

Spoofing Nmap Service Detection

@alex.kropivny

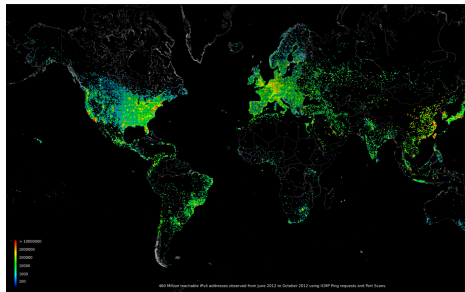
March 11, 2014

Generating fake Nmap results for:

- ① Fun
- ② A possible countermeasure to data collection
- ③ Gives better understand of security tools

Privacy is Dead

- What data is collected?
- How is it used?
- Who handles it?
- ???????????



Possible Stopgap

- If it's not important enough for manual analysis...
- And if automated analysis sucks (it usually does)...
- Then a "Big Data generator" may mask our signals



Travis Goodspeed
@travisgoodspeed

Follow

The Pastor Laphroaig reminds us of our only obligation when presented with a marketing survey: to lie in a way which is not easily filtered.

Reply Retweet Favorite More

RETWEETS

5

FAVORITES

4



5:15 AM - 10 Sep 2012

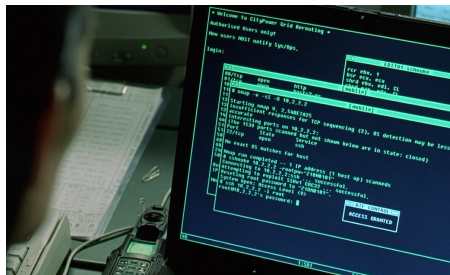
Know the Limits

- What assumptions were tools built under?
- Where do they start to fail?
- How does an adversarial scenario play out?

Target Selection

Nmap was chosen since:

- It's well known enough to be the first tool people reach for
- It's complex enough to have interesting bits of functionality



3 Stage Scan Process

- Port discovery
(open/closed/filtered)
- Service detection
(http/ftp/snmp/... + versions)
- NSE scripts (deep protocol details)

```
C:\Users\antall>nmap -Pn github.com -p 443 -A
Starting Nmap 6.40 ( http://nmap.org ) at 2014-03-11 00
Nmap scan report for github.com (192.30.252.131)
Host is up (0.083s latency)
DNS record for 192.30.252.131: ipid-1b3-prd.iad.github
PORT      STATE SERVICE
443/tcp    open  ssl/https?
|_ http-methods: No Allow or Public header in OPTIONS res
|_ http-robots.txt: 41 disallowed entries (15 shown)
|_ /ekansa/Open-Context-Data /ekansa/nopencontext-*
|_ /*/*/*pull/*/*/*tree/*/*/*blob/*/*/*wiki/*/*/*gist
|_ /gist/*/*download /gist/*/*/*/*/*/*issue/new /*/*is
|_ /*/*commits/*/*/*/*commits/*/*author /*/*commits/*
|_ /*/*branches
|_ http-title: GitHub \xC2\xB7 Build software better. to
|_ ssl-cert: Subject: commonName=github.com/organization
ProvinceName=California/countryName=US
|_ Not valid before: 2013-06-02T23:00:00+00:00
|_ Not valid after: 2015-03-02T11:00:00+00:00
|_ _ssl-date: 2014-03-11T07:24:45+00:00; -3s from local t
```

Spoof NSE Script Results?

- Uses a LUA interpreter that could be instrumented
- Uses some standard set of library calls for external access
- Could try and find "external" inputs that exercise maximum code paths through instrumented interpreter...

Conclusion: hard but potentially possible

Spoof Service Detection?

- Uses regular expressions and well defined probes:
`https://svn.nmap.org/nmap/nmap-service-probes`
- Patterns defined in simple text format:
`http://nmap.org/book/vscan-fileformat.html`
- Regular expressions can generate as well as consume text

Conclusion: low effort, high yield!

Used Erlang to rapid prototype:

- Partially for trivial cross-platform support
- Partially for memory safety + error recovery on
- Weekend hack that can invert over 90% of the regexes in current nmap-service-probes

```
[amtal@foo src]$ wc -l *.erl
  16 nnop_app.erl
 123 nnop_parse.erl
  34 nnop_port.erl
 157 nnop_regen.erl
  28 nnop_sup.erl
  75 nnop_tree.erl
 433 total
```

Simple:

- Parse nmap-service-probes file
- Generate valid matches for regexes found
- Pick a probe for a TCP port, and respond to it

```
nnop_port:run(2000,2099, [], ["ftp","telnet","http"]),  
nnop_port:run(2100,2149, ["backdoor"], []),  
nnop_port:run(2150,2200, ["telnet","ftp","http"], []).
```

How do we actually "lie in a way which is not easily filtered"?

- Nmap groups services by http, ftp, telnet, etc...
- Multiple telnet or http servers are a red flag, avoid
- Results can pass visual inspection if ports are non-consecutive
- Could start using Nmap's service frequency data to allocate fake ports

Resource Constrained Hosts

- Rewrite in C, while retaining memory safety and fault recovery? (No)
- How about tunneling to higher-resource device?

Setup Wireless Services Security Access Restrictions **NAT / QoS** Administration

Port Forwarding Port Range Forwarding Port Triggering UPnP DMZ QoS

Port Forward

Forwards

Application	Port from	Protocol	IP Address	Port to	Enable
SubSeven	27374	TCP ▼	192.168.1.119	27374	<input checked="" type="checkbox"/>

Add Remove

Save Apply Settings Cancel Changes

Not Being Easy to Filter

- Regex output is currently deterministic (see version numbers)
- Could match services to ports they're expected to show up on

Messing with Humans

- Could randomize success
- Humans hate random failure in tools
- Alternatively, proxy a legitimate service but catch Nmap probes

Plot Twist: Someone Already Did It!

`http://www.saltwaterc.eu/portspoof-trolling.html`
(Found out morning before presentation.)

- Targets a single platform
- Lightweight C implementation
- Aims to remove Nmap as a tool altogether
- Significantly farther along

Different goals, far more functional and featureful

- Can be very easy (see <https://github.com/amtal/nnop>)
- Look for well structured collections of pattern matches

Tool Limitations

- Things break when you run into an adversarial situation
- Attempting to fix resulting problem would result in arms race