# New Release: Decrypting NetWire C2 Traffic

researchcenter.paloaltonetworks.com/2014/08/new-release-decrypting-netwire-c2-traffic/

Phil Da Silva, Rob Downs, Ryan Olson August 4, 2014

By Phil Da Silva, Rob Downs and Ryan Olson

August 4, 2014 at 9:05 AM

Category: Threat Advisories - Advisories, Unit 42

Tags: 419 Scam, NetWire, Research, Tool, Unit 42

On July 22, Palo Alto Networks threat intelligence team, Unit 42, released our first report on the evolution of “Silver Spaniel” 419 scammers. Of particular note is how these actors use a Remote Administration Tool (RAT) named NetWire (part of the NetWiredRC malware family). This RAT gives a remote attacker complete control over a Windows, Mac OS X, or Linux system through a simple graphical user interface.

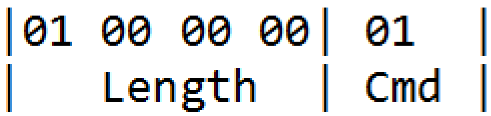
To better understand this RAT, our team reverse engineered the communication protocol that NetWire uses. Today we have released a tool that decrypts NetWire traffic and outputs any commands issued by the attacker.

## NetWire Encrytion Protocol

NetWire uses a custom, TCP-based protocol. The producer of the NetWire WorldWiredLabs, states that the tool uses 256-bit AES encryption, which we found to be accurate. The tool generates two encryption keys using a static password that the attacker chooses when creating the NetWire binary. Each packet has the following structure:

< 4 Byte Little-Endian length > < 1 Byte Command > < Data >

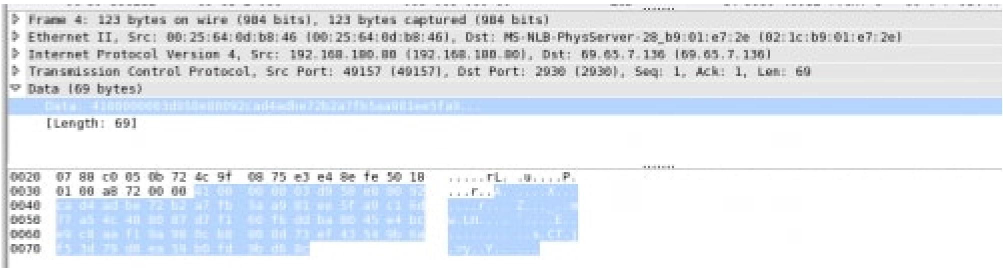
The shortest possible packet is the “HeartBeat” command, which NetWire generates every 10 seconds.



```
|01 00 00 00| 01  |
|   Length  | Cmd |
```

## Key Generation

The initial packet from the client to the server shows a data and command length of 65 bytes (0x41 listed at the beginning of the packet) with a command byte of 0x03.



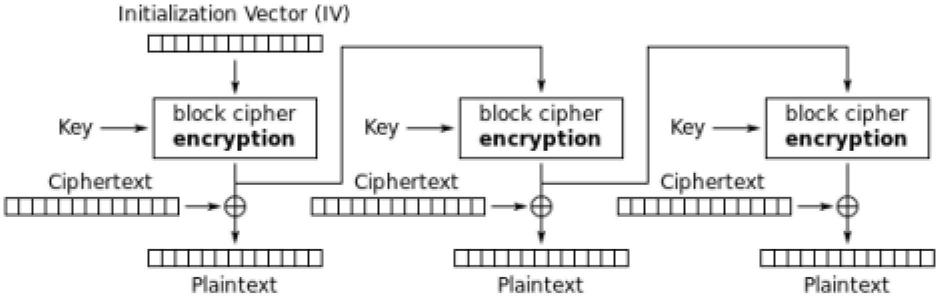
```
▷ Frame 4: 123 bytes on wire (984 bits), 123 bytes captured (984 bits)
▷ Ethernet II, Src: 00:25:64:0d:b8:46 (00:25:64:0d:b8:46), Dst: MS-NLB-PhysServer-28_b9:01:e7:2e (02:1c:b9:01:e7:2e)
▷ Internet Protocol Version 4, Src: 192.168.100.80 (192.168.100.80), Dst: 69.65.7.136 (69.65.7.136)
▷ Transmission Control Protocol, Src Port: 49157 (49157), Dst Port: 2930 (2930), Seq: 1, Ack: 1, Len: 69
▽ Data (69 bytes)
    Data: [illegible]
    [Length: 69]

0020  07 80 c0 05 0b 72 4c 9f  08 75 e3 e4 8e fe 50 18   .....rL. .u....P.
0030  01 00 a8 72 00 00 [illegible]                       ...r..[illegible]
0040  [illegible]
0050  [illegible]
0060  [illegible]
0070  [illegible]
```

Within that data is a 32-byte seed value followed by a 16-byte initialization vector (IV) value. The client then combines the 32-byte seed value with the static password in a predetermined fashion to form an AES key.

Upon receiving the initial packet, the server uses the seed value and password to generate the client's session key. It then generates its own 32-byte seed value to create it's own session key and sends the seed value to the client. The client combines this with the password and generates the same key. At this point, the key exchange is complete and both client and server hold the same two keys, which they can use to encrypt and decrypt traffic.

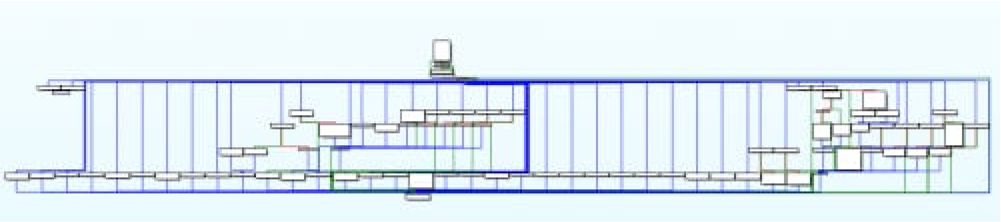
With the two keys in place, the malware uses the AES algorithm to encrypt traffic using Output Feedback (OFB) mode (Picture courtesy of Wikipedia).  The output of the block cipher encryption is eXclusive OR'ed (XOR'd) with 16 bytes of ciphertext to decrypt.  Each subsequent block of ciphertext will use the previous encrypted data as the IV passed into the block cipher encryption function.



Output Feedback (OFB) mode decryption

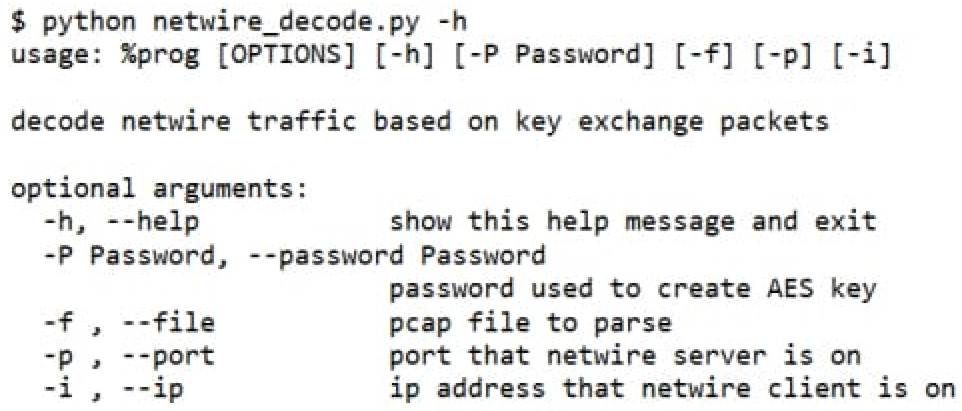
## Command Parsing

The malware has a full suite of possible commands, 76 to be exact.  Upon receipt of a command from the server, a single function is called to decrypt the payload data and execute the received instruction. The value in the command byte determines which of the commands is run through a 76 way switch statement.  A complete list of the possible commands available in NetWire was documented by CIRCL in April.



## NetWire Decoder

The NetWire decoder uses data from a packet capture file to generate the client and server session keys then decode the remaining encrypted packets. The user needs to know the IP of the infected client, the port used by malware and the encryption password to properly decode the traffic. This password is set to “Password” by default, but can be retrieved from NetWire binaries if the attacker used something more secure. The usage for the tool is show below.



```
$ python netwire_decode.py -h
usage: %prog [OPTIONS] [-h] [-P Password] [-f] [-p] [-i]

decode netwire traffic based on key exchange packets

optional arguments:
  -h, --help            show this help message and exit
  -P Password, --password Password
                        password used to create AES key
  -f , --file           pcap file to parse
  -p , --port           port that netwire server is on
  -i , --ip             ip address that netwire client is on
```

At this time the tool works against the latest version of NetWire, 1.5c. We hope this tool will be valuable to incident responders and others who are plagued by NetWire infections.