Covert Communications Through Network Configuration Messages

Ruben Rios, Jose A. Onieva, Javier Lopez

NICS Lab – University of Málaga

http://www.nics.uma.es

NICS

Agenda

- Introduction
- Motivating Scenario
- Protocol Analysis
- HIDE_DHCP implementation
- Conclusion

NICS

Introduction

- A covert channel is a form of hidden communication between processes

- Appeared in Multi-Level Security Systems
  - Storage channels
  - Timing channels

Introduction

- Network-based covert channels exploit ambiguous protocol specifications

- Some well-known storage channels
  - Covert_TCP TCP/IP
  - LOKI2, PingTunnel ICMP
  - FirePass HTTP
  - Ozyman DNS

- Any network protocol is exploitable!
  - HIDE_DHCP DHCP

NICS

Motivating Scenario

- IFIP Security Conference
  - Alice and Bob want to discuss some sensitive issues
  - Nobody can know they have been talking
    - No personal meetings
    - No encrypted communications
  - Hidden communication is necessary

NICS

Motivating Scenario

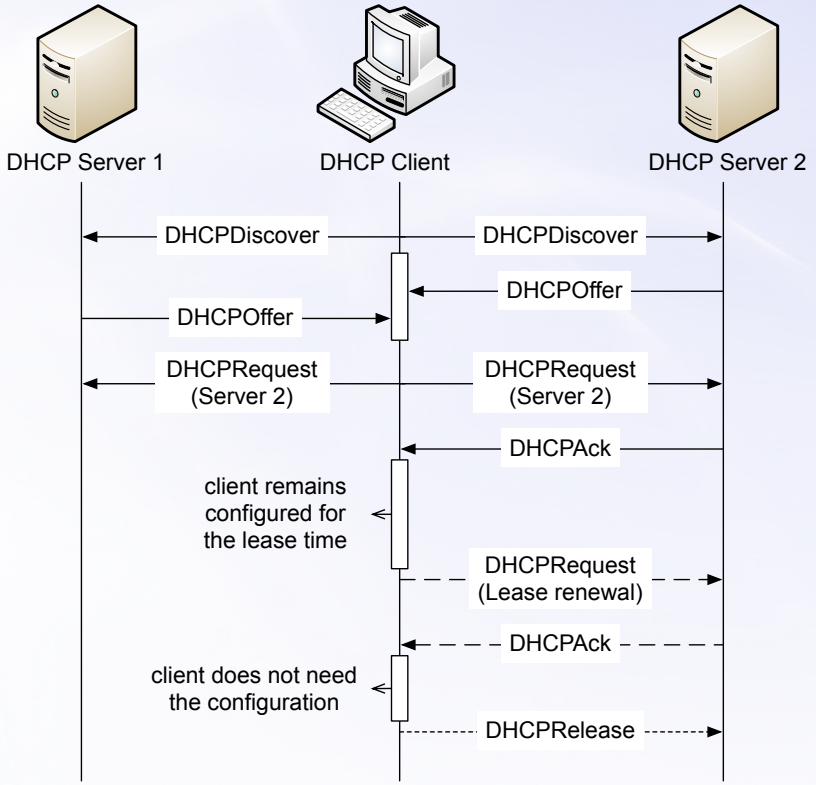
- Covert channel requirements
  - Stealthiness
  - Moderate capacity
  - Reliability
  - Locality
  - Unidirectionality

- DHCP is a suitable candidate
  - Has not been previously used for covert communications
  - Extensively deployed protocol
  - Intended for local area networks

NICS

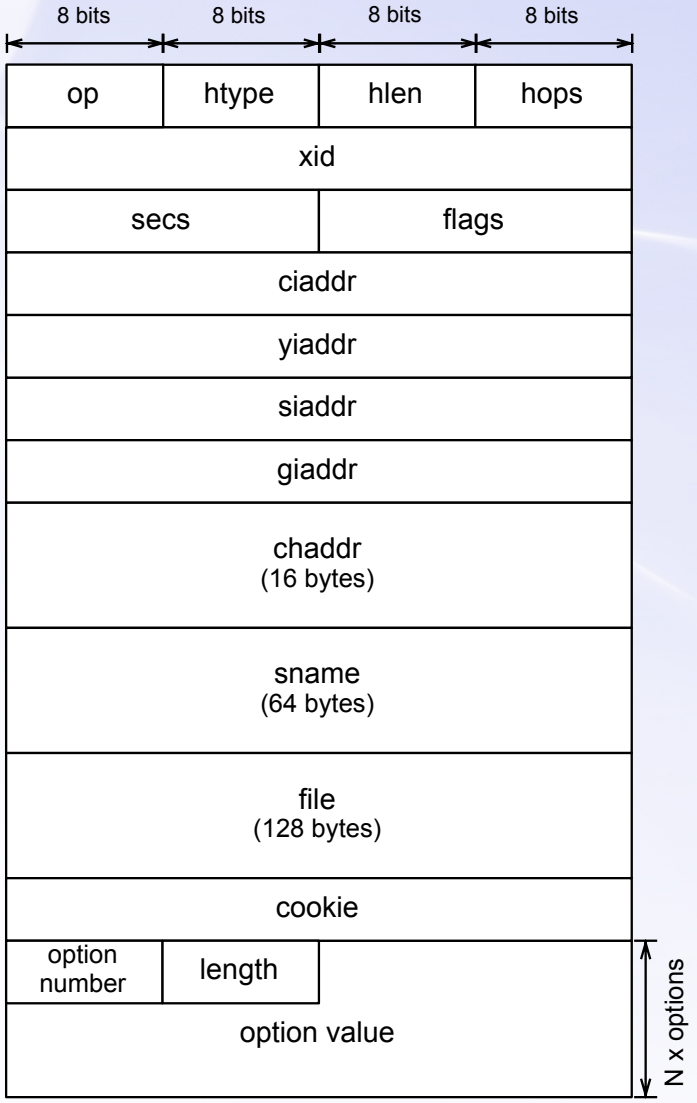
Protocol Analysis

- Dynamic Host Configuration Protocol
  - Application-layer protocol
  - UDP transport
  - Client-initiated communications
  - Transaction-based interaction
  - Two message exchange models
  - All messages has the same format



Protocol Analysis

- DHCP message format
  - Backward compatible with BOOTP
  - Messages share a common structure regardless of their type or sender
  - There are many fields and some of them are optional

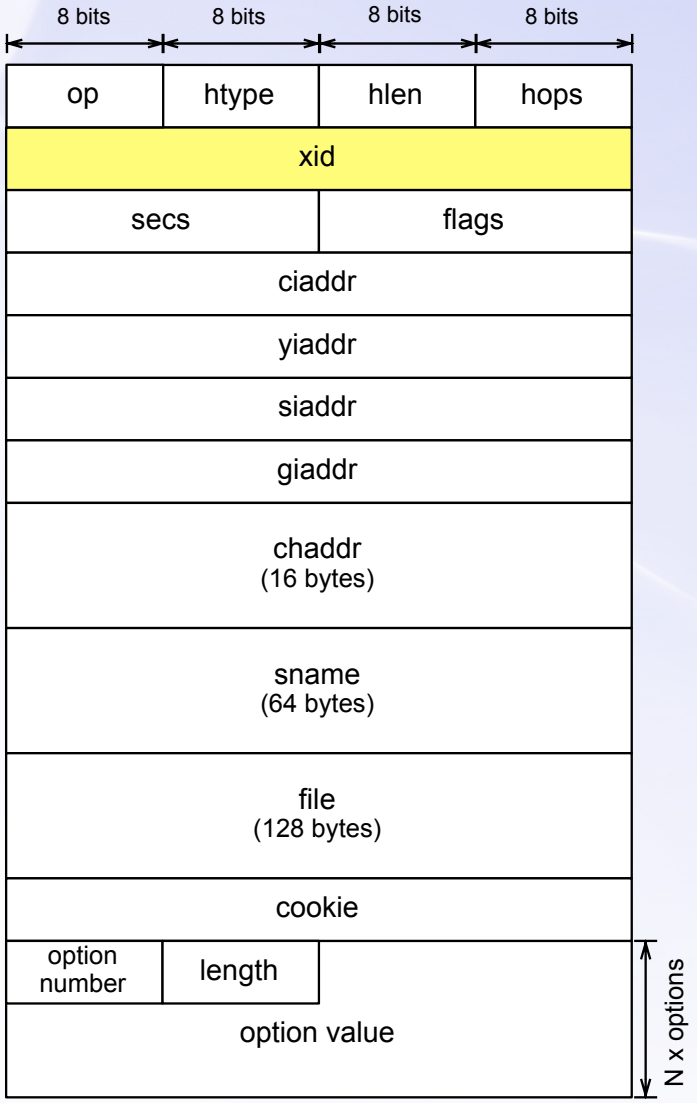


- Focus on storage channels
  - Do not alter protocol specification
  - Bandwidth vs. Detectability

NICS

Protocol Analysis

- Transaction identifier
  - Associate requests and responses
  - 4 bytes long
  - Randomly created by client!

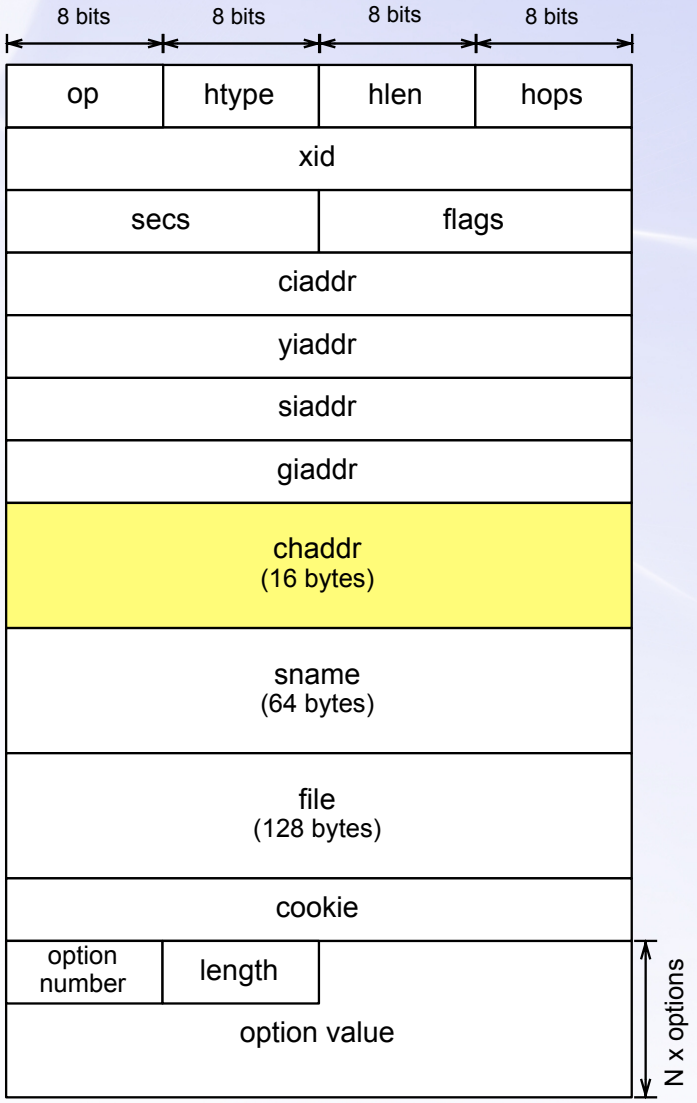


Protocol Analysis

- Transaction identifier
  - Associate requests and responses
  - 4 bytes long
  - Randomly created by client!
- Seconds
  - Elapsed time since start of transaction
  - 2 bytes long
  - Low-order bits changes

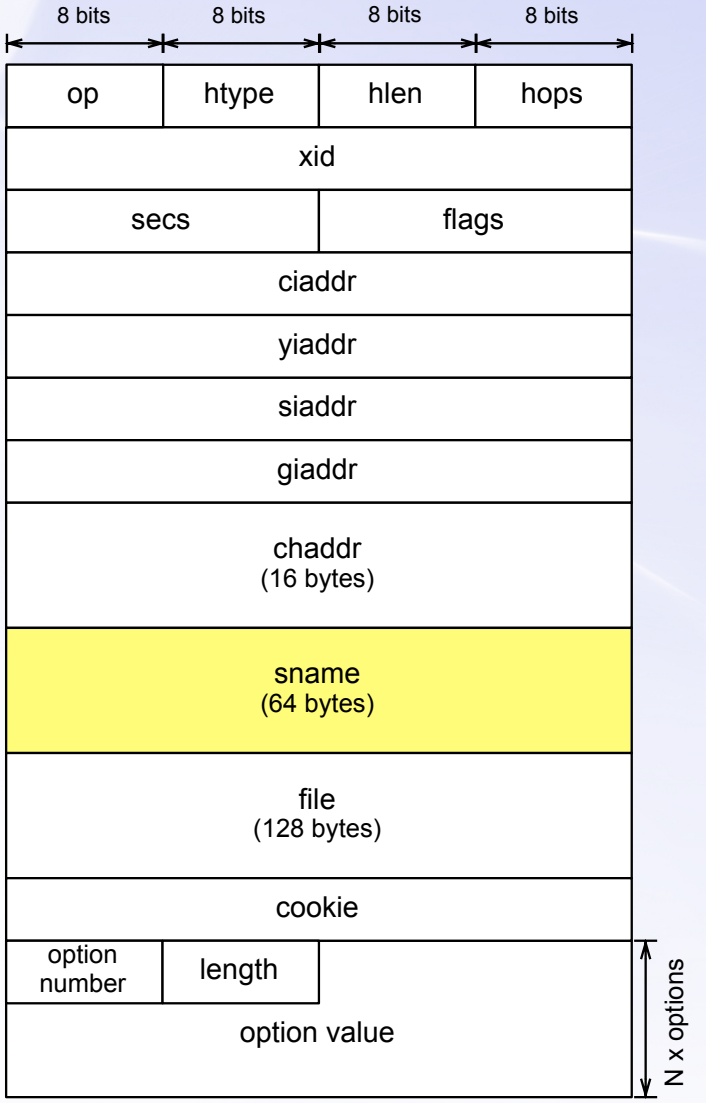
Protocol Analysis

- Client hardware address
  - Server responds to this address
  - 16 bytes long
  - Mostly used for Ethernet (6 bytes)
    - 10 bytes left for covert data
  - Bouncing DHCP Server
    - Send data to another client
    - Might be detected as a spoofing attack



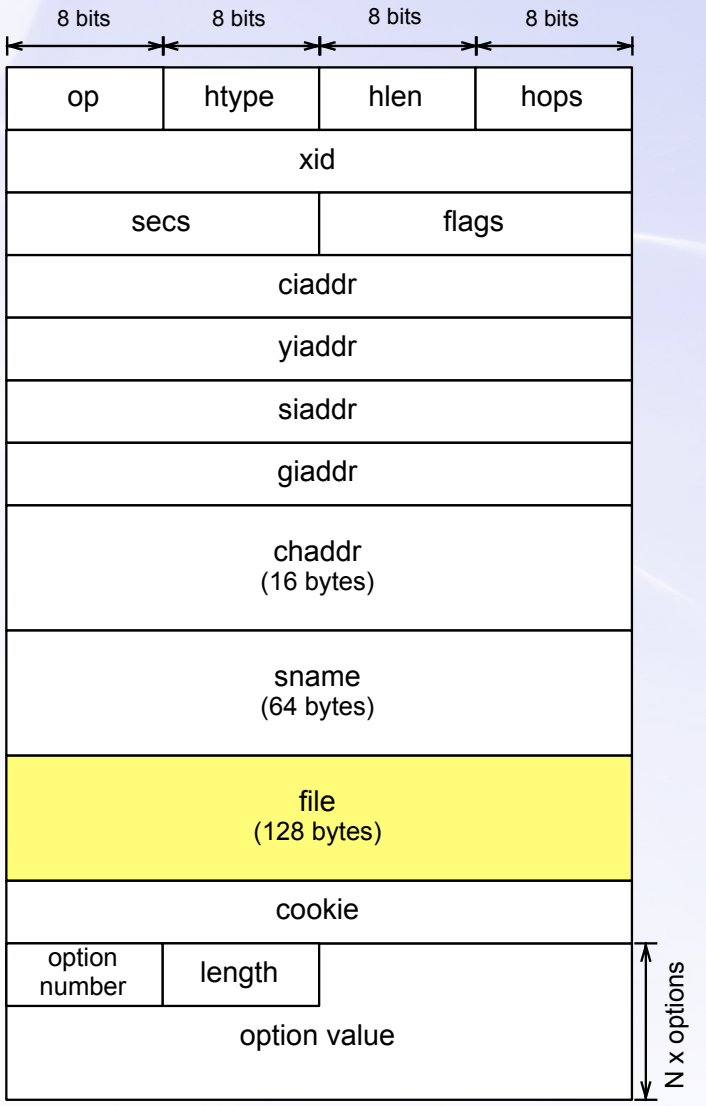
Protocol Analysis

- Server host name
  - Optionally contains the server name
  - 64 bytes long
  - Null terminated string



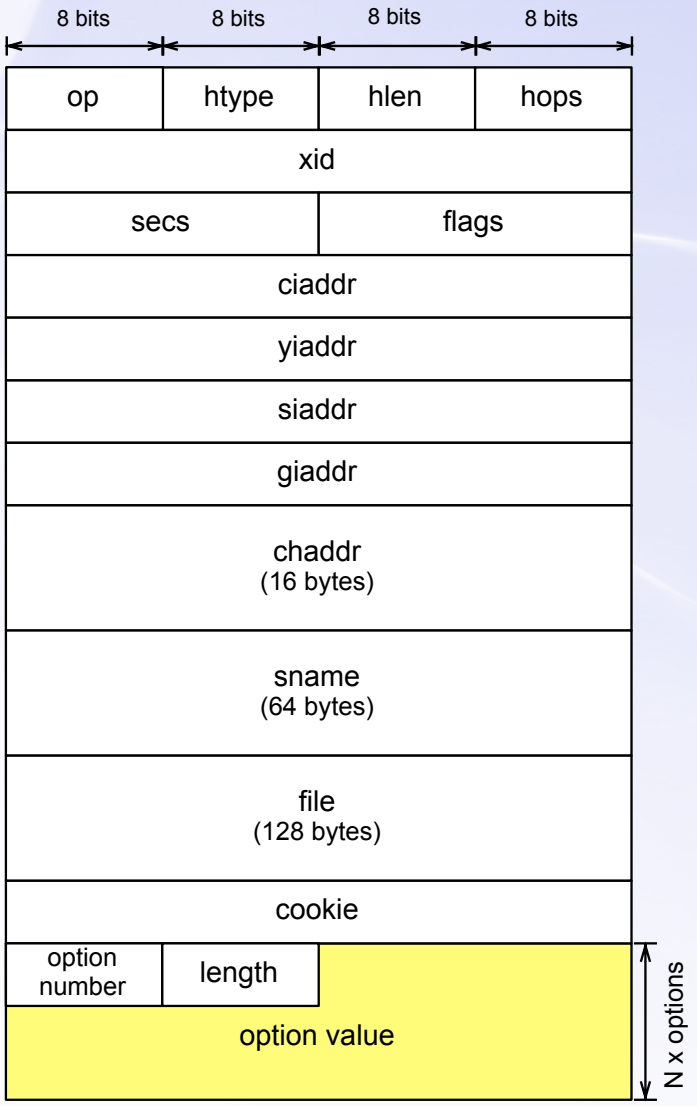
Protocol Analysis

- Server host name
  - Optionally contains the server name
  - 64 bytes long
  - Null terminated string
- Boot file name
  - Commonly used by BOOTP
  - 128 bytes long
  - Null terminated string



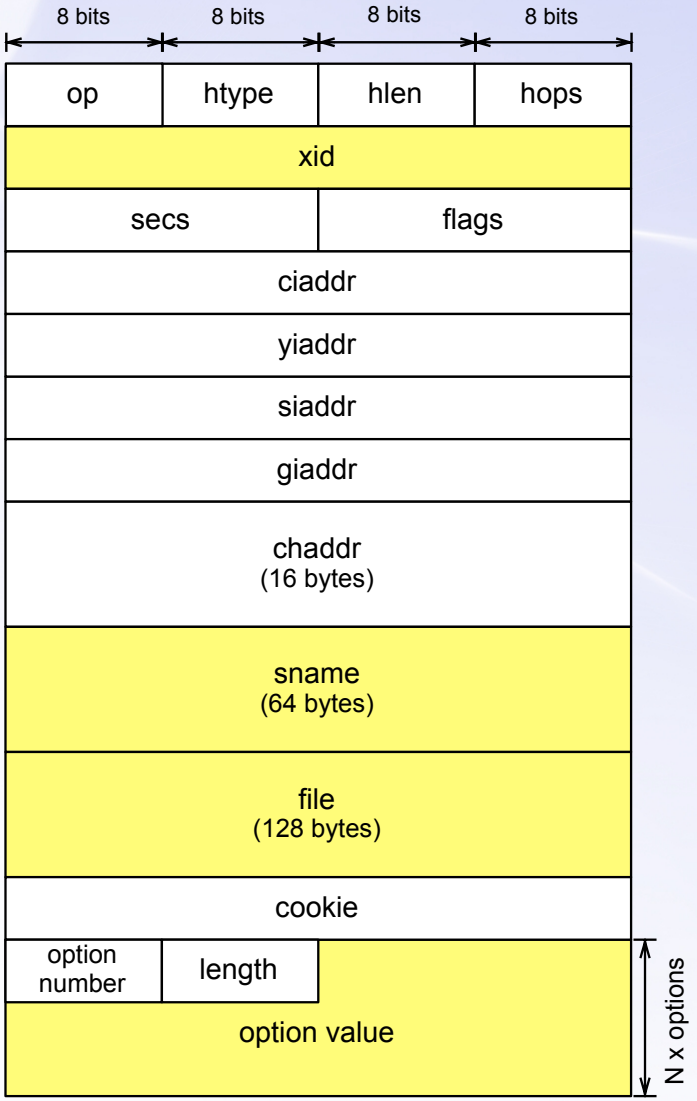
Protocol Analysis

- Options
  - Included options may depend on the type of packet
  - Variable length (up to 312 bytes)
  - Multiple covert channels
    - Number of options
    - Option number
    - Ordering of options
    - Private-use options



HIDE_DHCP Implementation

- HIDE_DHCP
  - Based on the ISC code 4.1.1-P1
  - Distributed in Linux OS
  - Integrates 3 covert channels
    - XID
    - Sname/File
    - Options
  - Fully compliant with protocol RFC

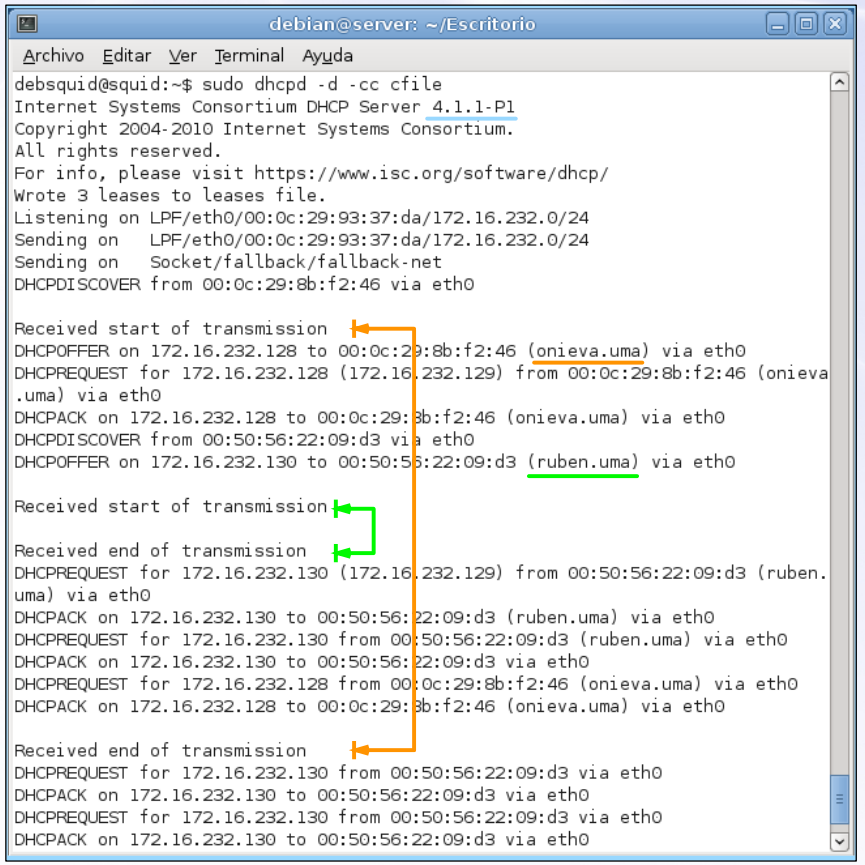


HIDE_DHCP Implementation

- Xid Implementation
  - One covert xid per transaction (4bytes)
  - Start and End delimiters to identify covert data
  - A client might detect a colluding server
  - Server retrieves covert data from DHCP Requests

HIDE_DHCP Implementation

- Sname/File Implementation
  - Pretend to be sending empty fields
  - Up to 190 bytes of covert data
  - DHCP Discover and Request as data carriers



HIDE_DHCP Implementation

- Options Implementation
  - Options for private use (#224)
  - Up to 255 bytes of covert data per packet
  - Several packets per transaction

HIDE_DHCP Analysis

- Different hiding methods present different features
  - Reliability: 100% in all cases
  - Detectability is at odds with bandwidth
    - Xid method
    - Sname/File method
    - Options method

NICS

Conclusion

- An exhaustive analysis for covert channels in DHCP
- Implemented HIDE_DHCP
  - Xid method
  - Sname/File method
  - Options method
- Future work
  - New hiding mechanisms
  - More tests on detectability and reliability
  - Countless number of vulnerable protocol exist

NICS

Thank you

Ruben Rios, Jose A. Onieva, Javier Lopez

NICS Lab – University of Málaga

http://www.nics.uma.es

NICS