XPROBE

Building Efficient Network Discovery Tools

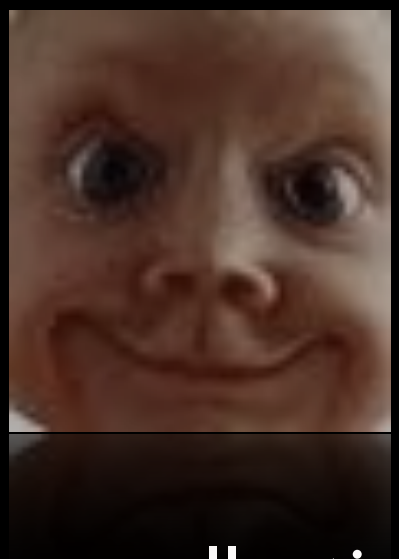
Fyodor Yarochkin

Outline

- Introduction
- Some motivating stories: real-life attacks
- Efficient network mapping with “Lazy Scan” mode
- Layer 7 extensions
- Scripting Extensions
- Data Mining and Experimental Data sharing network

Introducing presenter

- Fyodor.Y



- Interests:
  - Intelligence collection/analysis
  - Network discovery and network protocols
  - AI

Attack Trends

China vs. Taiwan

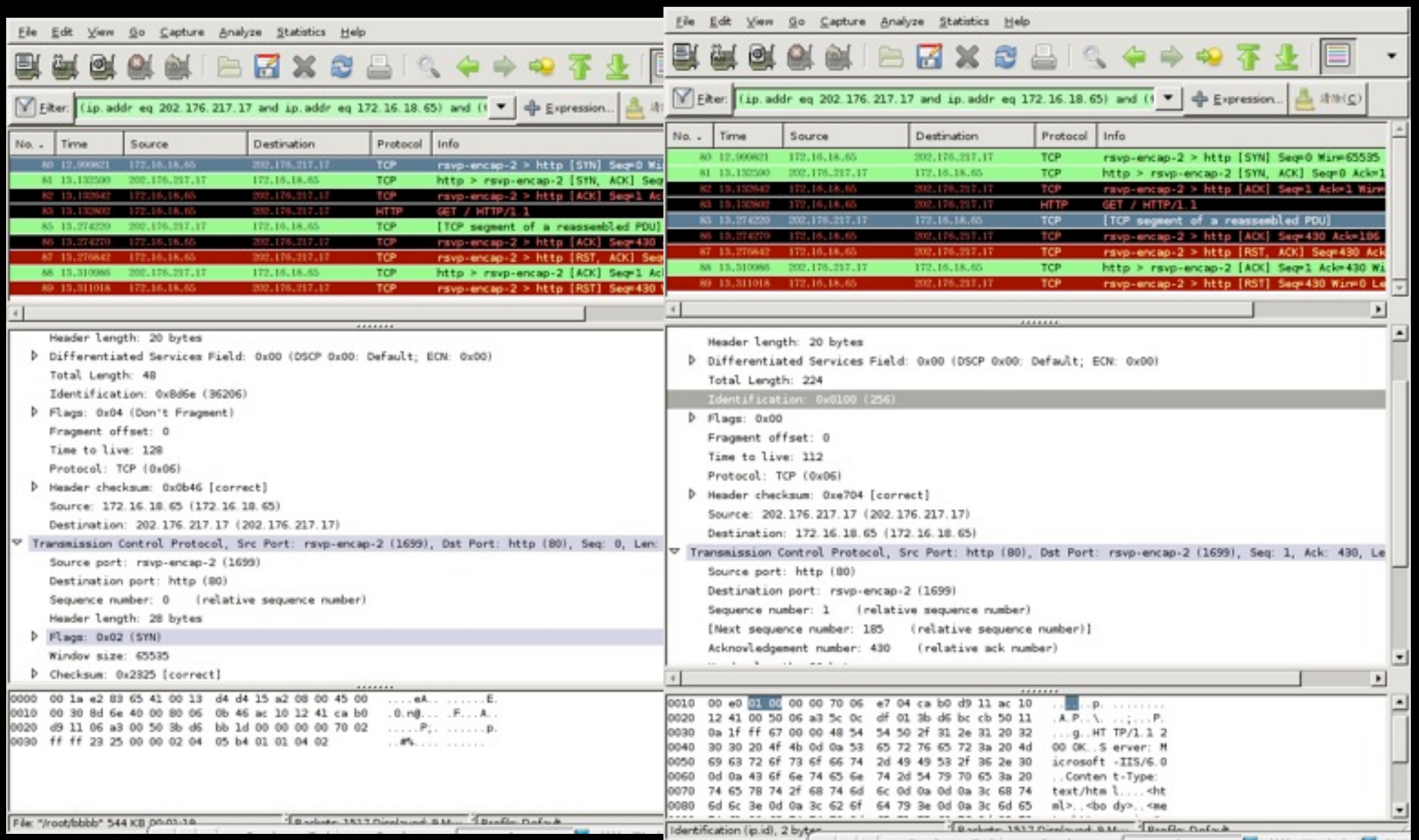
briefs of cyber "wars"

Mystic redirects (2009/03/05)

Attack observations

- Large number of users were redirected to malware-infected servers, while trying to visit legimate web sites hosted outside of Taiwan island (i.e. zdnet, msn.com, etc)

Traces


File Edit View Go Capture Analyze Statistics Help
Filter: (ip.addr eq 202.176.217.17 and ip.addr eq 172.16.18.65) and (
Expression...
No. Time Source Destination Protocol Info
80 12.000821 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [SYN] Seq=0 Wi
81 13.132500 202.176.217.17 172.16.18.65 TCP http > rsvp-encap-2 [SYN, ACK] Seq
82 13.132642 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [ACK] Seq=1 Ac
83 13.132802 172.16.18.65 202.176.217.17 HTTP GET / HTTP/1.1
85 13.274220 202.176.217.17 172.16.18.65 TCP [TCP segment of a reassembled PDU]
86 13.274270 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [ACK] Seq=430
87 13.276842 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [RST, ACK] Seq
88 13.310086 202.176.217.17 172.16.18.65 TCP http > rsvp-encap-2 [ACK] Seq=1 Ac
89 13.311016 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [RST] Seq=430
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 48
Identification: 0x8d6e (36206)
Flags: 0x04 (Don't Fragment)
Fragment offset: 0
Time to live: 128
Protocol: TCP (0x06)
Header checksum: 0x0b46 [correct]
Source: 172.16.18.65 (172.16.18.65)
Destination: 202.176.217.17 (202.176.217.17)
Transmission Control Protocol, Src Port: rsvp-encap-2 (1699), Dst Port: http (80), Seq: 0, Len:
Source port: rsvp-encap-2 (1699)
Destination port: http (80)
Sequence number: 0 (relative sequence number)
Header length: 28 bytes
Flags: 0x02 (SYN)
Window size: 65535
Checksum: 0x2325 [correct]
0000 00 1a e2 83 65 41 00 13 d4 d4 15 a2 08 00 45 00 ....eA.. ......E.
0010 00 30 8d 6e 40 00 80 06 0b 46 ac 10 12 41 ca b0 .0.n@... .F...A..
0020 d9 11 06 a3 00 50 3b d6 bb 1d 00 00 00 00 70 02 .....P;. ......p.
0030 ff ff 23 25 00 00 02 04 05 b4 01 01 04 02 ..#%.... ......
File: "/root/bbbb" 544 KB
File Edit View Go Capture Analyze Statistics Help
Filter: (ip.addr eq 202.176.217.17 and ip.addr eq 172.16.18.65) and (
Expression...
No. Time Source Destination Protocol Info
80 12.000821 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [SYN] Seq=0 Win=65535
81 13.132500 202.176.217.17 172.16.18.65 TCP http > rsvp-encap-2 [SYN, ACK] Seq=0 Ack=1
82 13.132642 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [ACK] Seq=1 Ack=1 Win=
83 13.132802 172.16.18.65 202.176.217.17 HTTP GET / HTTP/1.1
85 13.274220 202.176.217.17 172.16.18.65 TCP [TCP segment of a reassembled PDU]
86 13.274270 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [ACK] Seq=430 Ack=186
87 13.276842 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [RST, ACK] Seq=430 Ack
88 13.310086 202.176.217.17 172.16.18.65 TCP http > rsvp-encap-2 [ACK] Seq=1 Ack=430 Wi
89 13.311016 172.16.18.65 202.176.217.17 TCP rsvp-encap-2 > http [RST] Seq=430 Win=0 Le
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 224
Identification: 0x0100 (256)
Flags: 0x00
Fragment offset: 0
Time to live: 112
Protocol: TCP (0x06)
Header checksum: 0xe704 [correct]
Source: 202.176.217.17 (202.176.217.17)
Destination: 172.16.18.65 (172.16.18.65)
Transmission Control Protocol, Src Port: http (80), Dst Port: rsvp-encap-2 (1699), Seq: 1, Ack: 430, Le
Source port: http (80)
Destination port: rsvp-encap-2 (1699)
Sequence number: 1 (relative sequence number)
[Next sequence number: 185 (relative sequence number)]
Acknowledgement number: 430 (relative ack number)
0010 00 e0 01 00 00 00 70 06 e7 04 ca b0 d9 11 ac 10 ......p. ....
0020 12 41 00 50 06 a3 5c 0c df 01 3b d6 bc cb 50 11 .A.P..\. ..;...P.
0030 0a 1f ff 67 00 00 48 54 54 50 2f 31 2e 31 20 32 ...g..HT TP/1.1 2
0040 30 30 20 4f 4b 0d 0a 53 65 72 76 65 72 3a 20 4d 00 OK..S erver: M
0050 69 63 72 6f 73 6f 66 74 2d 49 49 53 2f 36 2e 30 icrosoft -IIS/6.0
0060 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 ..Conten t-Type:
0070 74 65 78 74 2f 68 74 6d 6c 0d 0a 0d 0a 3c 68 74 text/htm l....<ht
0080 6d 6c 3e 0d 0a 3c 62 6f 64 79 3e 0d 0a 3c 6d 65 ml>..<bo dy>..<me
Identification (ip.id), 2 byte

Guess..

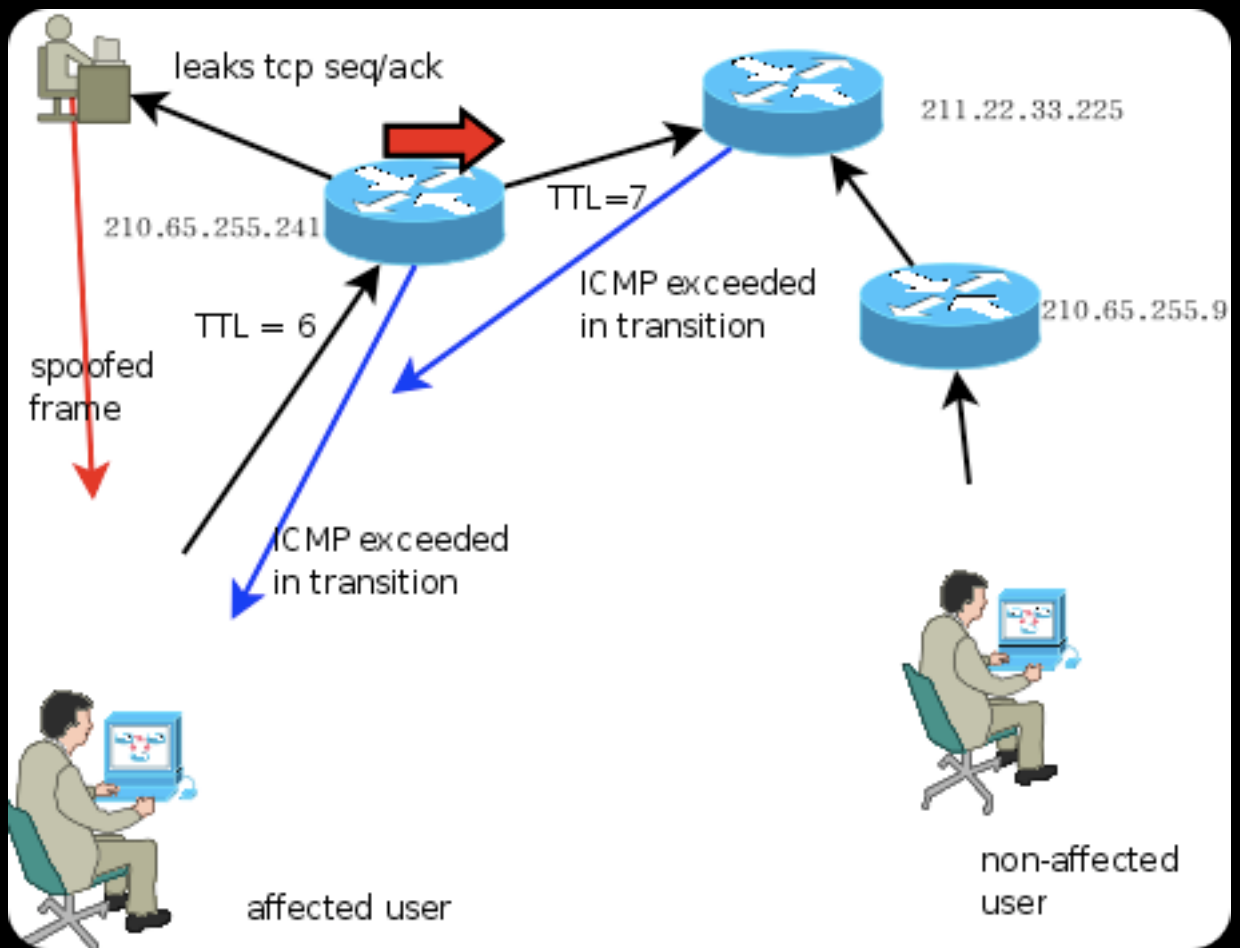
- A node was compromised somewhere en-route. TCP connections were non-blindly hijacked...

Tracing “ghost” node(s)

- some “spaghetti” to quickly discover the node

```
Tracing the path to www.orzteam.com (58.222.16.55) on TCP port 80 (http
s max, 791 byte packets
 2  114.45.208.254  157.892 ms  150.266 ms  151.822 ms
 3  168.95.71.62  151.827 ms  152.767 ms  166.531 ms
 4  220.128.4.118  155.682 ms  152.328 ms  151.788 ms
 5  * * *
 6  210.65.255.241  154.322 ms  160.305 ms  151.788 ms
 7  211.22.33.225  211.852 ms
    58.222.16.55 [unknown, ACK FIN]  109.508 ms
    211.22.33.225  315.486 ms
```

Discovered attack scenario



Lesson learnt

- Large number of target nodes are to be probed in order to identify potential ‘en-route’ attacks.
- We need a high-performance network discovery tool, capable of operating at Layer7
- we need automated tracing capability

more stuff @L7...

```
morozec ~ $ nc www.ebay.com 80
CONNECT 61.222.2.251:22 HTTP/1.0

HTTP/1.0 200 Connection established
Proxy-agent: CacheFlow-Proxy/1.0

SSH-2.0-OpenSSH_4.3
```

```
(echo -e "CONNECT 192.168.8
 Connection established
 CacheFlow-Proxy/1.0
```

```
Authorised access only

This system is the property of H
```

Motivation

- we need more application-level probes

And..

- we could actually correlate L7 data with network probing results

but ..

- we need to minimize network load, because L7 might mean “lots of traffic”

Also..

- Time is another player. We want to be able to monitor network fluctuations in time

So, the Xprobe

now “NG”

Xprobe

- The historical note:
  - Xprobe project started as remote fingngerprinting tool to probe remote systems using ICMP protocol queries.
  - Other protocols support was added later. Fuzzy fingerprinting mechanism was introduced to improve precision

Further motivation

- Exploring other protocols running on the top of IP
- Bulk scanning
- Probing “en-route” systems
- Migrating to IPv6
- Honeypots/Nets
- Improving precision by cross-correlation over time

On the top of IP

- SCTP/Sigtrans gateways
- IPv4 to IPv6 gateways
- ...

“en route” findings

- Caching systems, transparent proxies etc.
- L7 switches
- Reactive IDS/IPS
- Application Firewalls
- Active spoofing attacks ..

Honeypots

- Virtual Machines
- Virtual Networks
- Incomplete Services

Bulk Scanning

- Probing “en-route” devices by large-range scans
- IPv6

Data cross-correlation

- Currently correlating data between L7 and network layers.

Current Improvements

Minimizing Network Load

- Information Gain metrics
- “Lazy-Mode” execution
- “Target” driven execution
- New Scan engine (in progress)

Improving Precision

- Cross correlation between L7 and below

Improving Usability

- Language Extensions: Python (xprobepy)

Information Gain

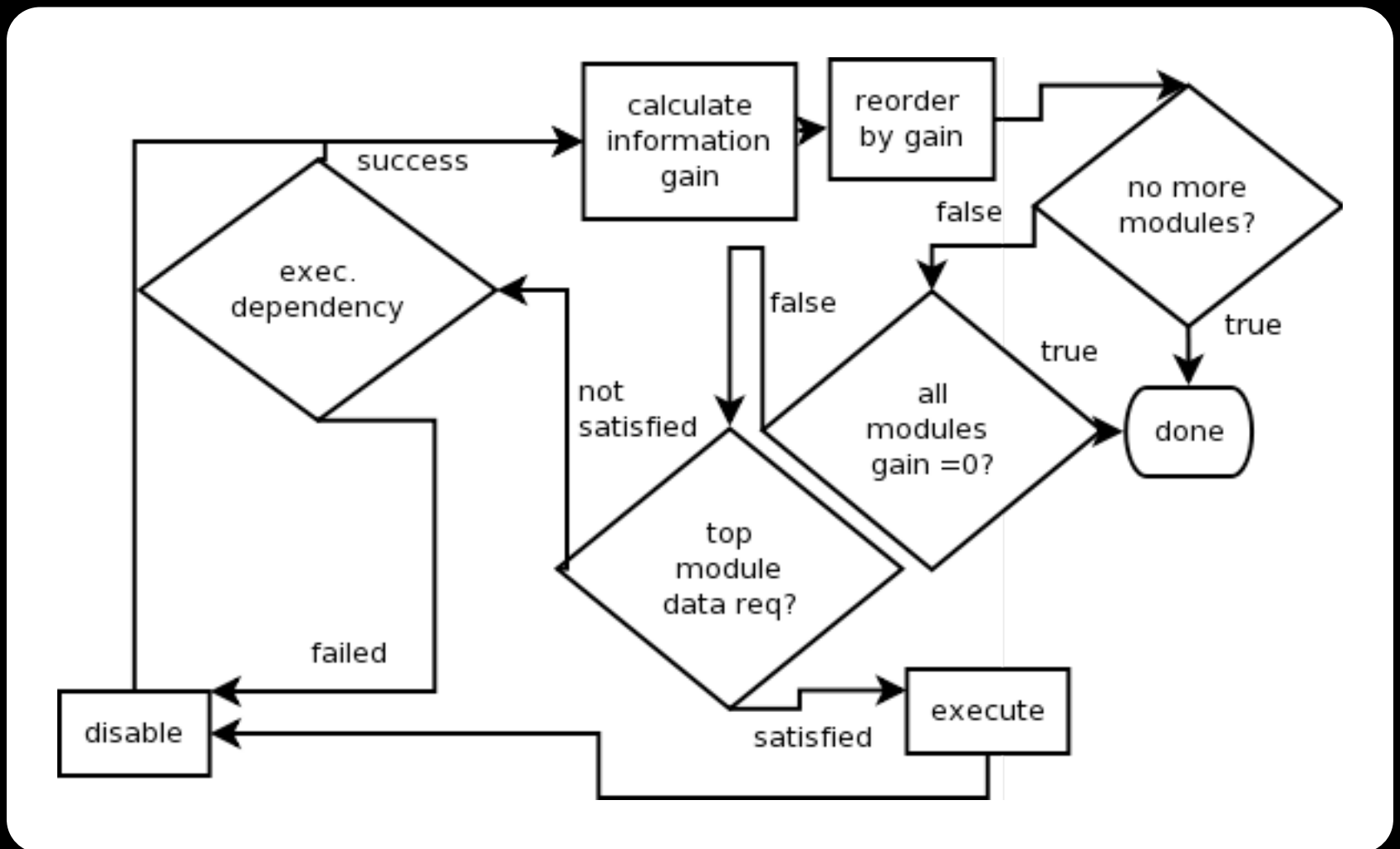
Information gain

- A "score" calculated for a probe, characterizing how much "information" the probe is going to bring

Benefits

- Highest information gain probes are executed first
- “0” information gain probes are not executed (unless are part of dependency)
- Possible to optimally minimize network overhead by executing “top X”/target

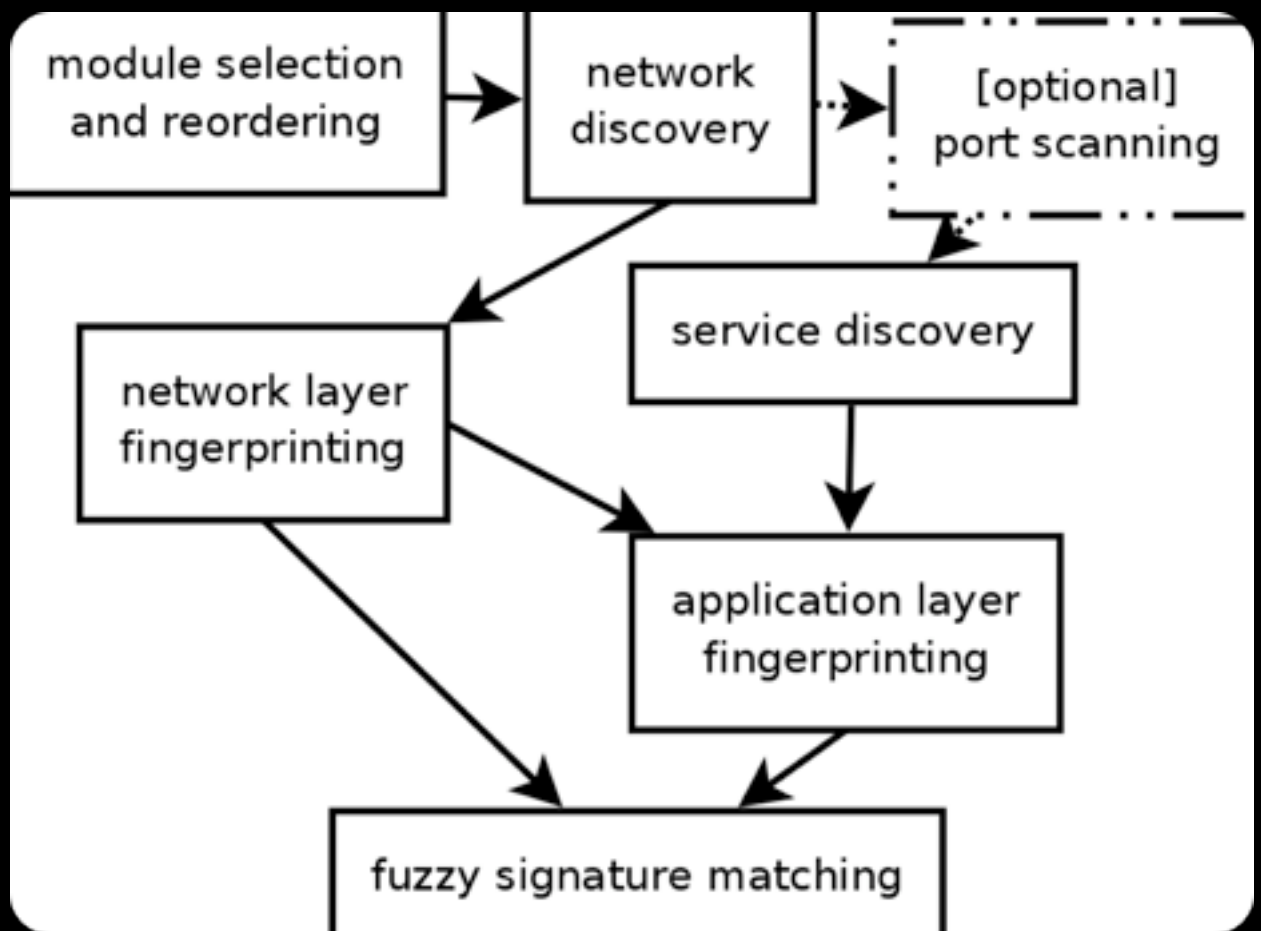
Algorithm


calculate information gain
reorder by gain
no more modules?
false
true
success
exec. dependency
false
not satisfied
all modules gain =0?
true
done
top module data req?
failed
disable
satisfied
execute

Lazy scan and target-driven execution

discovery process optimizations

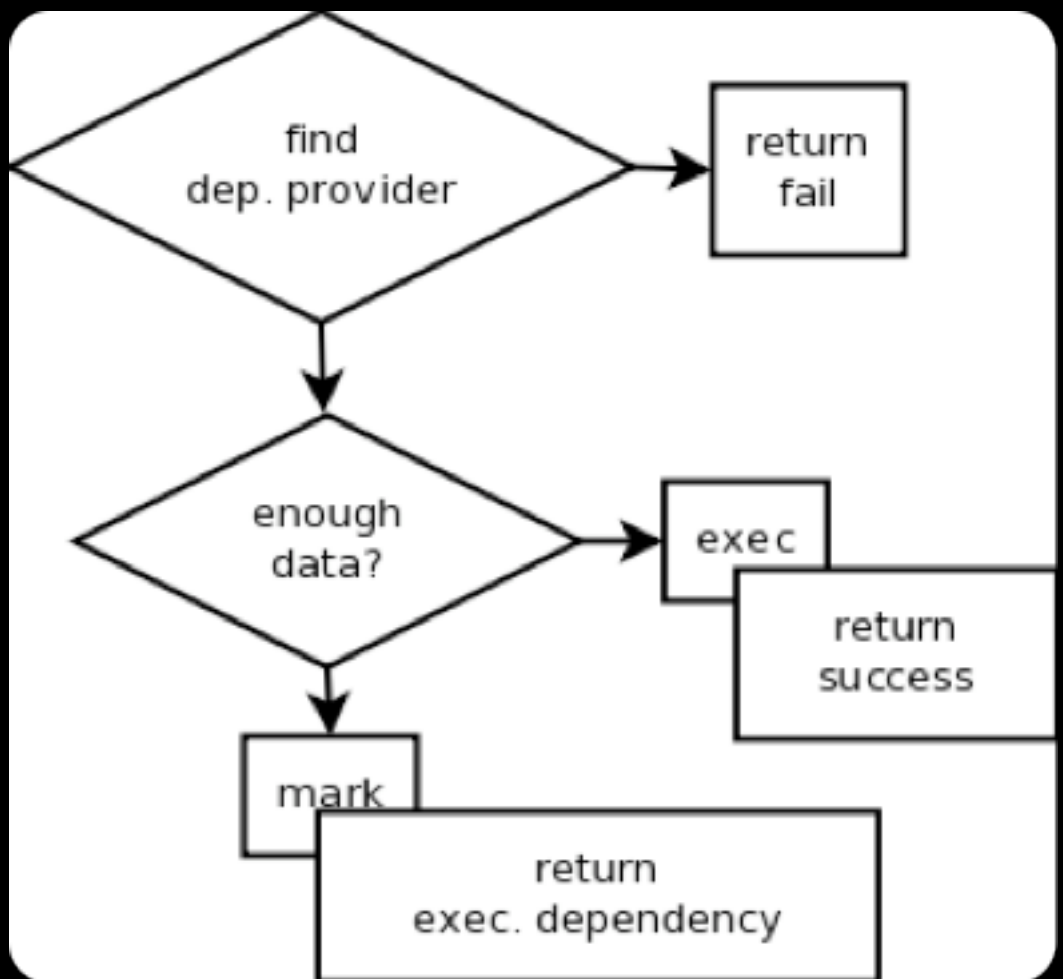
Architecture, briefly..



Data dependency chains

- Each module is characterized with type of data it "requires" and "provides"

Data Dependency based execution



No “portscan” per se

- This technically makes port scanning “AS IS” unnecessary
- Significally reduces tool “noise” on the wire

Wire “noise” rough comparision

Benefits of Data Chaining

- Probe focused execution (by specifying "intended" probe)
- Restrictions can be set:
  - no more than X queries/target
  - use only "normalized" packets

Negative impact

- You still may not know about certain ports and applications running on the target system.

Application level

Application level

- Improving fingerprinting precision
- “en-route” interaction
- Honeypots

L7 fingerprinting

- Underlying OS can be probed via L7 tests and correlated with other data

Test type	Usable Protocol	Test
Directory Separator	HTTP	Win/Unx
New line characters	HTTP	Win/Unx
Special/reserved filenames	HTTP	Win/Unx
Root directory	FTP	Win/Unx..
Special characters (EOF,EOL		
Filesystem limitations	HTTP, FTP	..
Filesystem illegal characters	HTTP, FTP	..
Case sensitivity	HTTP, FTP	Win/Unx
Special filenames handling	HTTP, FTP	Win/Unx
Special files in directory	HTTP, FTP	Win/Unx
Binary file fingerprinting	FTP	Win/Unx

Honeypots

VM tricks

- Possible to identify VMs (not all) by TCP stream analysis

Network level tricks

- Analyzing MAC addresses, when available

Application Level Tricks

- We can probe for incomplete implementations of L7 protocols

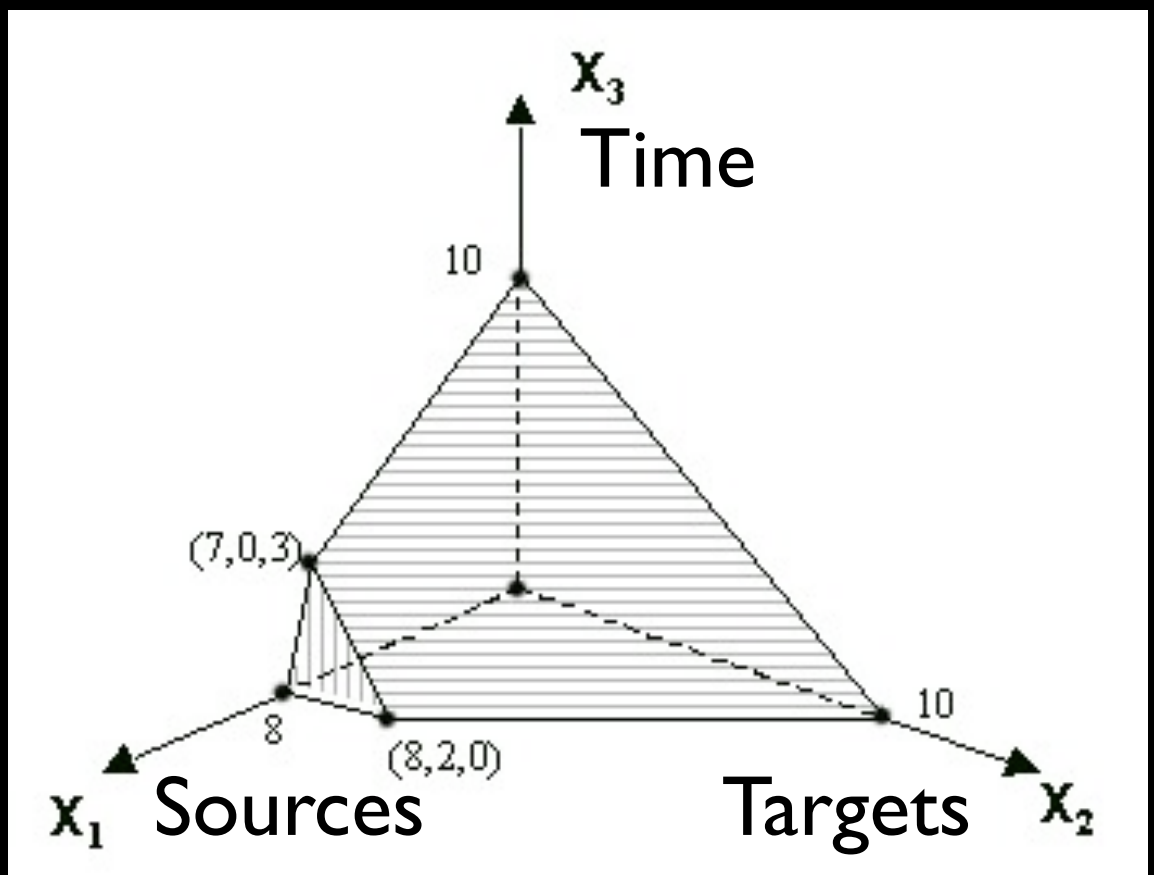
Current Developments

Work in progress

- Language bindings
- L7 modules
- new engine
- en-route modules

Future Plans

- By designing distributed data sharing network it'd be possible to collect Multi-dimensional data



IPv6 Action plan

- Local node discovery: straightforward (multicast)
- Remote segments: DNS, text file parsing, “educated” guessing, search engine, beforementioned networking capability

Availability

http://xprobe.sourceforge.net
(git push in a couple of days)
http://github.com/fygrave/xprobepy
(due Mid of July)

Questions

if you have no questions, feel free to throw your shoe ;-)
jk

fygrave@o0o.nu
(o-zero-o)