Content-based Alternatives to Conventional Network Monitoring

George Louthan    Matthew Walker
Brady Deetz    John Hale

CSIIRW09 : April 15th 2009

Overview

• Network monitoring and protocol identification
  • Why do we need a new approach?
• What is content based analysis? Where does it fit in?
• Role of content-based analysis
  • Network design
  • Network intelligence
• Call to action and future work
Network Monitoring

• Why monitor network traffic?
  • Security
  • Shaping Traffic
  • Due Diligence

• Basic Analysis
  • Bandwidth/throughput - How much are people sending?
  • Flow data - Where is it going?
  • Protocol identification - What are they saying?
  • Protocol decoding - What does it mean?

Conventional Protocol Identification

• Identification by port number or other header data
  • Well-known ports
  • Advantages
    • Very easy to do
    • Decent results (if everyone’s playing nice)
  • Drawbacks
    • Protocols on alternate ports
    • Applications like Skype

• Conventional Example: Wireshark
  • Powerful packet sniffer and protocol analyzer
  • Detects protocols by well-known port numbers
  • Provides detailed information about traffic
### Wireshark

**SSH on port 22 (default)**

**Frame 72** (106 bytes on wire, 106 bytes captured)

- **Ethernet II**
  - Src: Gigabit Ethernet (09:1d:7d:65:5c:cc)
  - Dst: Cisco 64-f8:ff (00:00:00:00:00:00)

- **Internet Protocol**
  - Src: 129.244.244.159 (129.244.244.159)
  - Dst: 67.223.232.238

- **Transmission Control Protocol**
  - Src Port: 48504 (48504)
  - Dst Port: ssh (22)

- **SSH Protocol**

```plaintext
0000 00 0a e8 60 f8 ff 03 1d 7d d6 5c cc 08 09 45 09 ...d.... )..E.
0010 00 5c b0 d1 40 00 40 06 e6 71 81 f4 f4 95 43 df ...,.@. .,....C.
0020 68 ee bd 78 00 16 ba 1f 35 79 15 68 c7 13 80 18 ...x.... 5y.h....
0030 00 2e 95 56 00 00 01 01 08 00 01 3c 85 dc 35 8c .. S....
0040 10 95 53 53 48 2d 32 2e 90 2d 47 70 69 76 7a 53 63 53 ..SSH-2. 0-OpenSS
0050 48 5f 34 2e 37 70 31 31 20 44 65 62 69 6c 61 6e 2d 38 53 53 53 H.4.7p1 Debian-8
0060 75 62 75 6e 74 75 31 2e 32 0a ubuntu.2
```

---

### Wireshark

**SSH on port 5322**

**Frame 4** (105 bytes on wire, 105 bytes captured)

- **Ethernet II**
  - Src: Cisco 64-f8:ff (00:09:e8:64:f8:ff)
  - Dst: Gigabit Ethernet (00:00:00:00:00:00)

- **Internet Protocol**
  - Src: 67.223.232.238 (67.223.232.238)
  - Dst: 129.244.244.159

- **Transmission Control Protocol**
  - Src Port: 5322 (5322)
  - Dst Port: ssh (22)

- **Data** (39 bytes)

```plaintext
0000 00 1d 7d d6 5c cc 00 09 e8 64 f8 ff 08 00 45 00 ...}.... .d....E.
0010 00 5b 07 50 00 00 34 06 9b f4 43 df e8 ee 81 f4 [.Pe.4. ...C....
0020 f4 96 14 ca 89 93 14 ad 0b b0 ba 77 2d 1e 80 1e ............w$...
0030 02 64 b9 d5 00 00 01 01 08 00 0a 35 8c 0a e0 01 3c ............S..<
0040 82 e3 53 53 48 2d 32 2e 30 2f 47 70 69 76 7a 53 53 ..SSH-2. U-OpenSS
0050 48 5f 35 33 2e 32 31 30 20 44 65 62 69 6c 61 6e 2d 33 53 53 53 H.5.3p1 Debian-3
0060 75 62 75 6e 74 75 31 31 2e 32 0a ubuntu.1
```
What is content-based network analysis?

- Making decisions based upon packet payloads, not headers
- Applied to protocol identification
  - Recognize the type of traffic detected
  - Determine the application using the traffic
  - Understand the purpose and role of that traffic
- Methods
  - Signature based
  - Statistical
  - Artificial Intelligence
  - Hybrid
Network Design

• Some initial questions
  • What traffic is actually flowing on my network?
  • Where is it coming from and going to?
  • What is in the traffic? (read: deep packet inspection)
  • How are we going to deal with the traffic?
    • Firewall, QoS, Packet Shaping, and Load Balancing Implementations

• And some opensource answers
  • Linux Netfilter!
  • L7-filter
  • ipp2p

Basic Netfilter & Conntrack

• Netfilter is the Linux firewall and filtering framework
  • Powerful, but not content based

• There exists a set of tools for offline usage: Conntrack
  • Allows interaction with the in-kernel Connection Tracking System
  • Example usage
    ```
    root@dijkstra:~# conntrack -L
    tcp 6 431950 ESTABLISHED src=129.244.244.170 dst=74.125.95.103 sport=80 dport=80 packets=4 bytes=1545
    src=74.125.95.103 dst=129.244.244.170 sport=80 dport=1845 packets=4 bytes=504 [ASSURED] use=1
    ```
  • Allows monitoring scripts, perhaps even script based firewalls
**L7-filter**

- **L7-filter** is a classifier for Linux's *Netfilter* that identifies packets based on application layer data
- Classifies on OSI layers 5-7
- Using MARK it can enable traffic QoS, packet shaping, anything really
  
  ```
  root@l7test:~# iptables -t mangle -A POSTROUTING -m layer7 --l7proto imap -j MARK --set-mark 3
  root@l7test:~# tc filter add dev eth0 protocol ip parent 1:0 prio 1 handle 3 fw flowid 1:3
  ```
- It can even just do simple accounting
  
  ```
  root@l7test:~# iptables -t mangle -A POSTROUTING -m layer7 --l7proto imap
  ```

**ipp2p**

- Same principle as **L7-filter**, except specific to P2P protocols
  - Looks at packet contents for P2P commands
- Very basic example for dropping BitTorrent
  
  ```
  iptables -A FORWARD -m ipp2p --ipp2p --bit -j DROP
  ```
- Once again, if you just want to know if it's on your network
  
  ```
  iptables -t mangle -A PREROUTING -j CONNMARK --restore-mark
  iptables -t mangle -A PREROUTING -m mark ! --mark 0 -j ACCEPT
  iptables -t mangle -A PREROUTING -m ipp2p --edk -j MARK --set-mark 1
  iptables -t mangle -A PREROUTING -j CONNMARK --save-mark
  iptables -t mangle -A POSTROUTING -m mark --mark 1 -j ACCEPT
  iptables -t mangle -L -n -v -x
  iptables -A FORWARD -m ipp2p --ipp2p --bit -j DROP
  ```
Content Aware Network Mapping

- An accurate network map can be produced using content aware mapping tools.
- Tools
  - Cheops
  - Nagios

Service Detection

- Targeted host-based service detection
- Service “fingerprinting” to identify particular protocols, implementations, versions, and platforms
  - amap
  - nmap
Fingerprinting with *amap*

```bash
#bdeetz@metheny:~$ amap -B localhost 993
Banner on 127.0.0.1:993/tcp : SSH-2.0-OpenSSH_5.1p1 Debian-3ubuntu1
amap v5.2 finished at 2009-04-09 22:27:58

#bdeetz@metheny:~$ amap -A localhost 993
amap v5.2 (www.thc.org/thc-amap) started at 2009-04-09 22:28:17 - MAPPING mode
Protocol on 127.0.0.1:993/tcp matches ssh
Protocol on 127.0.0.1:993/tcp matches ssh-openssh
Unidentified ports: none.
amap v5.2 finished at 2009-04-09 22:28:17
```

Similar features in *nmap*

```bash
root@metheny:~# nmap -A localhost -p 993
Starting Nmap 4.62 ( http://nmap.org ) at 2009-04-09 23:16 CDT
Interesting ports on localhost (127.0.0.1):
  PORT      STATE       SERVICE
  993/tcp    open        ssh          (protocol 2.0)
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.23
Uptime: 0.283 days (since Thu Apr  9 16:28:35 2009)
Network Distance: 0 hops
```
• Content aware traffic analysis provides a more accurate method of security and data assurance

• Tools
  • L7-filter
  • Network Processors and FPGAs (IXP2855, for instance)

• Catching SSH Protocol via RegEx:
  \^ssh-[\[2]\[0-9\]

Call to Action

• The question:
  • How can we focus our future efforts to maximize the success of our strategy to ensure our technologies can meet the challenge of cyber security?

• The answer
  • Our computational power has grown exponentially over time, but our network protocol analysis has remained quite simplistic.
  • Modern networks demand modern solutions that make decisions based upon highly adaptive threats

• Focus our efforts on deeper, more sophisticated network analysis
Future Work

- The least you can do
  - Run `nmap` and `amap` and submit undetected signatures
  - Build the body of knowledge regarding pesky protocols
    - Protocol Wiki: http://www.protocolinfo.org
    - http://www.pcapr.net

- Development work (just write the glue)
  - Wireshark-style sniffer with content-based protocol identification
  - More mature firewalls - content-based shorewall equivalent?

Future Work

- Research
  - Truly application-layer firewalls
  - FPGA or HPC methods
  - EBNF and compiler-like methods
  - Stream-based analysis
  - Novel identification strategies
    - Bloom Filters?