size 0x%x bytes:
    *** Found probable RSA private key ***
RSA public key structure at offset 0x%04x, size 0x%x bytes:
    *** Found probable RSA public key ***
RSA key structure at offset 0x%04x, size 0x%x bytes:
    RSA keys were generated at %s
VPN group structure at offset 0x%04x, size 0x%x bytes
    Split-tunnel ACL:  0x%08x       %s
    Idle-time:         0x%08x       [%d seconds]
    Max-time:          0x%08x       [%d %s]
    PFS:               0x%08x       %s
    Clear-client-cfg:  0x%08x       %s
    User-idle-timeout: 0x%08x       [%d seconds]
    Authen. server:    0x%08x       %s
    Secure-unit-auth:  0x%08x       %s
    User authen.:      0x%08x       %s
    Device pass-thru:  0x%08x       %s
```

It also shows that it appears to be reading a memory dump.

```
Per-thread stack structure at offset 0x%04x, size 0x%x bytes:
CLI buffer structure at offset 0x%04x, size 0x%x bytes:
ISAKMP key structure at offset 0x%04x, size 0x%x bytes
    Pointer:           0x%08x       %s
```

The tool's folder also contains various payloads for different encryption algorithms, and Maksym Zaitsev notes that the tool uses Internet Security

Association and Key Management Protocol (ISAKMP) for the payloads.

```
3DES_MD5_payload
3DES_SHA_payload
AES_MD5_payload
AES_SHA_payload
DES_MD5_payload
DES_SHA_payload
```