



# Network Fingerprinting

Erik Hjelmvik  
Jonas Lejon

Netnod Tech Meeting  
Oct 2024

What is this?

t13i190800\_9dc949149365\_97f8aa674fd9

# It's a JA4 fingerprint of Sliver!

```
IP (no domain)
|
| hash of ciphers
| #extensions /
| /
| /
| t13i190800_9dc949149365_97f8aa674fd9
| / \ \ /
| TLS 1.3 \ no ALPN /
| \
| #ciphers hash of extensions and
| signature algorithms
```

## About us

- Erik Hjelmvik  
CTO at Netresec
- Jonas Lejon  
Cybersecurity specialist at Triop

# Fi vill veta:

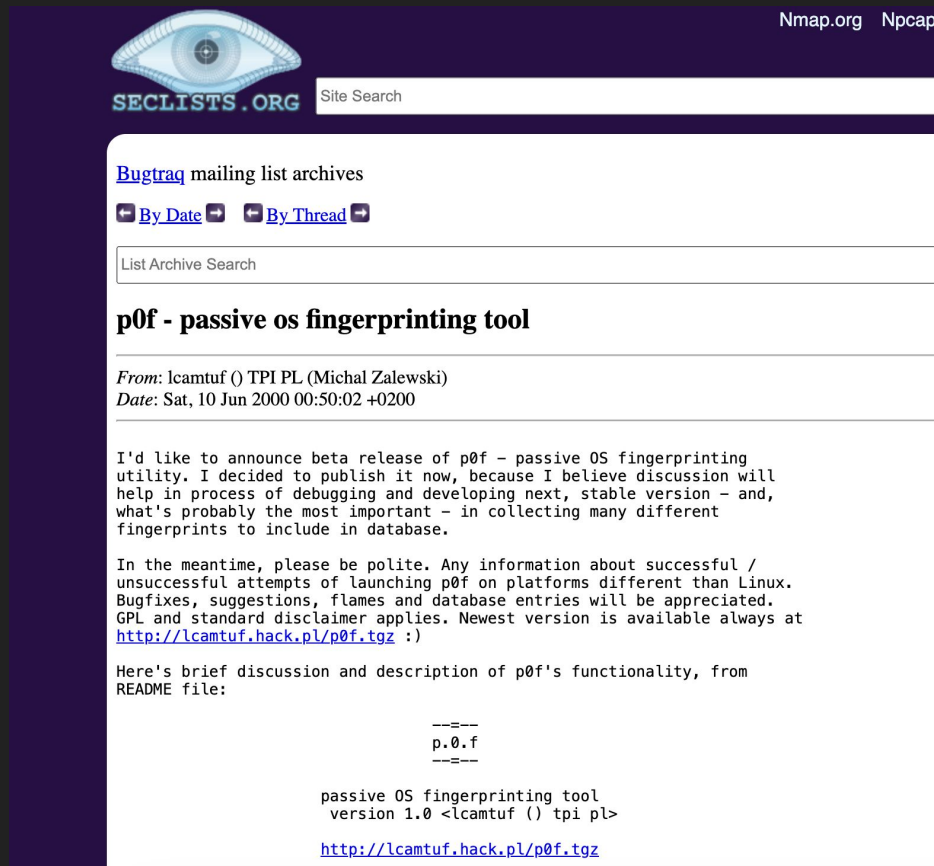
vem  
var  
vad  
när  
hur



Signalskydd –  
Sekretesskydd

# History

- p0f - Passive
  - Announced by Michal Zalewski in 2000
- JA3/S
  - Announced in 2017
- Nmap - Active
  - OS detection since 1998
  - Protocol detection



Nmap.org Npcap

SECLISTS.ORG Site Search

[Bugtraq](#) mailing list archives

[By Date](#) [By Thread](#)

List Archive Search

## p0f - passive os fingerprinting tool

*From:* lcamtuf () TPI PL (Michal Zalewski)  
*Date:* Sat, 10 Jun 2000 00:50:02 +0200

I'd like to announce beta release of p0f - passive OS fingerprinting utility. I decided to publish it now, because I believe discussion will help in process of debugging and developing next, stable version - and, what's probably the most important - in collecting many different fingerprints to include in database.

In the meantime, please be polite. Any information about successful / unsuccessful attempts of launching p0f on platforms different than Linux. Bugfixes, suggestions, flames and database entries will be appreciated. GPL and standard disclaimer applies. Newest version is available always at <http://lcamtuf.hack.pl/p0f.tgz> :)

Here's brief discussion and description of p0f's functionality, from README file:

```
-----  
p.0.f  
-----  
  
passive OS fingerprinting tool  
version 1.0 <lcamtuf () tpi pl>  
  
http://lcamtuf.hack.pl/p0f.tgz
```

# What is Network Fingerprinting?

- Identifying applications
- Identifying operating systems
- Identifying communication libraries
- Devices

## How?

- Passive (sniffing packets)
- Active (sending packets)

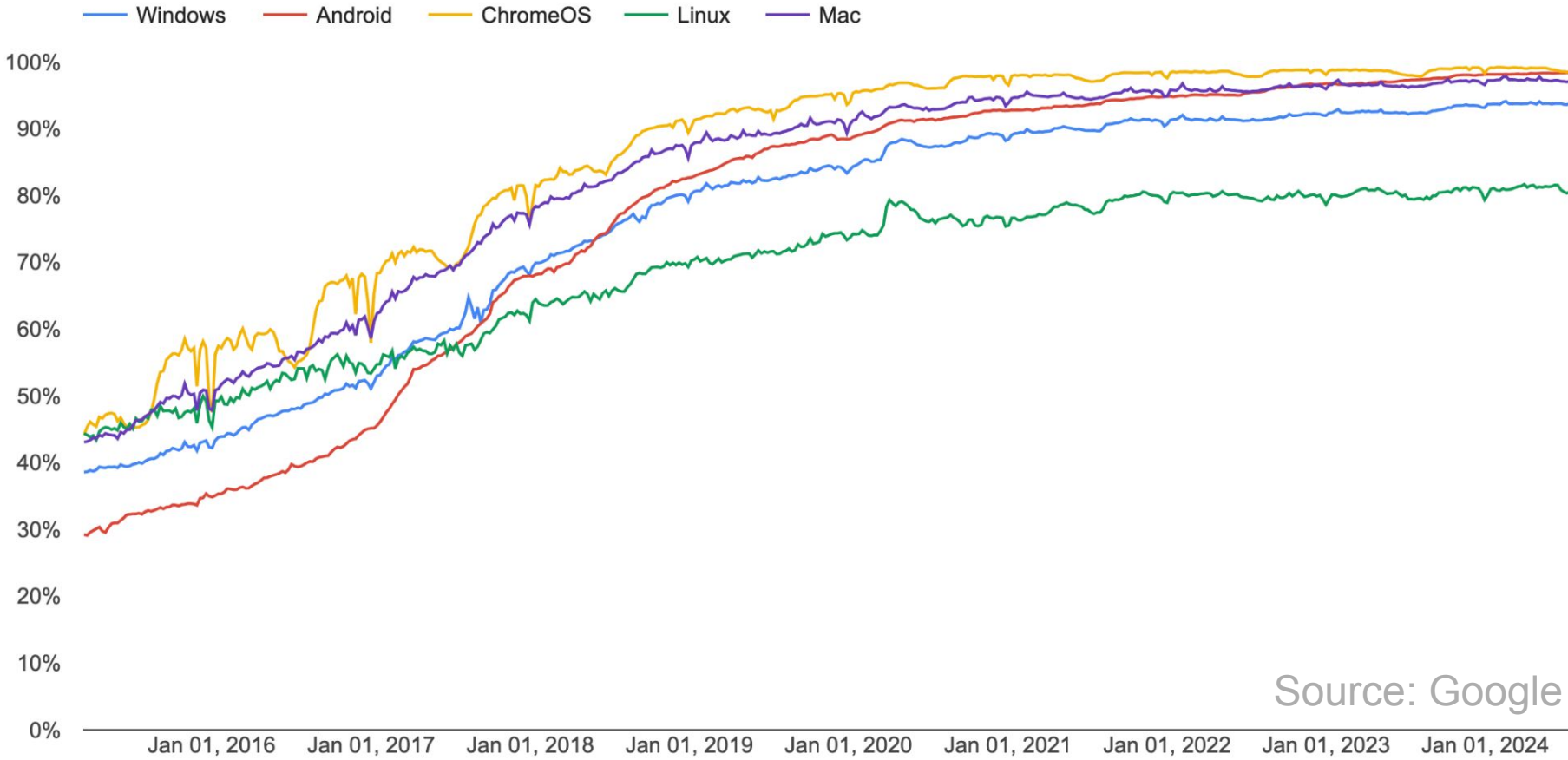


# Why?

- Almost all traffic on the internet is encrypted
- Malware identification
  - Backdoors, Command and Control traffic and servers
- Anti censorship identification
  - Tor
- Web Application Firewalls and Intrusion Detection Systems
- DoS/DDoS prevention
- QoS



Percentage of pages loaded over HTTPS in Chrome by platform



Source: Google



# What is this?

Mozilla/5.0 (Windows NT 6.0; rv:39.0) Gecko/20100101 Firefox/39.0

# It's a browser user-agent string!

Windows Vista  
\  
\  
Mozilla/5.0 (Windows NT 6.0; rv:39.0) Gecko/20100101 Firefox/39.0  
/  
/  
Firefox v.39

Operating System

Windows NT 4.0

Platform Token

Windows NT 4.0



## Operating System

Windows NT 4.0

Windows 2000

## Platform Token

Windows NT 4.0

Windows NT 5.0



## Operating System

## Platform Token

Windows NT 4.0

Windows NT 4.0



Windows 2000

Windows NT 5.0

Windows XP

Windows NT 5.1

Windows Vista

Windows NT 6.0

Windows 7

## Operating System

## Platform Token

Windows NT 4.0

Windows NT 4.0



Windows 2000

Windows NT 5.0

Windows XP

Windows NT 5.1

Windows Vista

Windows NT 6.0

Windows 7

Windows NT 6.1

WTF?

## Operating System

## Platform Token

Windows NT 4.0

Windows NT 4.0



Windows 2000

Windows NT 5.0

Windows XP

Windows NT 5.1

Windows Vista

Windows NT 6.0

Windows 7

Windows NT 6.1

WTF?

Windows 8

Windows NT 6.2

## Operating System

## Platform Token

Windows NT 4.0

Windows NT 4.0



Windows 2000

Windows NT 5.0

Windows XP

Windows NT 5.1

Windows Vista

Windows NT 6.0

Windows 7

Windows NT 6.1

WTF?

Windows 8

Windows NT 6.2

Windows 10

Windows NT 10.0





## Operating System

## Platform Token

Windows NT 4.0

Windows NT 4.0



Windows 2000

Windows NT 5.0

Windows XP

Windows NT 5.1

Windows Vista

Windows NT 6.0

Windows 7

Windows NT 6.1

WTF?

Windows 8

Windows NT 6.2

Windows 10

Windows NT 10.0



Windows 11

Windows NT 10.0

WTF?

## Operating System

## Platform Token

Windows NT 4.0

Windows NT 4.0



Windows 2000

Windows NT 5.0

Windows XP

Windows NT 5.1

Windows Vista

Windows NT 6.0

Windows 7

Windows NT 6.1

WTF?

Windows 8

Windows NT 6.2

Windows 10

Windows NT 10.0



Windows 11

Windows NT 10.0

WTF?

Windows 12

(???)

# JA3 vs. JA4

	JA3	JA4
Input	TLS Version, Cipher Suites, Extensions, Elliptic Curves, EC Point Formats	Everything in JA3 plus: SNI flag, ALPN value, signature algorithms
Hash	MD5	Raw_SHA256_SHA256
Example	a85be79f7b569f1df5e6087b69deb493	t13i010400_of2cb44170f4_5c4c70b73fa0



# What to Fingerprint?

Input	Tools / Fingerprints	Output
L7 protocol	User-Agent, JA4HTTP, httpprint, NIDS signatures	Application or Malware
TLS	JA3, JA4	TLS library Sometimes application / OS
IP / TCP / DHCP	p0f, Satori, JA4TCP	Operating System
Ethernet / WiFi	OUI	NIC Manufacturer / Hardware

# JA4+

Full Name	Short Name	Description
JA4	JA4	TLS Client Fingerprinting
JA4Server	JA4S	TLS Server Response / Session Fingerprinting
JA4HTTP	JA4H	HTTP Client Fingerprinting
JA4Latency	JA4L	Client to Server Latency Measurement / Light Distance
JA4LatencyServer	JA4LS	Server to Client Latency Measurement / Light Distance
JA4X509	JA4X	X509 TLS Certificate Fingerprinting
JA4SSH	JA4SSH	SSH Traffic Fingerprinting
JA4TCP	JA4T	TCP Client Fingerprinting
JA4TCPServer	JA4TS	TCP Server Response Fingerprinting
<a href="#">JA4TCPScan</a>	<a href="#">JA4TScan</a>	<a href="#">Active TCP Fingerprint Scanner</a>

# JA4+

Full Name	Short Name	Description
JA4	JA4	TLS Client Fingerprinting
JA4Server	JA4S	TLS Server Response / Session Fingerprinting
JA4HTTP	JA4H	HTTP Client Fingerprinting
JA4Latency	JA4L	Client to Server Latency Measurement / Light Distance
JA4LatencyServer	JA4LS	Server to Client Latency Measurement / Light Distance
JA4X509	JA4X	X509 TLS Certificate Fingerprinting
JA4SSH	JA4SSH	SSH Traffic Fingerprinting
JA4TCP	JA4T	TCP Client Fingerprinting
JA4TCPServer	JA4TS	TCP Server Response Fingerprinting
<a href="#">JA4TCPScan</a>	<a href="#">JA4TScan</a>	<a href="#">Active TCP Fingerprint Scanner</a>









# OS fingerprinting with TCP/IP

The screenshot shows the NetworkMiner 2.9.0 application window. The main display area shows the following information for the host 192.168.88.254:

- IP: 192.168.88.254
- MAC: 00269ECF5849
- NIC Vendor: Quanta Computer Inc.
- MAC Age: 2009-06-24
- Hostname: Donald-PC.local, Donald-PC
- OS: Windows
- pOf (NetSA): Windows Vista SP0/SP2, 7 SP0+, 2008 SP0 [Windows] (100,0%)
- Satori DHCP: Windows - Windows 10 (23,08 %) Windows - Windows 7 (15,3 %)
- Satori TCP: Windows - Windows 8 (28,57 %) Windows - Windows 7 (28,57 %)

The Case Panel on the right shows a table with the following data:

Filename	MD5
MD_201...	7dca84...

At the bottom of the window, there is a "Buffered Frames to Parse:" field.

pOf:

<https://lcamtuf.coredump.cx/pOf3/>

Carnegie Mellon CERT's pOf:

<https://tools.netsa.cert.org/pOf/index.html>

Satori TCP/DHCP:

<https://github.com/xnih/satori>

# Active Fingerprinting

- JARM - TLS Fingerprinting
  - Sends 10 TLS Client Hello packets
  - 62 character fingerprint
- Looks something like this:

00000000000000000000000043d43d00043de2a97eabb398317329f027c66e4c1b01

(Sliver gRPC admin listening on default TCP port 31337)

```
↳ ~ git clone https://github.com/salesforce/jarm.git
Cloning into 'jarm'...
remote: Enumerating objects: 102, done.
remote: Counting objects: 100% (62/62), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 102 (delta 53), reused 39 (delta 39), pack-reused 40 (from 1)
Receiving objects: 100% (102/102), 38.17 KiB | 640.00 KiB/s, done.
Resolving deltas: 100% (54/54), done.
```

```
↳ ~ cd jarm
```

```
↳ jarm git:(master) python3 jarm.py -p 31337 [REDACTED]
```

```
Domain: [REDACTED]
```

```
Resolved IP: [REDACTED]
```

```
JARM: 000000000000000000000043d43d00043de2a97eabb398317329f027c66e4c1b01
```

```
↳ jarm git:(master) python3 jarm.py -p 8834 [REDACTED]
```

```
Domain: [REDACTED]
```

```
Resolved IP: [REDACTED]
```

```
JARM: 2ad2ad0002ad2ad00042d42d000000020120996177a65431cde640fa58d2e8
```

```
↳ jarm git:(master) █
```



# Active Fingerprinting

Scan 10

JARM: 0000000000000000000043d43d00043de2a97eabb398317329f027c66e4c1b01

Scan 1: |||,

Scan 2: |||,

Scan 3: |||,

Scan 4: |||,

Scan 5: |||,

Scan 6: |||,

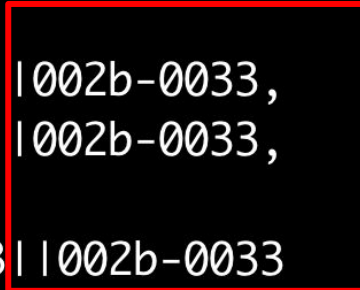
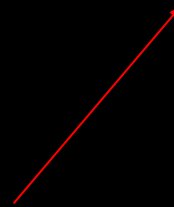
Scan 7: 1303|0303||002b-0033,

Scan 8: 1303|0303||002b-0033,

Scan 9: |||,

Scan 10: 1303|0303||002b-0033

hash of extensions



# Databases

- On Github
- ja4db.com
- <https://ja3.zone/>
- Censys, BinadyEdge Shodan
  - JARM
- VirusTotal - JA3/JA3S
- [ssllbl.abuse.ch/ja3-fingerprints/](https://ssllbl.abuse.ch/ja3-fingerprints/)

ja4db.com/fingerprint/ja4/t13d190900\_9dc949149365\_97f8aa674fd9

**Applications**  
Sliver Agent

**Libraries**  
GoLang

**OSs**

**Observations**  
1

## t13d190900\_9dc949149365\_97f8aa674fd9

a JA4_a t13d190900	b JA4_b 9dc949149365	c JA4_c 97f8aa674fd9	
🔒 Protocol TLS	📄 TLS Version 1.3	🚩 SNI Domain	# Cipher Count 19
# Extension Count 9	🔗 ALPN Value None	⚙️ Cipher Hash 9dc949149365	⚙️ Extension Hash 97f8aa674fd9

# Final Words

- Most traffic is encrypted
- Passive fingerprinting often gives false positives
  - Combine with active fingerprinting
  - And other intel such as X509 attributes, IP, DNS etc..
- The Internet is full of “middleboxes”
- TLS 1.3 = more fingerprints due to more extensions



# Anti fingerprinting

- Chrome sends extensions in random order
  - Works with JA4 but not JA3
- Cobalt Strike (C&C) TLS stack matches Java (server side)
  - Client uses Windows TLS stack
- Sliver C&C https implant listener
  - `-E, --disable-randomized-jarm`      disable randomized jarm fingerprints
- Traffic analysis
  - Maybenot, a framework for traffic analysis defenses (Pulls and Witwer)
    - Tor Circuit Padding Framework
  - OBFS4, Snowflake, meek-azure