

# Advanced network scanning with Nmap 6

Henri DOREAU  
henri.doreau@gmail.com



13<sup>th</sup> LSM - Geneva 2012

# Outline

- 1 Project presentation
  - Introduction
- 2 Nmap Scripting Engine
  - Presentation
  - Internals
  - Usage
- 3 Nmap 6 new features
  - IPv6 support
  - Performance improvements
  - Companion tools
  - NSE
- 4 Ongoing developments
  - Upcoming features
  - Project

# Outline

- 1 Project presentation
  - Introduction
- 2 Nmap Scripting Engine
  - Presentation
  - Internals
  - Usage
- 3 Nmap 6 new features
  - IPv6 support
  - Performance improvements
  - Companion tools
  - NSE
- 4 Ongoing developments
  - Upcoming features
  - Project

# Nmap Security Scanner

## Full-featured Network scanner

- Port scanner
- Version and OS fingerprinting
- Lua scripting engine
- Companion tools (zenmap, ncat, nping, ndiff...)

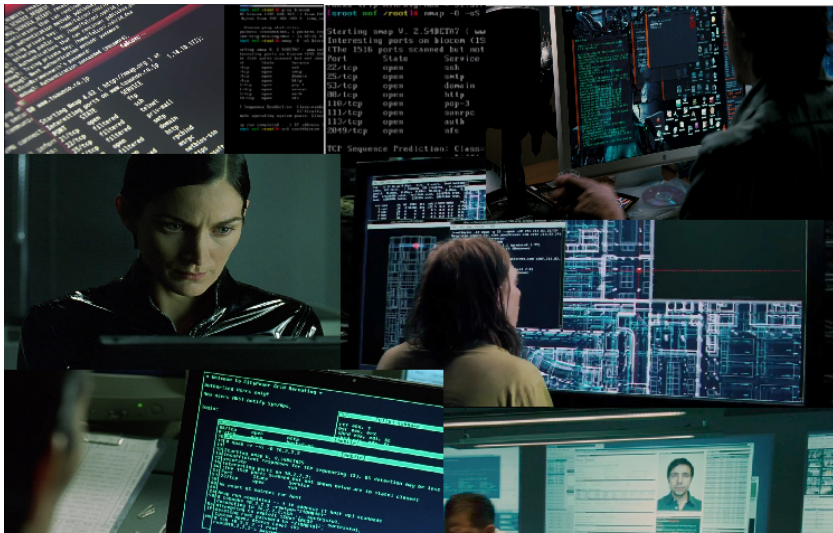
# Nmap Security Scanner

## **Vibrant community**

- Fingerprint DBs
- CPEs
- Scripts and NSE libraries

# Nmap Security Scanner

## Hollywood movie star



# Outline

- 1 Project presentation
  - Introduction
- 2 Nmap Scripting Engine
  - Presentation
  - Internals
  - Usage
- 3 Nmap 6 new features
  - IPv6 support
  - Performance improvements
  - Companion tools
  - NSE
- 4 Ongoing developments
  - Upcoming features
  - Project

# Introduction

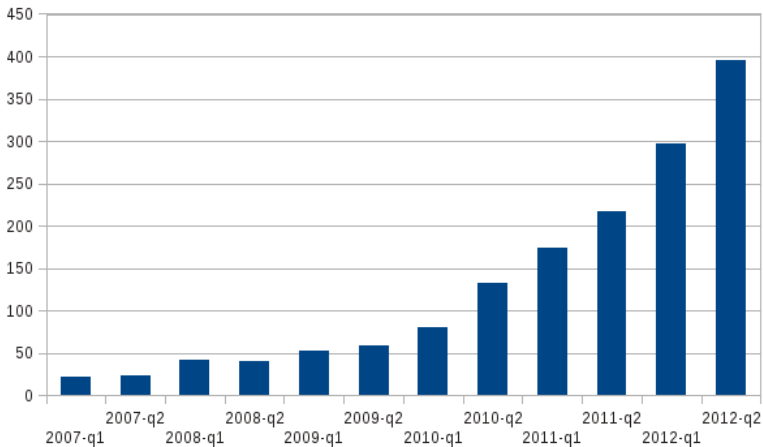
## Built-in lua scripting engine

- Network exploration
- Sophisticated version detection
- Vulnerability detection
- Scan results post-processing



# NSE development

## Script collection growth



# Script phases

## Four execution modes

- Prerules
- Service
- Host
- Postrules

- **NSE Pre-scan**
  - 1 Host enumeration
  - 2 Host discovery
  - 3 Reverse DNS resolution
  - 4 Port scan
  - 5 Version detection / RPC grind
  - 6 OS fingerprinting
  - 7 Traceroute
  - 8 **Script scan**
  - 9 Output
- **NSE Post-scan**

# Script structure

## When to run?

```
hostrule = function(host)
  return host.directly_connected
end
```

```
portule = shortport.http
```

⇒ script can have several rule and action functions

# Sample output

```
Nmap scan report for scanme.nmap.org (74.207.244.221)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7
80/tcp    open  http     Apache httpd 2.2.14 ((Ubuntu))
|_http--title: Go ahead and ScanMe!
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Host script results:
| firewall:
| HOP  HOST                PROTOCOL  BLOCKED PORTS
| 0    192.168.0.15        tcp       139
|_10   64.62.250.6         tcp       135,445
```

# Design

## NSE parallelism

- Single nmap thread
- lua coroutines

- ⇒ Lightweight and efficient non-blocking mechanism
- ⇒ Script writers get parallelism for free
- ⇒ No concurrent memory access concerns *ever*

# Adaptive workflow

## Two ways to invoke scripts

### Point and shoot

```
nmap --script samba-vuln-cve-2012-1182 <target>  
nmap --script +mongodb-info -p80 <target>
```

⇒ No silent dependencies

### Aim oriented

```
nmap --script "http-* and not brute" <target>
```

# Script categories

## Grouped by categories

- default
- intrusive
- external
- . . .

| NSEDoc               |  |
|----------------------|--|
| index                |  |
| NSE Documentation    |  |
| <b>Categories</b>    |  |
| auth                 |  |
| broadcast            |  |
| brute                |  |
| default              |  |
| discovery            |  |
| dos                  |  |
| exploit              |  |
| external             |  |
| fuzzer               |  |
| intrusive            |  |
| malware              |  |
| safe                 |  |
| version              |  |
| vuln                 |  |
| <b>Scripts (396)</b> |  |
| acarsd-info          |  |
| address-info         |  |
| afp-brute            |  |
| afp-is               |  |
| afp-path-vuln        |  |
| afp-serverinfo       |  |
| afp-showmount        |  |
| ajp-auth             |  |
| ajp-brute            |  |
| ajp-headers          |  |
| ajp-methods          |  |
| ajp-request          |  |
| amqp-info            |  |

| Scripts        |   |
|----------------|---|
| acarsd-info    | Retrieves information from a listening acarsd daemon. Acarsd decodes ACARS (Aircraft Communication Addressing and Reporting System) data in real time. The information retrieved by this script includes the daemon version, API version, administrator e-mail address and listening frequency. |
| address-info   | Shows extra information about IPv6 addresses, such as embedded MAC or IPv4 addresses when available.  |
| afp-brute      | Performs password guessing against Apple Filing Protocol (AFP).   |
| afp-is         | Attempts to get useful information about files from AFP volumes. The output is intended to resemble the output of ls.   |
| afp-path-vuln  | Detects the Mac OS X AFP directory traversal vulnerability, CVE-2010-0533.  |
| afp-serverinfo | Shows AFP server information. This information includes the server's hostname, IPv4 and IPv6 addresses, and hardware type (for example Macm.ln.ln or MacBookPro).   |
| afp-showmount  | Shows AFP shares and ACLs.  |
| ajp-auth       | Retrieves the authentication scheme and realm of an AJP service (Apache JServ Protocol) that requires authentication.   |
| ajp-brute      | Performs brute force passwords auditing against the Apache JServ protocol. The Apache JServ Protocol is commonly used by web servers to communicate with back-end Java application server containers.   |
| ajp-headers    | Performs a HEAD or GET request against either the root directory or any optional directory of an Apache JServ Protocol server and returns the server response headers.  |
| ajp-methods    | Discovers which options are supported by the AJP (Apache JServ Protocol) server by sending an OPTIONS request and lists potentially risky methods.  |
| ajp-request    | Requests a URI over the Apache JServ Protocol and displays the result (or stores it in a file). Different AJP methods such as, GET, HEAD, TRACE, PUT or DELETE may be used.   |
| amqp-info      | Gathers information (a list of all server properties) from an AMQP (advanced message queuing protocol) server.  |

see <http://nmap.org/nosedoc>

# Outline

- 1 Project presentation
  - Introduction
- 2 Nmap Scripting Engine
  - Presentation
  - Internals
  - Usage
- 3 Nmap 6 new features
  - IPv6 support
  - Performance improvements
  - Companion tools
  - NSE
- 4 Ongoing developments
  - Upcoming features
  - Project



# Full IPv6 support

## Long standing wish

- All features (provided it makes any sense)
- All supported platforms

# Full IPv6 support

## Long standing wish

- All features (provided it makes any sense)
- All supported platforms

YEAH!!!

# Brand new OS fingerprinting engine

## Innovative approach: machine learning techniques

- Reduced dataset
- Increased adaptiveness
- Very accurate

⇒ See <http://nmap.org/book/osdetect>

# IPv6 support

**Honestly, who cares?**

# IPv6 support

**Honestly, who cares?**



The future is already there!

# Enhanced performances

## Three main axis of improvement

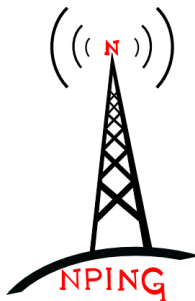
- Memory footprint
- High performance and scalable I/O notification facilities
- Application-specific optimizations (NSE)

cf. *Scanning the Internet*, by Fyodor

# Nping

## Reimplementation of the venerable hping2

- Modern, high performance tool
- Leverages nmap libraries
- Provides new packet crafting classes to nmap



# Nping Echo mode

## Replacement for ping+tcpdump

- ① nping in server mode on target
- ② client probes the target
- ③ server returns captured probes to the client(s) as encrypted payloads



# Zenmap tologoy tab

Finally: actual network maps from the network mapper!

The screenshot displays the Zenmap application interface. At the top, the menu bar includes 'Scan', 'Tools', 'Profile', and 'Help'. The 'Target' field is set to 'scanme.nmap.org reddit.com facet', and the 'Profile' is empty. The 'Command' field contains the following text: `nmap -T4 -A -oA popsites scanme.nmap.org reddit.com facebook.com google.com 4chan.org news.ycombinr`. Below the command field are tabs for 'Hosts', 'Services', 'Nmap Output', 'Ports / Hosts', 'Topology', 'Host Details', and 'Scans'. The 'Topology' tab is currently selected, showing a network map with nodes and connecting lines. The nodes are labeled with IP addresses, such as 216.187.89.101, 74.207.244.221, 134.25, 206.228.118.89, 184.105.213.198, 127.0.0.1/8, 64.214.174.245, 74.207.254.18, 111.4, 72.52.92.110, 184.105.149.85, 198.92.170.74, 119.76, 204.15.20, 104.15.20, 69.171.229.11, 123.116.176, 64.62.250.5, 184.105.213.174, 184.105.213.66, 198.32.275.68, 137.164.47.175, 137.164.130.57, 184.105.213.78, 137.164.47.19, 137.164.47.122, 20.223, 209.65.148.102, 137.164.47.112, 68.65.168.33, 171.64.255.129, 173.223.232.51, 207.285.17.129, 137.164.50.31, 171.64.255.164, 72.4.122.10, 128.32.0.37, 128.32.255.41, 72.4.122.121, 207.97.227.294, and 169.229.131.8. The interface also includes a 'Hosts' list on the left with entries like 'mit.edu (18.9', 'facebook.com', 'lwn.net (72.5', 'scanme.nmap.', 'nmap.org (74', '4chan.org (10', '2600.com (16', 'berkeley.edu', 'stanford.edu', 'defcon.org (1', 'forum.defcon.', 'reddit.com (1', 'news.ycombinr', 'github.com (2', 'google.com (2', and 'redhat.com (2'. A 'Filter Hosts' field is at the bottom left, and a 'Save Graphic' button is at the top right of the map area.

# Better web scanning

## Big focus on web technologies

- Pipelining
- Built-in web crawler
- Caching
- Web-specific security checks

# NSE frameworks

## Implemented as NSE libraries

### brute

Parallel network authentication cracking module.

### credentials

Leverage and report discovered credentials.

### vulns

Consistent vulnerability reports and efficient post-processing.

# Outline

- 1 Project presentation
  - Introduction
- 2 Nmap Scripting Engine
  - Presentation
  - Internals
  - Usage
- 3 Nmap 6 new features
  - IPv6 support
  - Performance improvements
  - Companion tools
  - NSE
- 4 Ongoing developments
  - Upcoming features
  - Project

# Upcoming: web scanning

## Continued effort on HTTP

- Implement latest performance-related protocols and paradigms
- WebSocket mode to ncat

## Upcoming: extend NSE

### Expand the role and features of NSE

- Leveraging native libraries from lua
- NSE-based port scanning
- Re-implementing older code within NSE
- Adapting NSE to the companion tools

## Upcoming: misc

### but also...

- Combining IP v4/v6 scans
- Improving scalability
- Scanning through proxies
- Remote checks through authenticated SSH connections
- Updater

# Get involved!

**Your own awesome idea!**

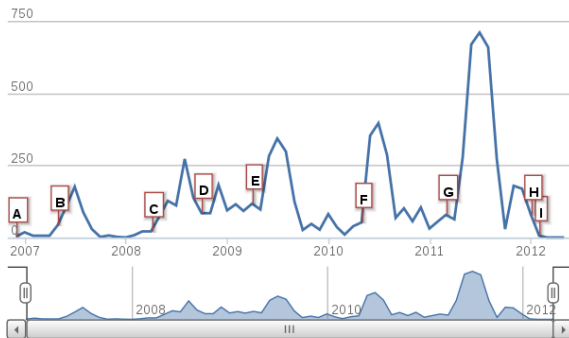
...and code? ;)



# Development

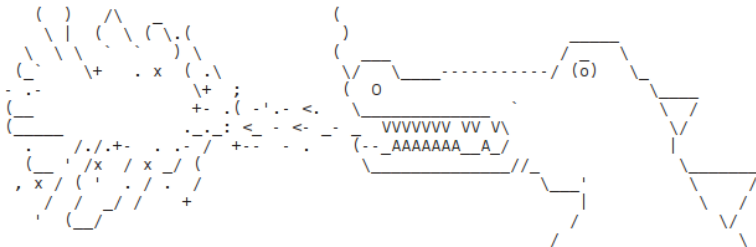
## Increasing development pace

- 2011 was the most active year ever in the project history! (ohloh.net).
- 8<sup>th</sup> consecutive Google Summer of Code



# Happy birthday nmap!

15<sup>th</sup> birthday this year (Sept. 1<sup>st</sup>)



# Questions?

`http://nmap.org`

`nmap-dev@insecure.org` (it's cool, join!)