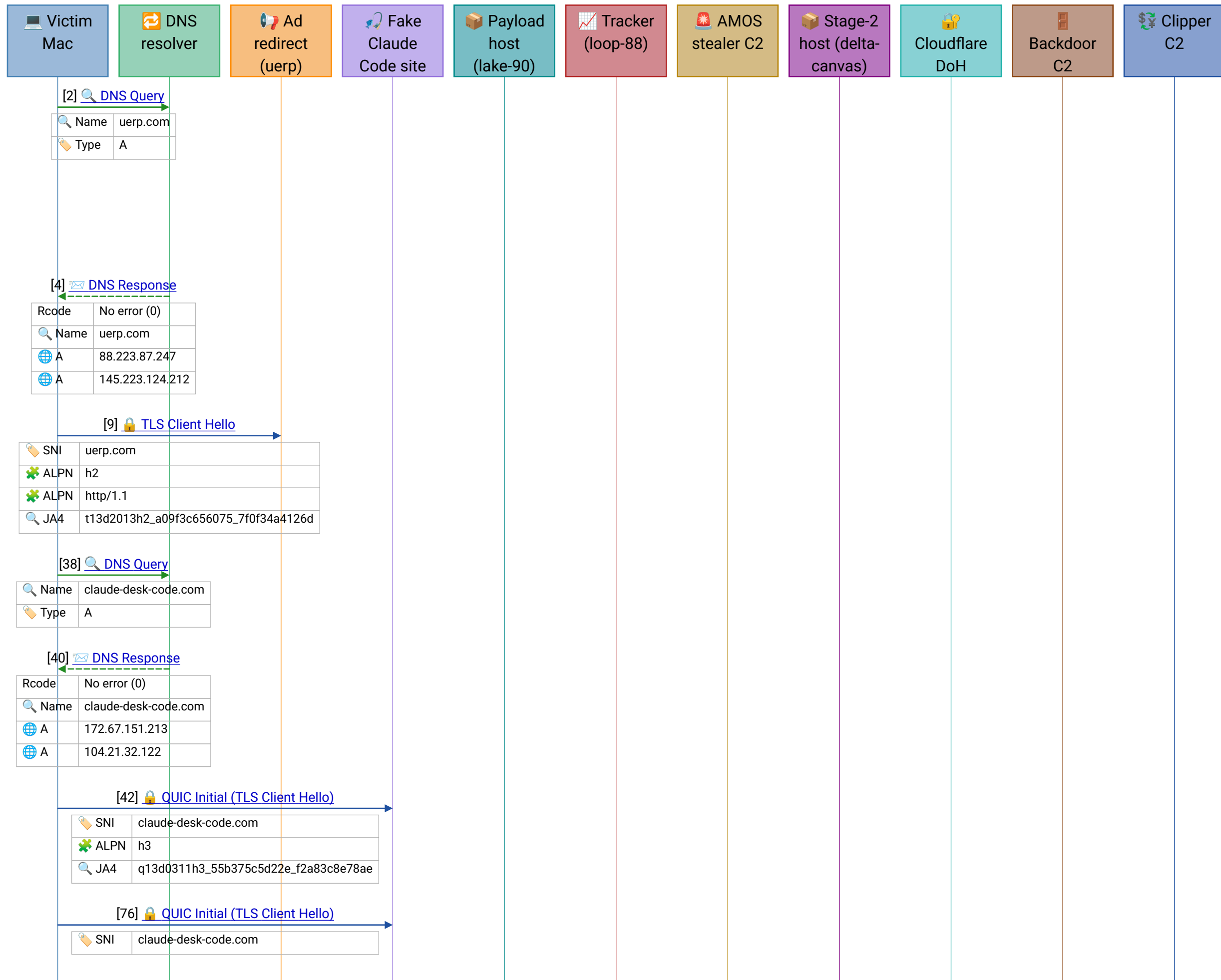




[Frame 2]

Phase 1: A search ad leads to a fake Claude Code site

The Mac's first lookup in the capture is for uerp [.]com, the domain behind the Google ad Brad clicked at 22:25 UTC. The ad shows a trusted-looking listing, but its click-through URL passes through this redirect first.

Safari also asks for an HTTPS -type record alongside each name; the diagram keeps only the A (IPv4 address) lookups.

[Frame 4] The local resolver answers with two IPv4 addresses for uerp [.].com. Nothing here is unusual yet: a single DNS answer cannot tell a redirector from an ordinary site.

[Frame 9] Safari opens HTTPS to the redirector. The contents are encrypted, but the TLS **Client Hello** is not: it names the site in the SNI field and carries the browser's fingerprint.

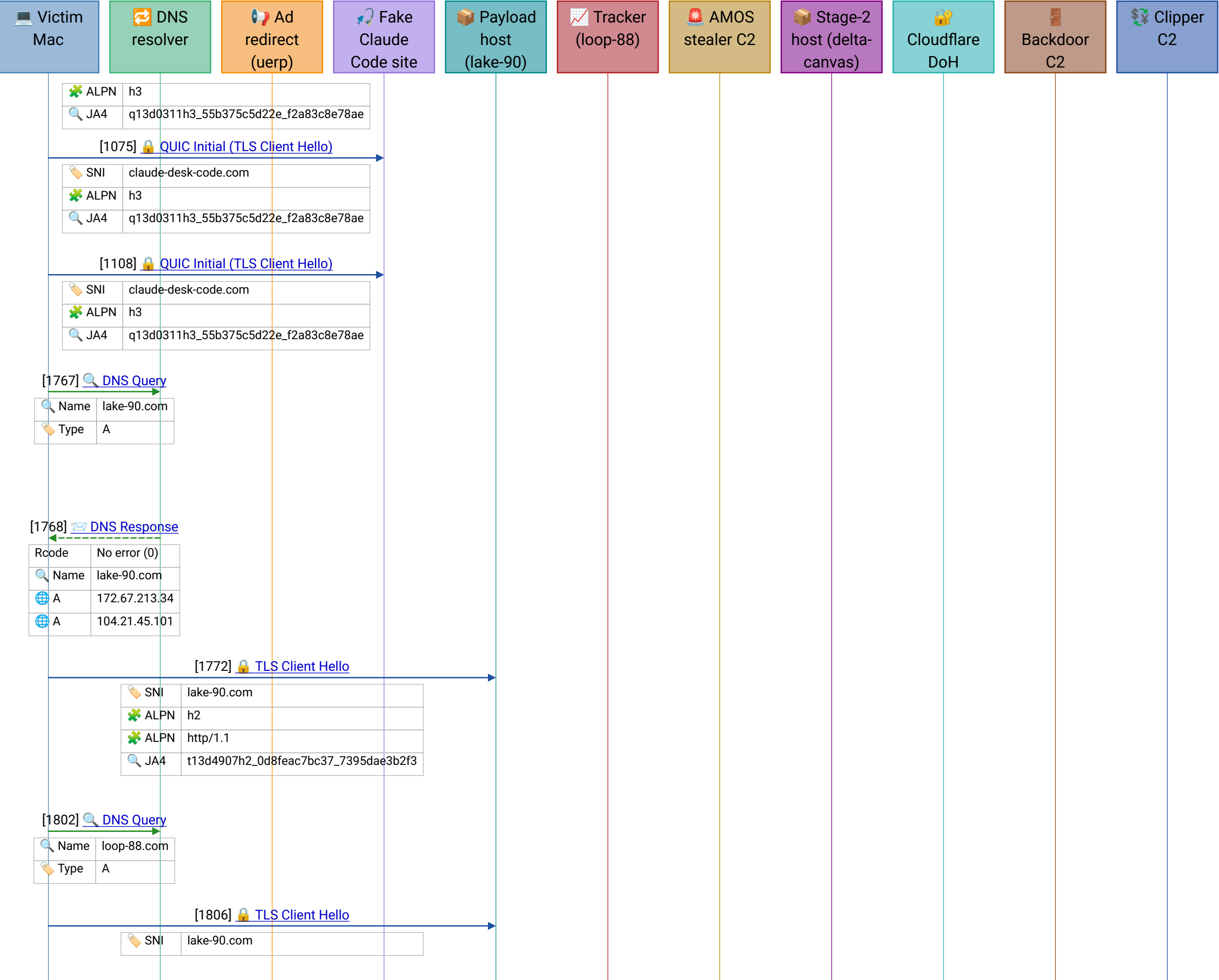
Remember this JA4 value, t13d2013h2_ It is Safari's. Later, a different fingerprint will show that something other than the browser is talking.

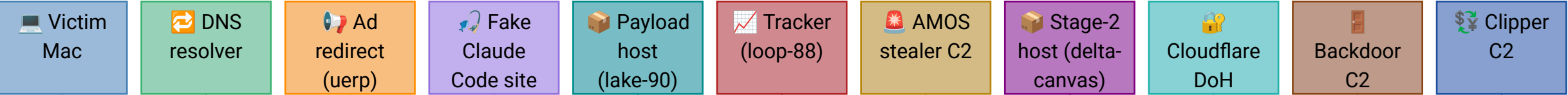
[Frame 38] The redirect sends Safari to claude-desk-code [.]com, a look-alike domain hosting a page that impersonates the Claude Code install page. This is the lookup that a DNS-based blocklist would have to catch.

[Frame 40] Two addresses come back, 172.67.151 [.]213 and 104.21.32 [.]122. Both belong to Cloudflare, which fronts the fake site, so blocking by IP address would also block many legitimate sites behind the same provider. The domain name is the useful indicator.

[Frame 42] Safari reaches the fake site over QUIC (HTTP/3, UDP port 443). The first QUIC packet, the Initial, is encrypted only with keys anyone can derive, so Wireshark can read the Client Hello inside it: SNI `claudio-desk-code[.]com`, ALPN `h3`, and Safari's QUIC fingerprint `q13d0311h3_...`

[Frame 76] A second QUIC connection to the same site, a third of a second later: Safari loading the rest of the page.





ALPN	h2
ALPN	http/1.1
JA4	t13d4907h2_0d8feac7bc37_7395dae3b2f3

[1807] DNS Response

Rcode	No error (0)
Name	loop-88.com
A	104.21.3.173
A	172.67.130.250

[1822] TLS Client Hello

SNI	loop-88.com
ALPN	h2
ALPN	http/1.1
JA4	t13d4907h2_0d8feac7bc37_7395dae3b2f3

[2885] TLS Client Hello

SNI	loop-88.com
ALPN	h2
ALPN	http/1.1
JA4	t13d4907h2_0d8feac7bc37_7395dae3b2f3

[2962] POST /api/metrics/run

Method	POST
Host	152.42.143.45
User-Agent	curl/8.7.1
Content-Type	application/json
Content-Length	201
event	started
stage	boot

[2964] HTTP 204

Phrase	No Content
Server	nginx
Request in frame	2962
Time since request	181.213000 milliseconds
For	/api/metrics/run

/tmp/.zxqa03cu : a universal Mach-O file for both Intel and Apple-silicon Macs.

[Frame 1807] loop-88 [.]com resolves to Cloudflare addresses as well.

[Frame 1822] curl reports to the tracker. Same curl fingerprint; the request itself (event=pasted) is inside TLS, so only Brad's notes show its path.

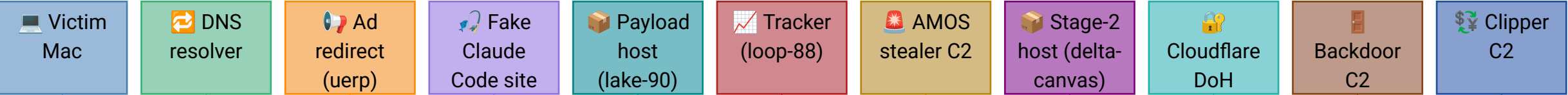
[Frame 2885] The second tracker report, event=stager , a second after the first. From here on the stealer talks to its own server over plain HTTP, and every byte is readable.

[Frame 2962] Phase 3: The stealer checks in and starts collecting

The stealer's first message to its command-and-control (C2) server at 152.42.143 [.]45: a JSON POST over plain HTTP on port 80, by bare IP address with no DNS lookup.

The body describes the victim (macOS, 27.0.1 , arm64, en_US) , a victim ID u , a build ID b , and event : started , stage : boot . Three red flags in one packet: curl as the User-Agent , a raw IP as the Host , and data posted to a server nobody looked up.

[Frame 2964] The server replies 204 No Content : received, nothing to say. It answers every progress report this way.



[2973] 🗑️ [POST /api/metrics/run](#)

Method	POST
🏠 Host	152.42.143.45
🕸 User-Agent	curl/8.7.1
📄 Content-Type	application/json
📏 Content-Length	207
🔑 event	stage
🔑 stage	init_session

[2977] 🗑️ [POST /api/metrics/run](#)

Method	POST
🏠 Host	152.42.143.45
🕸 User-Agent	curl/8.7.1
📄 Content-Type	application/json
📏 Content-Length	205
🔑 event	stage
🔑 stage	messengers

[2981] ✉️ [HTTP 204](#)

💬 Phrase	No Content	
💻 Server	nginx	
🔗 Request in frame	2973	
🕒 Time since request	153.354000 milliseconds	
🔙 For	/api/metrics/run	

[2986] 🗑️ [POST /api/metrics/run](#)

Method	POST
🏠 Host	152.42.143.45
🕸 User-Agent	curl/8.7.1
📄 Content-Type	application/json
📏 Content-Length	206
🔑 event	stage
🔑 stage	credentials

[2989] ✉️ [HTTP 204](#)

💬 Phrase	No Content	
💻 Server	nginx	

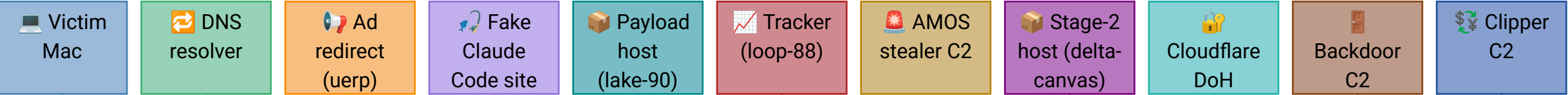
[Frame 2973] stage : init_session . The stealer reports each step as it starts, so the operators can see how far each infection gets. Each report opens a new TCP connection.

[Frame 2977] stage : messengers : it searches for chat apps such as Telegram, whose local data can be used to take over the account.

[Frame 2981] 204 for the init_session report (frame 2973). Replies arrive out of order because each report travels on its own connection.

[Frame 2986] stage : credentials : cloud and developer credentials. Brad's file list shows it searched for AWS, Google Cloud, Docker, and FileZilla configuration, plus the shell history.

[Frame 2989] 204 for the messenger s report (frame 2977).



 Request in frame	2977	
 Time since request	153.573000 milliseconds	
 For	/api/metrics/run	

[2995] 🗼 POST /api/metrics/run

Method	POST
 Host	152.42.143.45
 User-Agent	curl/8.7.1
 Content-Type	application/json
 Content-Length	203
 event	stage
 stage	browsers

[2998] 🗼 POST /api/metrics/run

Method	POST
 Host	152.42.143.45
 User-Agent	curl/8.7.1
 Content-Type	application/json
 Content-Length	202
 event	stage
 stage	wallets

[3002] ✉ HTTP 204

 Phrase	No Content	
 Server	nginx	
 Request in frame	2986	
 Time since request	148.592000 milliseconds	
 For	/api/metrics/run	

[3007] 🗼 POST /api/metrics/run

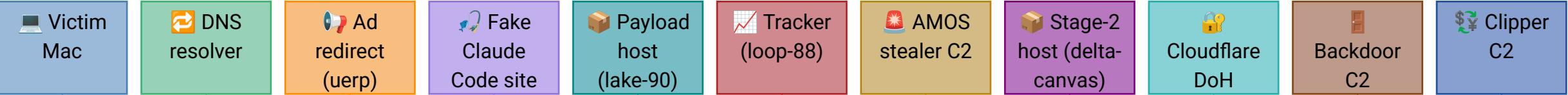
Method	POST
 Host	152.42.143.45
 User-Agent	curl/8.7.1
 Content-Type	application/json
 Content-Length	202
 event	stage
 stage	passmgr

[Frame 2995] stage : browsers : saved passwords, cookies, and autofill data from installed browsers.

[Frame 2998] stage : wallets : desktop cryptocurrency wallets. The file list shows folders for Binance and TonKeeper.

[Frame 3002] 204 for the credentials report (frame 2986).

[Frame 3007] stage : passmgr : password managers. The first upload, next, includes a passmgr /Bitwarden folder.



[3015] 🏠 [POST /contact](#)

Method	POST
🏠 Host	152.42.143.45
🔍 User-Agent	curl/8.7.1
📏 Content-Length	4069
📄 Content-Type	multipart/form-data; boundary=-----7JvFidNkO9sR2yj2cmeXgy
🔗 Form field	u
🔗 Form field	b
🔗 Form field	l
🔗 Form field	n
🔗 Form field	p
📎 File part	.A86Eo_A86ED3CE-61F0-40E0-A4F3-6F85D42B88DA.zip

[3017] ✉️ [HTTP 204](#)

💬 Phrase	No Content
💻 Server	nginx
🔗 Request in frame	2995
🕒 Time since request	147.274000 milliseconds
➡️ For	/api/metrics/run

[3021] ✉️ [HTTP 204](#)

💬 Phrase	No Content
💻 Server	nginx
🔗 Request in frame	2998
🕒 Time since request	149.693000 milliseconds
➡️ For	/api/metrics/run

[3027] ✉️ [HTTP 204](#)

💬 Phrase	No Content
💻 Server	nginx
🔗 Request in frame	3007
🕒 Time since request	148.281000 milliseconds
➡️ For	/api/metrics/run

[3036] ✉️ [HTTP 200](#)

💬 Phrase	OK
💻 Server	nginx

[Frame 3015] The first theft, 0.8 seconds after boot. A multipart/form-data upload to /contact carries the victim ID u, build ID b, three one-digit flags (l, n, p), and a 3,249-byte zip named like a hidden macOS temp file.

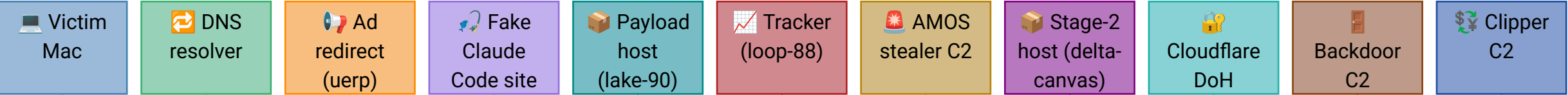
Exporting the zip from the capture shows what it holds so far: an info file (1,338 bytes), the username , the shell history, and empty folders for the apps it searched. The most valuable item, the user's password, is still missing; see frame 3157.

[Frame 3017] 204 for the browser's report (frame 2995).

[Frame 3021] 204 for the wallet's report (frame 2998).

[Frame 3027] 204 for the passmgr report (frame 3007).

[Frame 3036] The C2 server accepts the first upload with 200 OK and a 3-byte body, OK and a newline.



Content-Type	text/plain; charset=utf-8	
Content-Length	3	
Request in frame	3015	
Time since request	213.214000 milliseconds	
For	/contact	

[3151] 🍷 POST /api/metrics/run

Method	POST	
Host	152.42.143.45	
User-Agent	curl/8.7.1	
Content-Type	application/json	
Content-Length	207	
event	stage	
stage	resolve_auth	

[3157] 🍷 POST /contact

Method	POST	
Host	152.42.143.45	
User-Agent	curl/8.7.1	
Content-Length	4411	
Content-Type	multipart/form-data; boundary=-----m7tKak2CUgkzViUM0A93PB	
Form field	u	
Form field	b	
Form field	l	
Form field	n	
Form field	p	
File part	.A86Eo_A86ED3CE-61F0-40E0-A4F3-6F85D42B88DA.zip	

[3159] 📧 HTTP 204

Phrase	No Content	
Server	nginx	
Request in frame	3151	
Time since request	157.299000 milliseconds	
For	/api/metrics/run	

[3166] 📧 HTTP 200

Phrase	OK	
Server	nginx	

[Frame 3151]

Phase 4: The stolen password

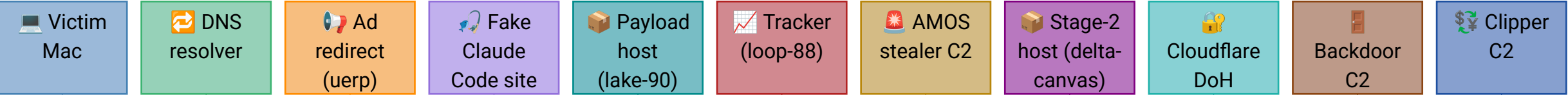
110 seconds of silence, then stage: **resolve_auth**. In the gap, the stealer showed the fake system dialog seen in Brad's screenshots, asking for the Mac's password. The pause is the time it took a person to read the dialog and type.

[Frame 3157] The password leaves the Mac. The second upload repeats the first, plus one new 9-byte file in the zip: pwd , the macOS login password typed into the fake dialog.

On a Mac, the login password also protects the user's keychain, the encrypted store of saved passwords. The next upload takes the keychain files.

[Frame 3159] 204 for the resolve_auth report (frame 3151).

[Frame 3166] 200 OK for the second upload.



 Content-Type	text/plain; charset=utf-8
 Content-Length	3
 Request in frame	3157
 Time since request	171.243000 milliseconds
 For	/contact

[3250] 🏠 POST /api/metrics/run

Method	POST
 Host	152.42.143.45
 User-Agent	curl/8.7.1
 Content-Type	application/json
 Content-Length	205
 event	stage
 stage	local_data

[3255] ✉ HTTP 204

 Phrase	No Content
 Server	nginx
 Request in frame	3250
 Time since request	149.781000 milliseconds
 For	/api/metrics/run

[3259] ✉ HTTP 100

 Phrase	Continue
--	----------

[6295] 🏠 POST /contact

Method	POST
 Host	152.42.143.45
 User-Agent	curl/8.7.1
 Content-Length	3072906
 Content-Type	multipart/form-data; boundary=-----1oiXjs4bJwHj3cXgzmgzXS
 Form field	u
 Form field	b
 Form field	l
 Form field	n
 File part	.A86Eo_A86ED3CE-61F0-40E0-A4F3-6F85D42B88DA.zip

[7668] ✉ HTTP 200

[Frame 3250]

Phase 5: The bulk upload

stage : local_data , 35 seconds later: the stealer has finished gathering files from the disk.

[Frame 3255] 204 for the local_data report.

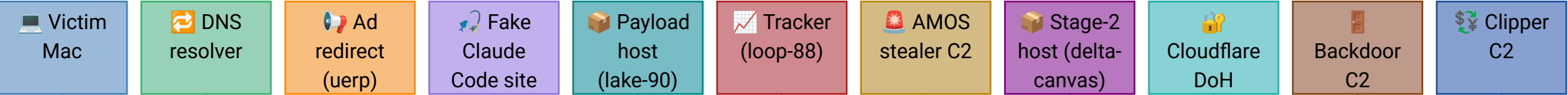
[Frame 3259] 100 Continue. For a large body, curl sends the headers first with Expect : 100-continue (frame 3253) and waits for the server's go-ahead. The headers already declare the size: Content-Length : 3072906 .

[Frame 6295] The bulk theft: 3 MB in 2.3 seconds. Wireshark completes this request after reassembling about 2,250 TCP segments. The zip, 3,072,183 bytes, adds the most damaging items:

- keychain-2.db with its WAL and shared-memory files and the user .kb keybag (about 2.6 MB): the keychain database. The password from frame 3157 is what the attackers need to try to decrypt it.
- Five screenshots from the desktop.
- A process list and a list of installed software, which show what security tools are running.
- Apple Notes' database files (empty on this test Mac).

All of it crosses the network unencrypted, which is why the full contents can be read from the capture.

[Frame 7668] 200 OK, 4.4 seconds after the last byte. Brad's notes show that the stealer then deleted its staging



Phrase	OK	
Server	nginx	
Content-Type	text/plain; charset=utf-8	
Content-Length	3	
Request in frame	6295	
Time since request	4.394177000 seconds	
For	/contact	

[7671] DNS Query

Name	delta-canvas.com
Type	A

[7672] DNS Response

Rcode	No error (0)
Name	delta-canvas.com
A	104.21.30.254
A	172.67.174.63

[7678] TLS Client Hello

SNI	delta-canvas.com
ALPN	h2
ALPN	http/1.1
JA4	t13d4907h2_0d8feac7bc37_7395dae3b2f3

[8166] TLS Client Hello

SNI	delta-canvas.com
ALPN	h2
ALPN	http/1.1
JA4	t13d4907h2_0d8feac7bc37_7395dae3b2f3

[8272] DNS Query

Name	cloudflare-dns.com
Type	A

folder and zip from / tmp , removing the evidence on disk.
The network evidence remains.

[Frame 7671]
Phase 6: Staying on the Mac

Within a second of the upload, the Mac looks up a new domain, delta-canvas [.].com. The stealing is done; this phase sets up persistence.

[Frame 7672] Another Cloudflare-fronted domain, the fourth in this infection.

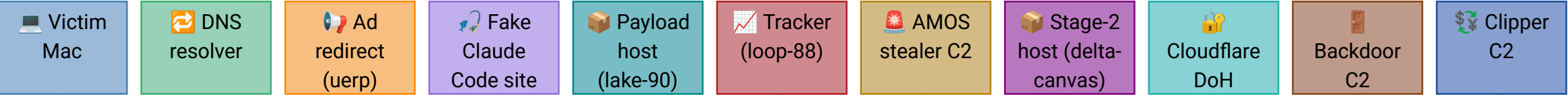
[Frame 7678] curl 's fingerprint again. This connection downloads about 0.54 MB on the wire, which fits the size of one of the two persistent programs Brad found, Accountshelper (504,752 bytes). The name mimics Apple's accountsd service.

[Frame 8166] A second download of about 0.57 MB on the wire, fitting the other persistent program, mdworker_shared (535,440 bytes), named after a real macOS Spotlight process.

Brad's notes show both installed as LaunchDaemons (/Library/LaunchDaemons/com.apple.accountsd.helper.plist and com.apple.metadata.mds.worker.plist) , which macOS starts at every boot, normally as root. Whether each download feeds the backdoor or the clipper is inferred from the order and sizes, not visible in the encrypted traffic.

[Frame 8272]
Phase 7: A backdoor checks in

The Mac looks up cloudflare-dns [.].com, Cloudflare's legitimate DNS-over-HTTPS (DoH) service. Browsers can use DoH, but the next Client Hello carries curl 's fingerprint, not Safari's.



[8273] DNS Response

Rcode	No error (0)
Name	cloudflare-dns.com
A	104.16.249.249
A	104.16.248.249

[8277] TLS Client Hello

SNI	cloudflare-dns.com
ALPN	h2
ALPN	http/1.1
JA4	t13d4907h2_0d8feac7bc37_7395dae3b2f3

[8587] POST /api/join/

Method	POST
Host	199.91.221.4
User-Agent	curl/8.7.1
Content-Length	75
Content-Type	application/x-www-form-urlencoded
Body	aG0d01B0TuWqI0bcCKz_iasaMoZv5GMqN0HtugY76Fo\nFF40665JVF\n3.0\ned75724b934625c3

[8754] HTTP 200

Phrase	OK
Server	nginx
Content-Type	text/plain
Content-Length	82
Request in frame	8587
Time since request	140.317000 milliseconds
For	/api/join/

[8771] GET /ws/sync/x3

Method	GET
Query	uid=9d745cbae23ac70dba18338b799896d3
Query	v=3.0
Query	s=49f2a2f25ae0afdc550b4bf8f89c15b4bc-cedcdab9aef116a928166daf3ac0c0
Host	199.91.221.4
Upgrade	websocket

[Frame 8273] Cloudflare's DoH addresses. The name is legitimate, so it would never appear on a blacklist.

[Frame 8277] curl's fingerprint, to cloudflare-dns [.]com. Within 0.3 seconds of this connection opening, the Mac contacts a new C2 address it never looked up through ordinary DNS. The likely explanation: the backdoor resolves its C2 server over encrypted DoH, hiding the lookup from the network's resolver and its logs.

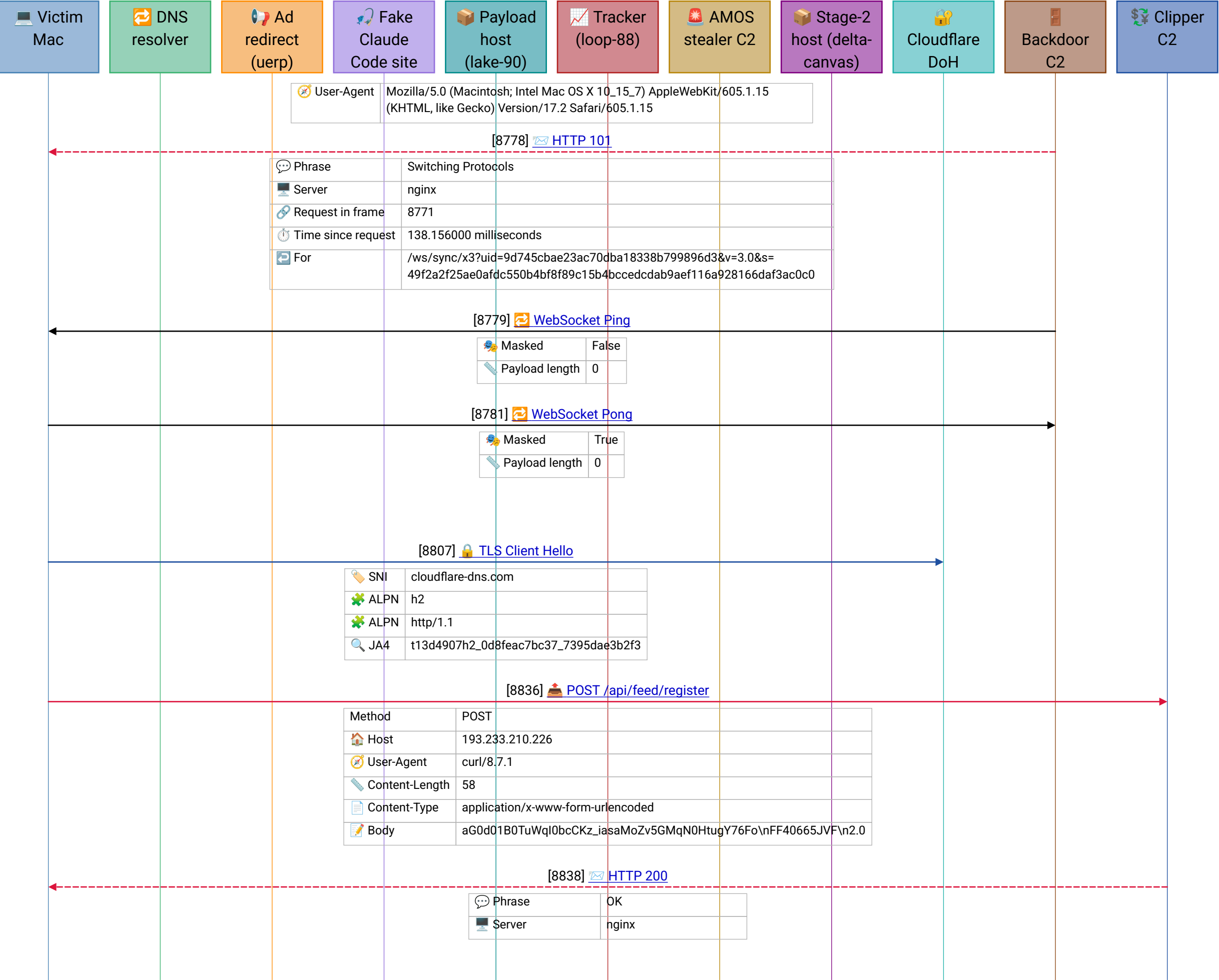
[Frame 8587] Backdoor registration. A POST /api/join/ to 199.91.221 [.]4, again plain HTTP to a bare IP. The body is four lines: the same victim ID u the stealer used in frame 2962, a 10-character identifier FF40665JVF, a version 3.0, and a 16-hex-digit value.

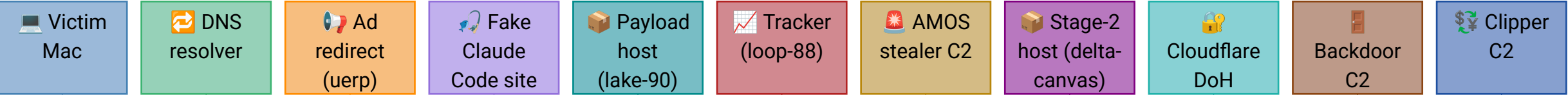
Reusing the stealer's victim ID ties the backdoor to the stealer infection on the operators' side.

[Frame 8754] The C2 server assigns the bot an ID, nXRcuuI6xw26GD0LeZiW0w, and a token on a second line. Watch for this ID: it returns 15 minutes later, in frame 27884.

[Frame 8771] A WebSocket with a borrowed identity. The backdoor asks to upgrade a new HTTP connection to a WebSocket at /ws/sync/x3, with a uid, a version, and a 64-hex-digit value s (likely a signature) in the query string.

Every request so far said curl/8.7.1. This one claims to be Safari 17.2, a version from December 2023, on a Mac that frame 2962 reported as running macOS 27. A copied browser User-Agent is a common disguise for non-browser traffic, and here the capture contradicts it.





	Content-Type	text/plain
	Content-Length	32
	Request in frame	8836
	Time since request	183.210000 milliseconds
	For	/api/feed/register

[8846] 🍷 GET /api/feed/items/f3aa8302e0e86cc3051267ac27b90755

Method	GET
Host	193.233.210.226
User-Agent	curl/8.7.1

[8850] ✉ HTTP 200

	Phrase	OK
	Server	nginx
	Content-Type	application/json
	Content-Length	1483
	Request in frame	8846
	Time since request	174.057000 milliseconds
	For	/api/feed/items/f3aa8302e0e86cc3051267ac27b90755
	Chain	ADA_STAKE
	Chain	AVAX_XP
	Chain	BCH_CASHADDR
	Chain	BNB_BEP2
	Chain	BTC_BECH32
	Chain	BTC_P2PKH
	Chain	BTC_P2SH
	Chain	BTC_TAPROOT...
		12 more

[27874] 📡 WebSocket Ping

	Masked	False
	Payload length	0

[27876] 📡 WebSocket Pong

	Masked	True
	Payload length	0

[Frame 8846] The implant fetches its feed: GET /api/feed/items/f3aa8302....

[Frame 8850] The clipper's rule list. The JSON reply (version : 5) holds 19 rules, each pairing an address format with an attacker-owned address. They cover 15 blockchains: Bitcoin has four rules, one per address format (legacy 1... , script 3... , SegWit bc1q... , Taproot bc1p...), Litecoin two, and Ethereum, Solana, Dogecoin, XRP, Tron, Cardano, Avalanche, and six other chains one each.

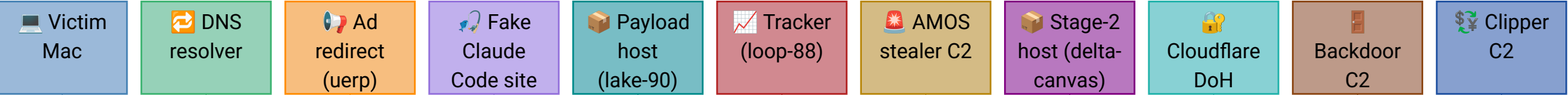
This is a **clipper**: malware that watches the clipboard and, when the user copies a wallet address, silently swaps in the attacker's address of the same format. The victim pastes it into a payment and sends funds to the attacker. Each format gets its own rule because a swapped address must match the format of the one the user copied.

[Frame 27874]

Phase 9: Fifteen minutes later

The 37th Ping , at 22:44:42, exactly 900 seconds (36 × 25 s) after the first. The heartbeat never missed. Meanwhile, the capture between these frames is Brad's ordinary browsing (weather, news, social sites), left out of the diagram.

[Frame 27876] The 37th Pong .



[27878] 🖱️ TCP FIN

📺 Stream	35
----------	----

[27884] 🍷 GET /api/tasks/nXRcuul6xw26GDOLeZiW0w

Method	GET	
🔍 Query	v=3.0	
🏠 Host	199.91.221.4	
🔑 User-Agent	curl/8.7.1	

[27886] ✉️ HTTP 200

💬 Phrase	OK		
💻 Server	nginx		
📄 Content-Type	text/plain		
📏 Content-Length	4		
🔗 Request in frame	27884		
🕒 Time since request	145.028000 milliseconds		
🔄 For	/api/tasks/nXRcuul6xw26GDOLeZiW0w?v=3.0		

[Frame 27878] The Mac closes the WebSocket itself, less than a second after its last pong: a plain TCP FIN, with no WebSocket close message. The socket was open for 900.9 seconds, almost exactly 15 minutes, which points to a timer rather than a failure.

[Frame 27884] 35 ms after the FIN, before the server has acknowledged it, the Mac opens a new connection to the same server. Over it, the backdoor polls over plain HTTP: GET /api/tasks/nXRcuuI6xw26GDOLeZiW0w, using the bot ID the C2 server assigned in frame 8754. The User-Agent is curl again.

[Frame 27886] The server's whole reply is one word, pong . With frame 27878, this looks like a 15-minute cycle: hold the WebSocket, drop it, check for queued tasks over HTTP, and probably reconnect. The pong could mean "no task", or it could simply answer a keepalive; one cycle cannot tell them apart, and the capture ends 17 seconds later, before any reconnect.

When the capture ends, the Mac is fully compromised: its passwords, keychain, and files gone; a backdoor waiting for orders; and a clipper ready to redirect any cryptocurrency payment.