A SMOOTH SEA NEVER MADE A SKILLED PHISHERMAN

DEEP DIVE INTO THE EVER-EVOLVING WORLD OF PHISHING

Kuba Gretzky

@mrgretzky

00 // WHOAMI

KUBA GRETZKY

Offensive Security Tools Developer

Ex-MMO Game Hacker

breakdev.org - offensive security blog

EVILGINX + **EVILGINX PRO** (coming soon)

pwndrop - dropbox for red teams

BREAKDEV RED - community for red teamers

Evilginx Mastery - phishing with Evilginx 101

@mrgretzky

00 // WHOAMI

IT STARTED @X33FCON

Lunchtime WiFi Hacking (7 years ago) 2017

Thank you

- Evilginx – http://bit.ly/evilginx
- Email: kuba@breakdev.org

01 // INTRO

WHAT IS THE TALK ABOUT?

- Defences against phishing are evolving
- Phishing is getting harder
- Black market phishing toolkits keep evolving
- Red teamers left alone in the dark with open-source tools

HELP?

BREAKDEV

evilginxpro

02 // EVILGINX PRO

ELEPHANT IN THE ROOM

- Bad guys like phishing
- Bad guys like free tools
- Red teams need to simulate bad guys
- Red teams need better tools
- Bad guys should not have better tools

BREAKDEV RED

02 // EVILGINX PRO

WHAT'S NEW?

CLIENT-SERVER ARCHITECTURE

- Evilginx server deployed as a daemon
- Evilginx client able to deploy servers and connect to them
- No need to SSH to each server
- Multi-user collaboration on servers
- Admin API carefully hidden behind HTTPS port 443
- Easy server deployment with several commands:

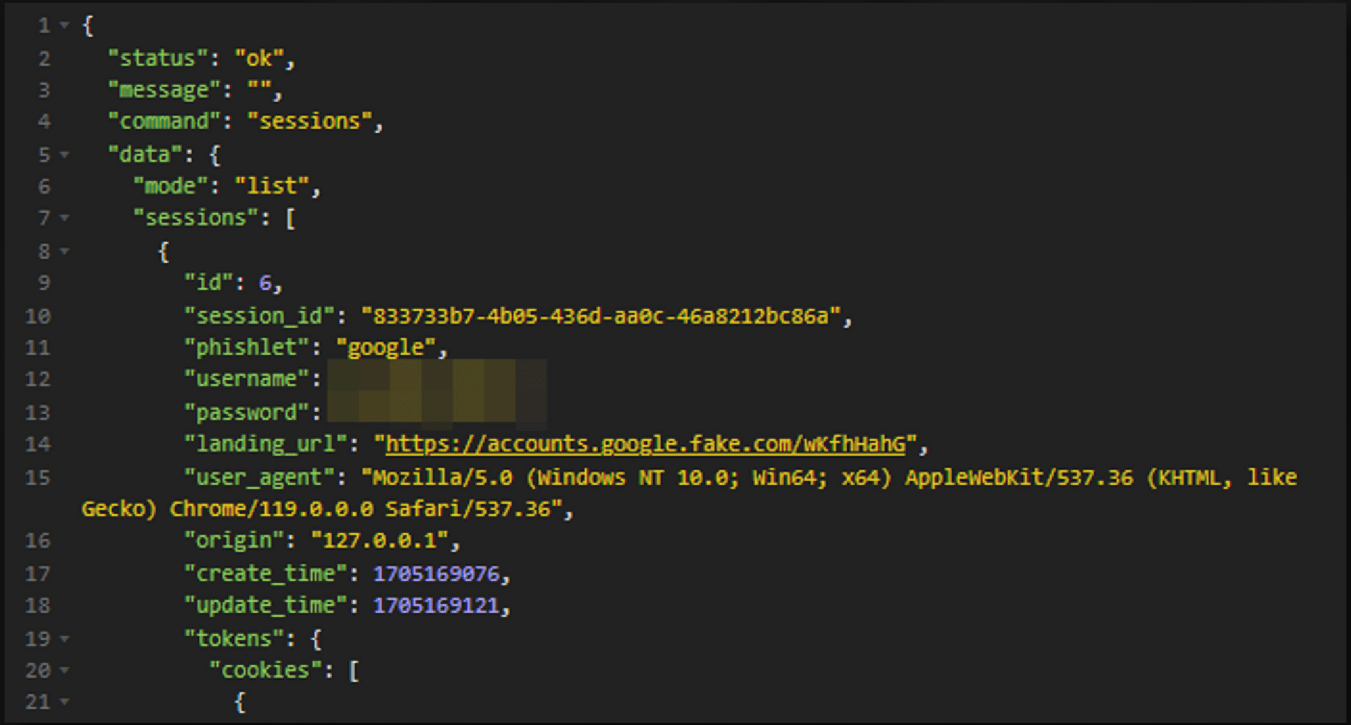
```
servers add evilx33f 1.2.3.4
servers register evilx33f
servers deploy evilx33f
```

02 // EVILGINX PRO

WHAT'S NEW?

CLIENT-SERVER ARCHITECTURE

- Evilginx API accessible via HTTPS requests or a persistent WebSockets connection:



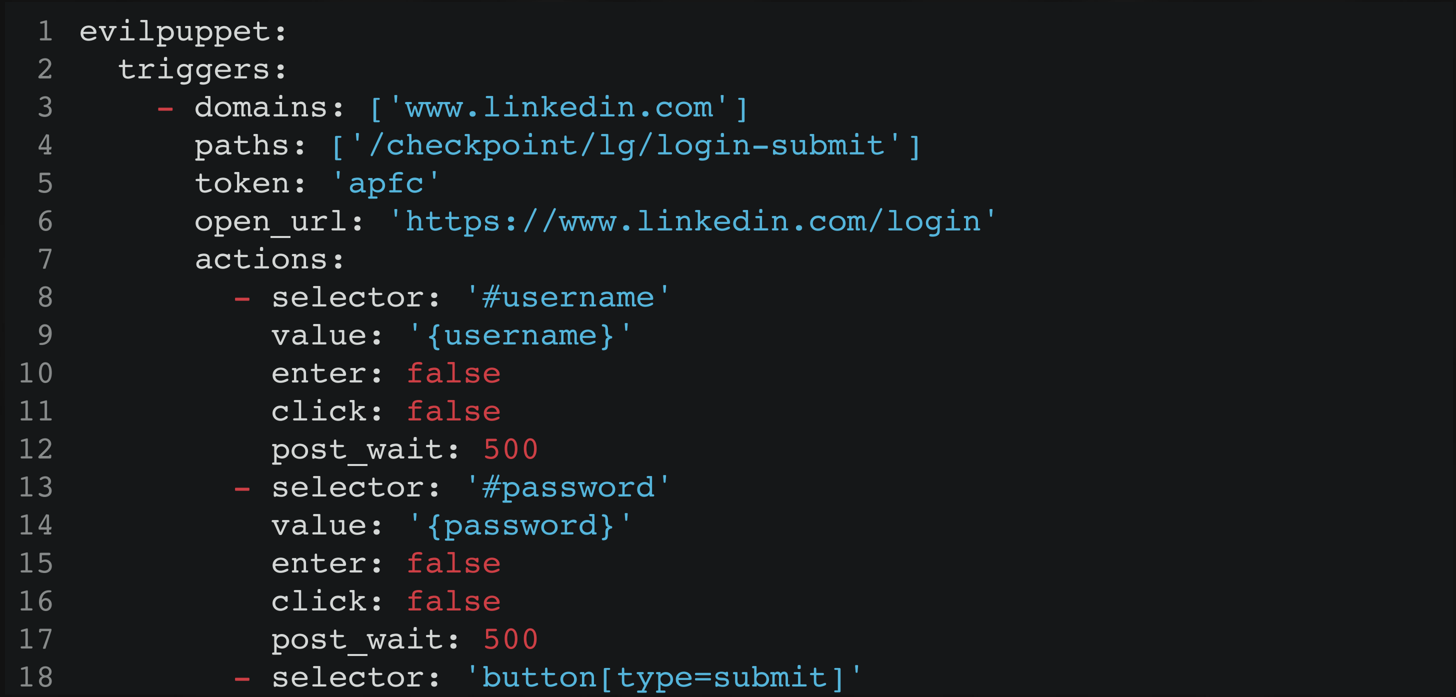
```
{
  "status": "ok",
  "message": "",
  "command": "sessions",
  "data": {
    "mode": "list",
    "sessions": [
      {
        "id": 6,
        "session_id": "833733b7-4b05-436d-aa0c-46a8212bc86a",
        "phishlet": "google",
        "username": [redacted]
        "password": [redacted]
        "landing_url": "https://accounts.google.fake.com/wKfhHahG",
        "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36",
        "origin": "127.0.0.1",
        "create_time": 1705169076,
        "update_time": 1705169121,
        "tokens": {
          "cookies": [
            {
```

02 // EVILGINX PRO

WHAT'S NEW?

EVILPUPPET

- Background browser controllable with phishlets
- Extraction of shadow tokens in real-time



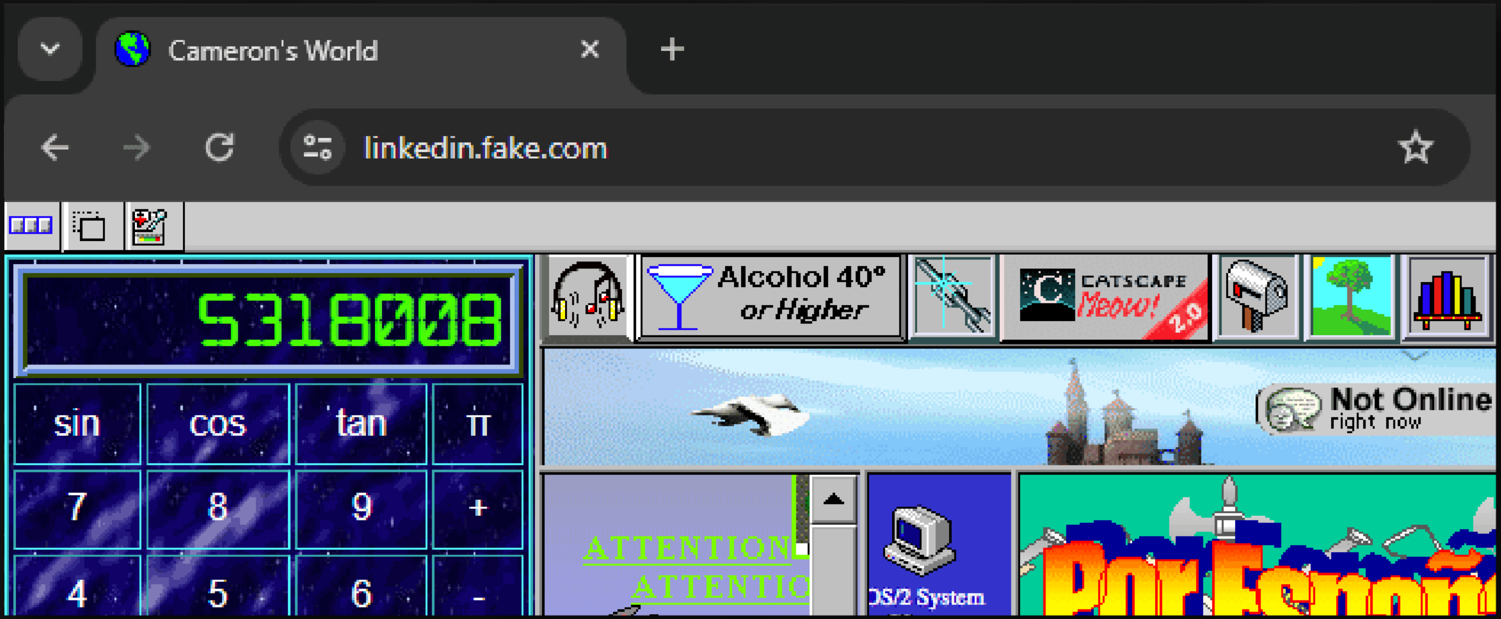
```yaml
evilpuppet:
  triggers:
    - domains: ['www.linkedin.com']
      paths: ['/checkpoint/lg/login-submit']
      token: 'apfc'
      open_url: 'https://www.linkedin.com/login'
      actions:
        - selector: '#username'
          value: '{username}'
          enter: false
          click: false
          post_wait: 500
        - selector: '#password'
          value: '{password}'
          enter: false
          click: false
          post_wait: 500
        - selector: 'button[type=submit]'
```

02 // EVILGINX PRO

WHAT'S NEW?

REVERSE PROXY AS A WEBSITE SPOOFER

- Display external websites in the context of the phishing domain
- Unauthorized clients will see a legitimate website under a phishing URL



BREAKDEV

02 // EVILGINX PRO

WHAT'S NEW?

TLS WILDCARD CERTIFICATES

- Automated retrieval and renewal
- Prevents exposing your phishing hostnames through TLS Transparency Log
- Scanners see TLS certificates registered for `*.phish.com` instead of `your.phish.com`

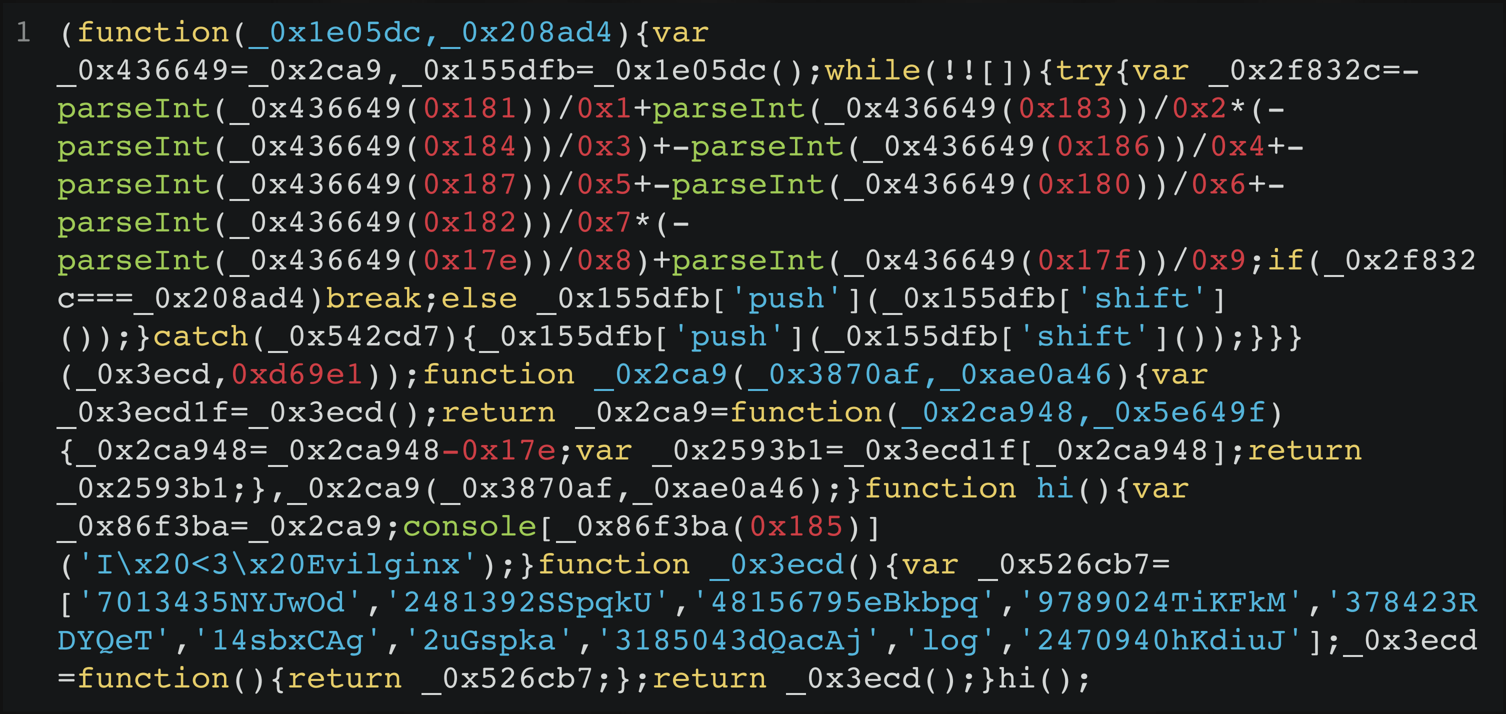
BREAKDEV

02 // EVILGINX PRO

WHAT'S NEW?

AUTOMATED JAVASCRIPT OBFUSCATION

- Auto-obfuscation for all injected scripts with `obfuscator.io` engine



```
(function(_0x1e05dc,_0x208ad4){var
_0x436649=_0x2ca9,_0x155dfb=_0x1e05dc();while(!![]){try{var _0x2f832c=-
parseInt(_0x436649(0x181))/0x1+parseInt(_0x436649(0x183))/0x2*(-
parseInt(_0x436649(0x184))/0x3)+-parseInt(_0x436649(0x186))/0x4+-
parseInt(_0x436649(0x187))/0x5+-parseInt(_0x436649(0x180))/0x6+-
parseInt(_0x436649(0x182))/0x7*(-
parseInt(_0x436649(0x17e))/0x8)+parseInt(_0x436649(0x17f))/0x9;if(_0x2f832
c===_0x208ad4)break;else _0x155dfb['push'](_0x155dfb['shift']
());}catch(_0x542cd7){_0x155dfb['push'](_0x155dfb['shift']());}}}
(_0x3ecd,0xd69e1));function _0x2ca9(_0x3870af,_0xae0a46){var
_0x3ecd1f=_0x3ecd();return _0x2ca9=function(_0x2ca948,_0x5e649f)
{_0x2ca948=_0x2ca948-0x17e;var _0x2593b1=_0x3ecd1f[_0x2ca948];return
_0x2593b1;},_0x2ca9(_0x3870af,_0xae0a46);}function hi(){var
_0x86f3ba=_0x2ca9;console[_0x86f3ba(0x185)]
('I\x20<3\x20Evilginx');}function _0x3ecd(){var _0x526cb7=
['7013435NYJwOd','2481392SSpqkU','48156795eBkbpq','9789024TiKFkM','378423R
DYQeT','14sbxCAg','2uGspka','3185043dQacAj','log','2470940hKdiuJ'];_0x3ecd
=function(){return _0x526cb7;};return _0x3ecd();}hi();
```

02 // EVILGINX PRO

WHAT'S NEW?

SQLITE DATABASE

- BuntDB no more
- Sorry, Melvin!

Bobber: https://github.com/Flangvik/Bobber

The TriForce of Initial Access:
https://trustedsec.com/blog/the-triforce-of-initial-access

02 // EVILGINX PRO

WHAT'S NEW?

EXTERNAL DNS MANAGEMENT

- Multiple domains support
- DNS zones controlled through external nameservers
  - Cloudflare
  - Digital Ocean
  - Route 53 (AWS)
- Plug & play different providers using **libdns** interface: https://github.com/libdns/libdns

BREAKDEV

02 // EVILGINX PRO

WHAT'S NEW?

JA4 SIGNATURE SPOOFING

- Spoofing the outbound TLS connection fingerprint

03 // JA4 SPOOFING

DESCRIPTION

> *“ JA4+ is a suite of network fingerprinting methods that are easy to use and easy to share. These methods are both human and machine readable to facilitate more effective threat-hunting and analysis.*

- Created by **John Althouse** from **Fox-IO**
- Successor to JA3
- Signature generated from TLS handshake Client Hello packet

https://blog.foxio.io/ja4+-network-fingerprinting

https://github.com/FoxIO-LLC/ja4

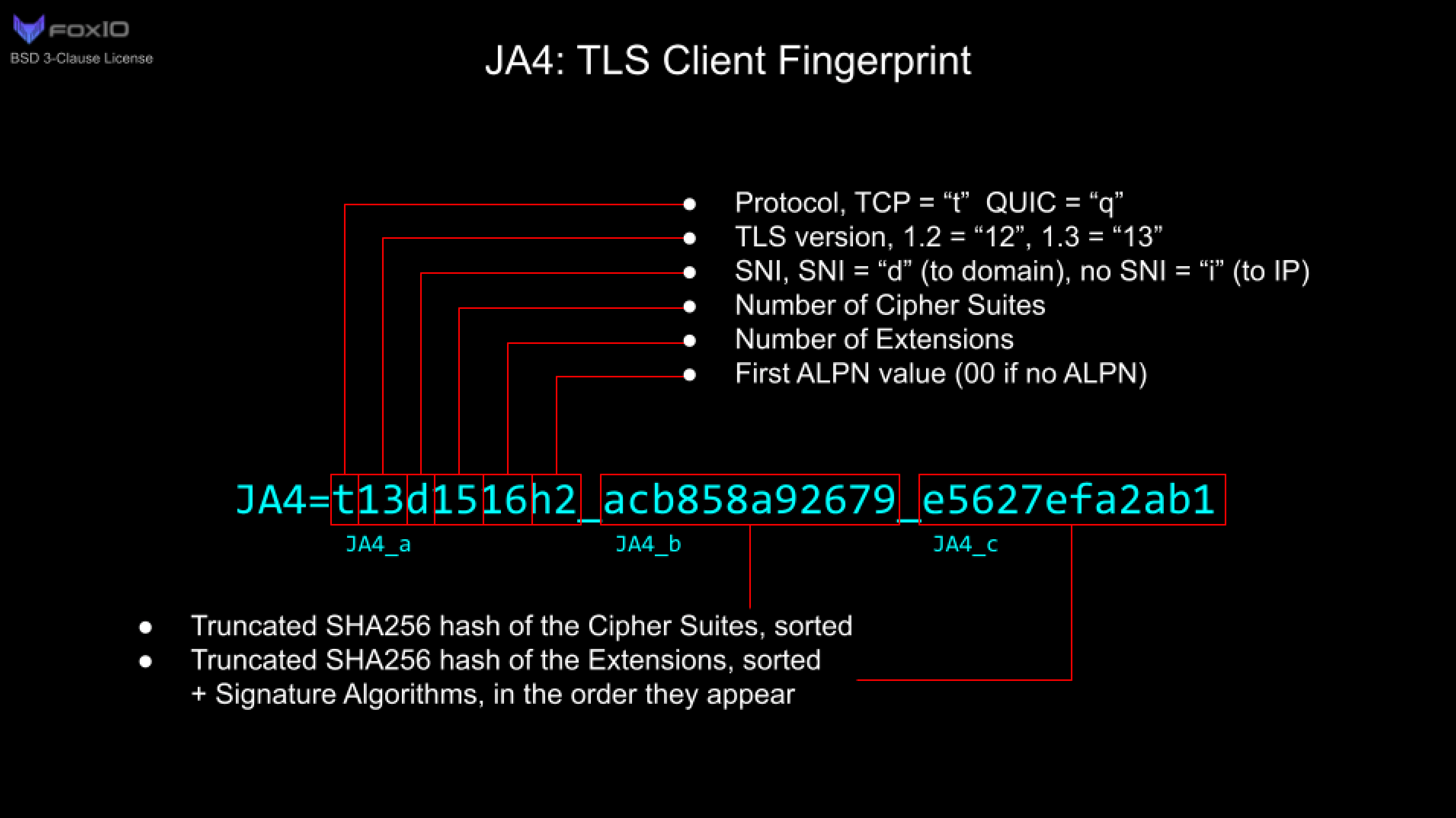
BREAKDEV

03 // JA4 SPOOFING

CLIENT HELLO

- Maximum supported TLS version
- ALPN (HTTP/2 or QUIC supported?)
- Supported cipher suites
- List of TLS extensions

03 // JA4 SPOOFING



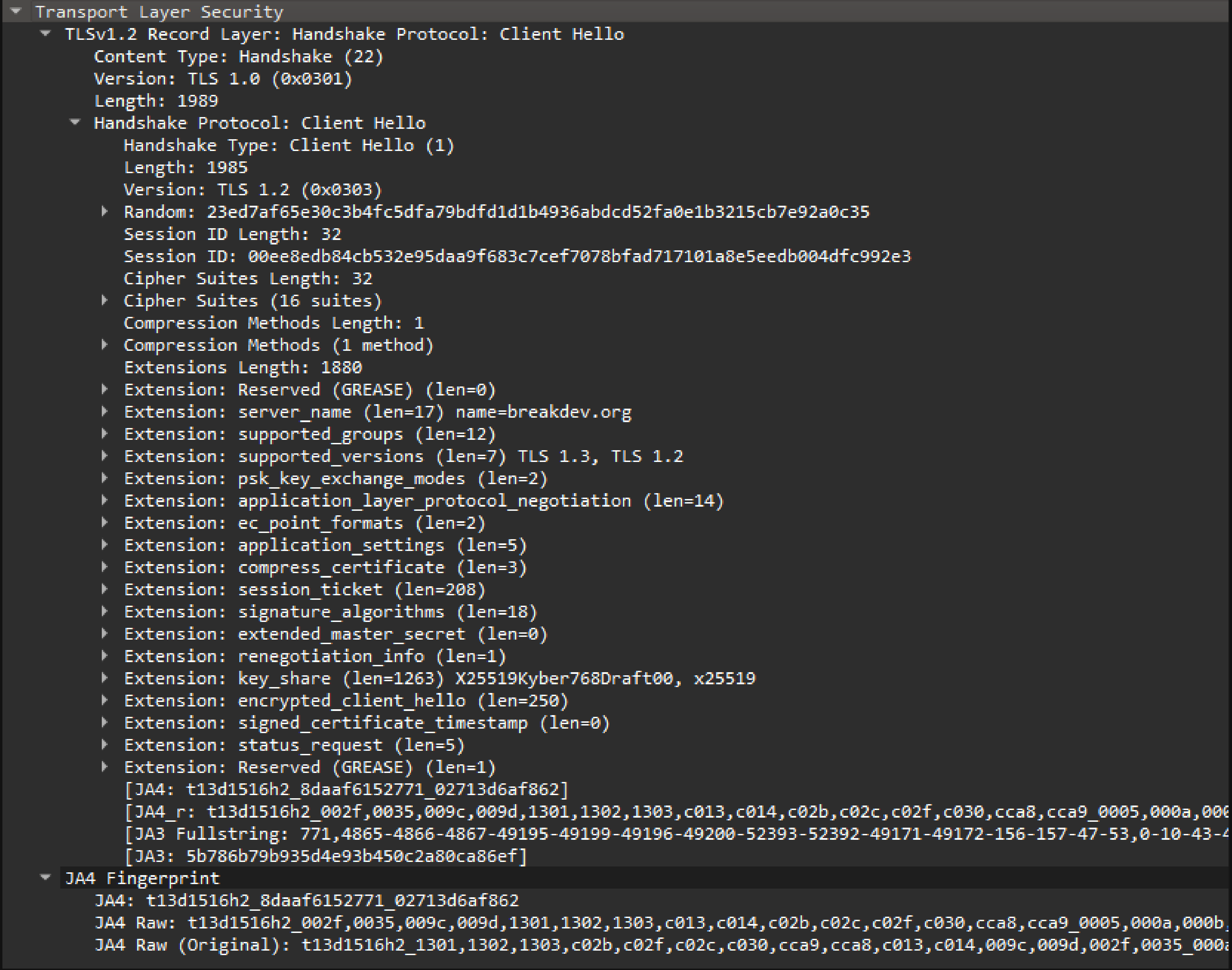
FoxIO

JA4: TLS Client Fingerprint

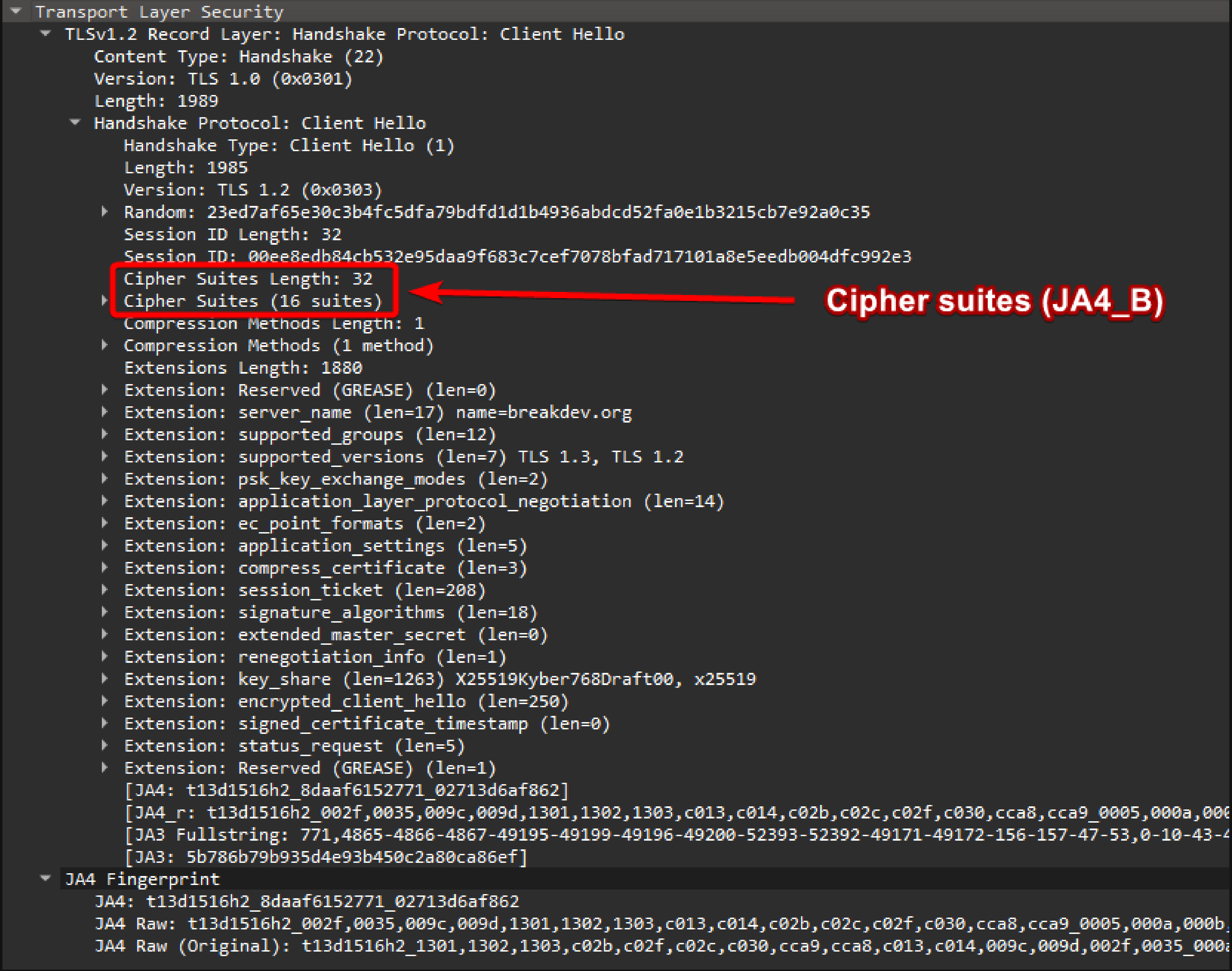
- Protocol, TCP = “t” QUIC = “q”
- TLS version, 1.2 = “12”, 1.3 = “13”
- SNI, SNI = “d” (to domain), no SNI = “i” (to IP)
- Number of Cipher Suites
- Number of Extensions
- First ALPN value (00 if no ALPN)

```
JA4=t13d1516h2_acb858a92679_e5627efa2ab1
```

- Truncated SHA256 hash of the Cipher Suites, sorted
- Truncated SHA256 hash of the Extensions, sorted
+ Signature Algorithms, in the order they appear

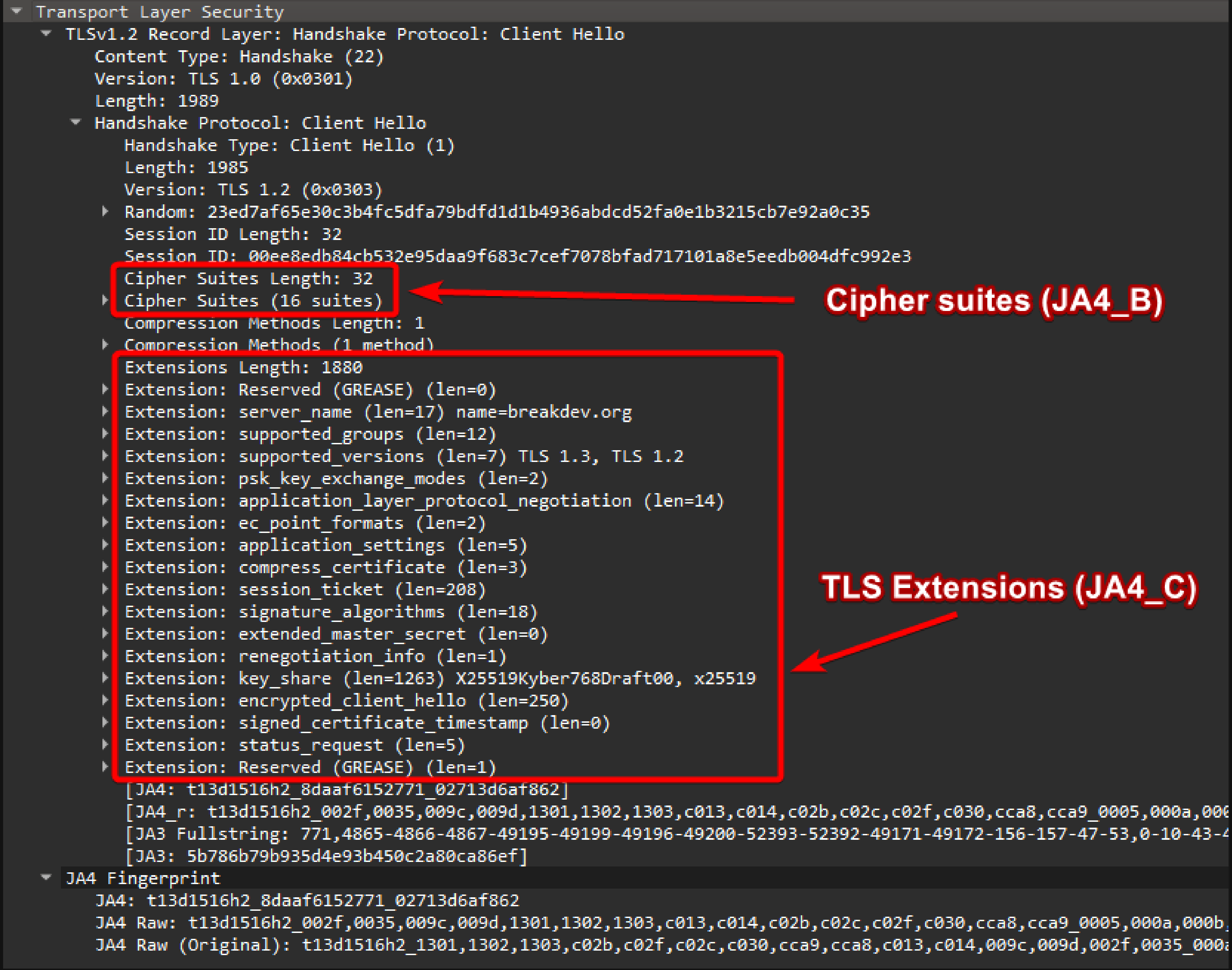


```
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 1989
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 1985
      Version: TLS 1.2 (0x0303)
      Random: 23ed7af65e30c3b4fc5dfa79bdfd1d1b4936abdcd52fa0e1b3215cb7e92a0c35
      Session ID Length: 32
      Session ID: 00ee8edb84cb532e95daa9f683c7cef7078bfad717101a8e5eedb004dfc992e3
      Cipher Suites Length: 32
      Cipher Suites (16 suites)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 1880
      Extension: Reserved (GREASE) (len=0)
      Extension: server_name (len=17) name=breakdev.org
      Extension: supported_groups (len=12)
      Extension: supported_versions (len=7) TLS 1.3, TLS 1.2
      Extension: psk_key_exchange_modes (len=2)
      Extension: application_layer_protocol_negotiation (len=14)
      Extension: ec_point_formats (len=2)
      Extension: application_settings (len=5)
      Extension: compress_certificate (len=3)
      Extension: session_ticket (len=208)
      Extension: signature_algorithms (len=18)
      Extension: extended_master_secret (len=0)
      Extension: renegotiation_info (len=1)
      Extension: key_share (len=1263) X25519Kyber768Draft00, x25519
      Extension: encrypted_client_hello (len=250)
      Extension: signed_certificate_timestamp (len=0)
      Extension: status_request (len=5)
      Extension: Reserved (GREASE) (len=1)
      [JA4: t13d1516h2_8daaf6152771_02713d6af862]
      [JA4_r: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000
      [JA3 Fullstring: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53,0-10-43-
      [JA3: 5b786b79b935d4e93b450c2a80ca86ef]
  JA4 Fingerprint
    JA4: t13d1516h2_8daaf6152771_02713d6af862
    JA4 Raw: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000b,
    JA4 Raw (Original): t13d1516h2_1301,1302,1303,c02b,c02f,c02c,c030,cca9,cca8,c013,c014,009c,009d,002f,0035_000a
```



```
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 1989
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 1985
      Version: TLS 1.2 (0x0303)
      Random: 23ed7af65e30c3b4fc5dfa79bdfd1d1b4936abdcd52fa0e1b3215cb7e92a0c35
      Session ID Length: 32
      Session ID: 00ee8edb84cb532e95daa9f683c7cef7078bfad717101a8e5eedb004dfc992e3
      Cipher Suites Length: 32
      Cipher Suites (16 suites)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 1880
      Extension: Reserved (GREASE) (len=0)
      Extension: server_name (len=17) name=breakdev.org
      Extension: supported_groups (len=12)
      Extension: supported_versions (len=7) TLS 1.3, TLS 1.2
      Extension: psk_key_exchange_modes (len=2)
      Extension: application_layer_protocol_negotiation (len=14)
      Extension: ec_point_formats (len=2)
      Extension: application_settings (len=5)
      Extension: compress_certificate (len=3)
      Extension: session_ticket (len=208)
      Extension: signature_algorithms (len=18)
      Extension: extended_master_secret (len=0)
      Extension: renegotiation_info (len=1)
      Extension: key_share (len=1263) X25519Kyber768Draft00, x25519
      Extension: encrypted_client_hello (len=250)
      Extension: signed_certificate_timestamp (len=0)
      Extension: status_request (len=5)
      Extension: Reserved (GREASE) (len=1)
      [JA4: t13d1516h2_8daaf6152771_02713d6af862]
      [JA4_r: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000
      [JA3 Fullstring: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53,0-10-43-
      [JA3: 5b786b79b935d4e93b450c2a80ca86ef]
  JA4 Fingerprint
    JA4: t13d1516h2_8daaf6152771_02713d6af862
    JA4 Raw: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000b,
    JA4 Raw (Original): t13d1516h2_1301,1302,1303,c02b,c02f,c02c,c030,cca9,cca8,c013,c014,009c,009d,002f,0035_000a
```

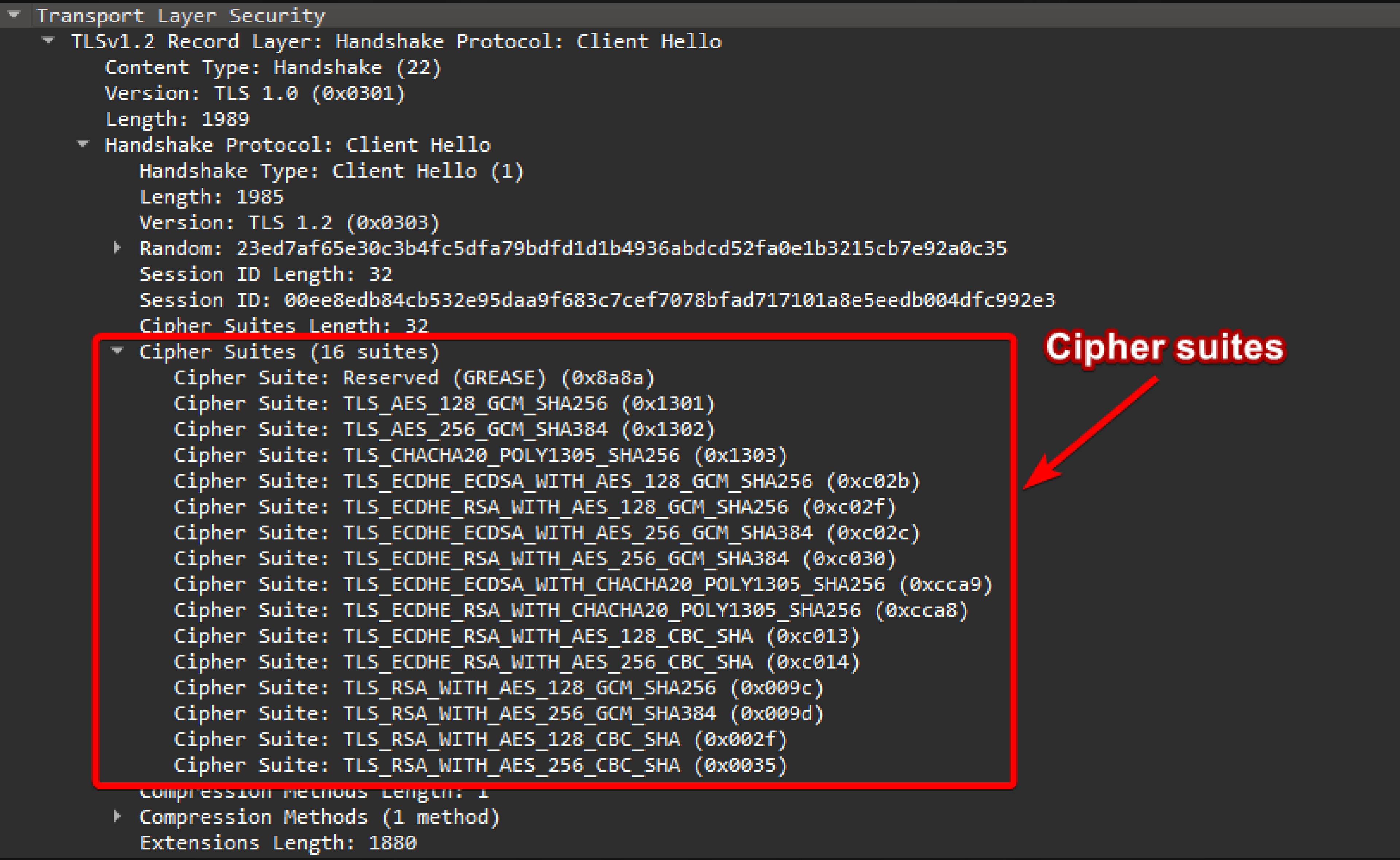
Cipher suites (JA4_B)



```
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1989
    ▾ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 1985
        Version: TLS 1.2 (0x0303)
      ▸ Random: 23ed7af65e30c3b4fc5dfa79bdfd1d1b4936abdcd52fa0e1b3215cb7e92a0c35
        Session ID Length: 32
        Session ID: 00ee8edb84cb532e95daa9f683c7cef7078bfad717101a8e5eedb004dfc992e3
        Cipher Suites Length: 32
      ▸ Cipher Suites (16 suites)
        Compression Methods Length: 1
      ▸ Compression Methods (1 method)
        Extensions Length: 1880
      ▸ Extension: Reserved (GREASE) (len=0)
      ▸ Extension: server_name (len=17) name=breakdev.org
      ▸ Extension: supported_groups (len=12)
      ▸ Extension: supported_versions (len=7) TLS 1.3, TLS 1.2
      ▸ Extension: psk_key_exchange_modes (len=2)
      ▸ Extension: application_layer_protocol_negotiation (len=14)
      ▸ Extension: ec_point_formats (len=2)
      ▸ Extension: application_settings (len=5)
      ▸ Extension: compress_certificate (len=3)
      ▸ Extension: session_ticket (len=208)
      ▸ Extension: signature_algorithms (len=18)
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
      ▸ Extension: key_share (len=1263) X25519Kyber768Draft00, x25519
      ▸ Extension: encrypted_client_hello (len=250)
      ▸ Extension: signed_certificate_timestamp (len=0)
      ▸ Extension: status_request (len=5)
      ▸ Extension: Reserved (GREASE) (len=1)
        [JA4: t13d1516h2_8daaf6152771_02713d6af862]
        [JA4_r: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000
        [JA3 Fullstring: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53,0-10-43-4
        [JA3: 5b786b79b935d4e93b450c2a80ca86ef]
  ▾ JA4 Fingerprint
      JA4: t13d1516h2_8daaf6152771_02713d6af862
      JA4 Raw: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000b,
      JA4 Raw (Original): t13d1516h2_1301,1302,1303,c02b,c02f,c02c,c030,cca9,cca8,c013,c014,009c,009d,002f,0035_000a
```

Cipher suites (JA4_B)

TLS Extensions (JA4_C)



```
▼ Transport Layer Security
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1989
    ▼ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 1985
        Version: TLS 1.2 (0x0303)
      ▶ Random: 23ed7af65e30c3b4fc5dfa79bdfd1d1b4936abdcd52fa0e1b3215cb7e92a0c35
        Session ID Length: 32
        Session ID: 00ee8edb84cb532e95daa9f683c7cef7078bfad717101a8e5eedb004dfc992e3
        Cipher Suites Length: 32
      ▼ Cipher Suites (16 suites)
          Cipher Suite: Reserved (GREASE) (0x8a8a)
          Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
          Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
          Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
          Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
          Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
          Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Compression Methods Length: 1
      ▶ Compression Methods (1 method)
        Extensions Length: 1880
```

Cipher suites

03 // JA4 SPOOFING

SIGNATURE GENERATION

JA4:

```
t13d1516h2_8daaf6152771_02713d6af862
```

JA4 Raw:

```
t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013
,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000b
,000d,0012,0017,001b,0023,002b,002d,0033,4469,fe0d
,ff01_0403,0804,0401,0503,0805,0501,0806,0601
```

BREAKDEV

04 // THE HUNT FOR EVILGINX

SCOUTING FOR PREY

Application	JA4
Google Chrome	t13d1516h2_8daaf6152771_02713d6af862 (TCP) q13d0312h3_55b375c5d22e_06cda9e17597 (QUIC)
Mozilla Firefox	t13d1715h2_5b57614c22b0_7121afd63204
Safari	t13d2014h2_a09f3c656075_14788d8d241b
IcedID Malware	t13d201100_2b729b4bf6f3_9e7b989ebec8
Sliver Malware	t13d190900_9dc949149365_97f8aa674fd9
SoftEther VPN	t13d880900_fcb5b95cb75a_b0d3b4ac2a14
Evilginx	t13d191000_9dc949149365_e7c285222651

BREAKDEV

04 // THE HUNT FOR EVILGINX

SCOUTING FOR PREY

Common JA4_B signatures:

- Google Chrome: `8daaf6152771`
- Golang (Sliver, Evilginx): `9dc949149365`

Cloudflare uses JA3/JA4:

https://developers.cloudflare.com/bots/concepts/ja3-ja4-fingerprint/

04 // THE HUNT FOR EVILGINX

WHAT CAN BE DONE?

SPOOF TLS CLIENT CONFIG

- Modify the list of supported TLS ciphers
- Use random TLS configurations with uTLS library: https://github.com/refraction-networking/utls
  - Different JA4 signature with every TLS connection
  - Good to avoid JA4 blacklists
  - Enough until defenders deploy more advanced detections
- Copy TLS configuration directly from client connecting to the proxy

BREAKDEV

What if we could harness the power of JA4 and use it to our advantage?

And make...

THE HUNTERS BECOME THE HUNTED

LOOK AT ME

LOOK AT ME
I AM THE DEFENDER NOW

WELL, WELL, WELL,
HOW THE TURNTABLES...

05 // THE HUNT FOR BOTS

THE WHEEL REINVENTED

- Cloudflare is already great at it
- Cloudflare Turnstile as Evilginx redirector:
  https://github.com/kgretzky/evilginx2/blob/master/redirectors/turnstile/index.html
- Why not implement our own botguard?

BEHOLD

POOR MAN'S CLOUDFLARE

05 // THE HUNT FOR BOTS

PREPARATIONS

- Forked `go-vhost` library used to extract hostnames from the TLS ClientHello packet's SNI extension data https://github.com/inconshreveable/go-vhost
- Added code to generate JA4 signatures for every connection
- Set up database logging of JA4 and User-Agent for every unauthorized request
- Disabled usage of wildcard certificates to trigger as many scans as possible
- Uploaded the phishing link to any URL scanning service I could find
- I gathered data for one month

05 // THE HUNT FOR BOTS

RESULTS

- 820 requests
- 680 unique IPs (IP blacklisting is dead)
- 52 different ASNs (database available for free on IPinfo.io)
- Most popular **JA4_B** signatures:

JA4_B	Name	Count	Percentage
8daaf6152771	**Google Chrome**	650	80%
9dc949149365	**Golang**	90	11%
e8a523a41297	**Googlebot**	10	1%

IPInfo.io: https://ipinfo.io/

05 // THE HUNT FOR BOTS

RESULTS

- Partial failure
- JA4 signatures are not enough to detect bots
- Most bots use the Chromium engine (headless browsers)

05 // THE HUNT FOR BOTS

NEW IDEA

- Phished users must have JavaScript enabled
- Safe to assume JavaScript will always be available
- How many bots are able to run JavaScript?

05 // THE HUNT FOR BOTS

GATHERING BROWSER TELEMETRY

- JavaScript injected into the landing page (the reverse proxied spoofed page) to gather browser telemetry
- Telemetry sent back to the Evilginx server for analysis
- If authorized, Evilginx redirects to the phishing page

Q: How many page views out of 820 resulted in telemetry data being sent back to the Evilginx server?

A: ~35 🤦‍♂️

Fp-Collect: https://github.com/antoinevastel/fp-collect

BREAKDEV

05 // THE HUNT FOR BOTS

ANALYZING BROWSER TELEMETRY

- Decided to go for the low-hanging fruit:
  - Browser window size
  - User-Agent
- Used `ua-parser-js` library for analyzing User-Agents: https://github.com/faisalman/ua-parser-js

05 // THE HUNT FOR BOTS

WINDOW SIZE ANALYSIS

SCREEN SIZES

```
"wInnerHeight": 1200,
"wInnerWidth": 1600,
"wOuterHeight": 1200,
"wOuterWidth": 1600,
```

- Possible only while browser is in fullscreen mode
- Unlikely anyone would be opening a phishing link while in fullscreen mode

05 // THE HUNT FOR BOTS

WINDOW SIZE ANALYSIS

OUTER WINDOW SMALLER THAN INNER WINDOW

```
"wDevicePixelRatio": 1,
"wInnerHeight": 768,
"wInnerWidth": 1024,
"wOuterHeight": 600,
"wOuterWidth": 800,
```

- Possible only when zoomed out (Control+'-')
- This should be reflected with `wDevicePixelRatio < 1`, but never is
- Unlikely anyone would be zoomed out when opening a new link

BREAKDEV

05 // THE HUNT FOR BOTS

WINDOW SIZE ANALYSIS

UNREALISTIC WINDOW SIZES

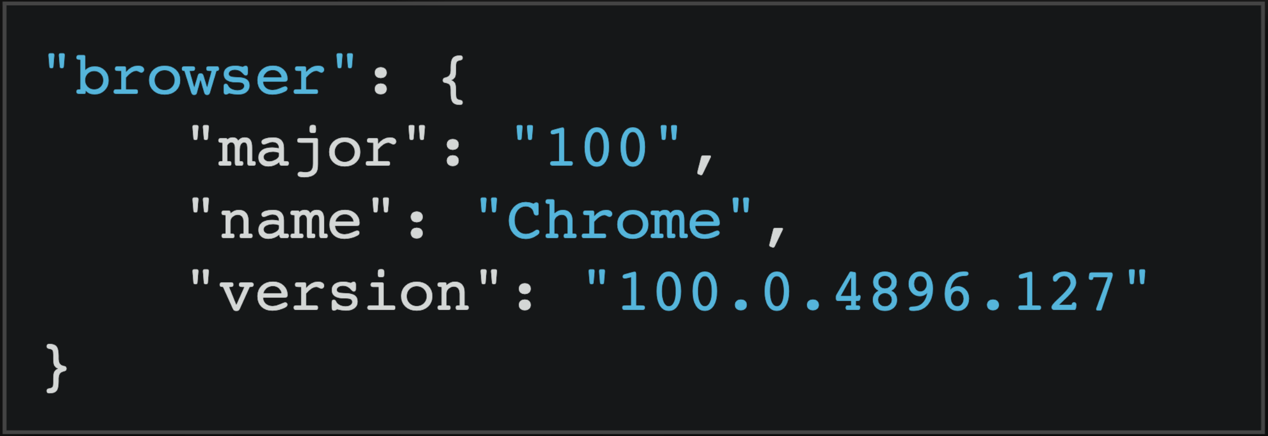
```
"wDevicePixelRatio": 1,
"wInnerHeight": 768,
"wInnerWidth": 1366,
"wOuterHeight": 1,
"wOuterWidth": 1,
```

- Outer window unnaturally small

05 // THE HUNT FOR BOTS

BROWSER VERSION ANALYSIS

OUTDATED VERSIONS



```
"browser": {
    "major": "100",
    "name": "Chrome",
    "version": "100.0.4896.127"
}
```

- Almost every single bot used a browser version older than 6 months

05 // THE HUNT FOR BOTS

INTERESTING CASES

SAFARI ON IPHONE

```
"browser": {
    "major": "17",
    "name": "Mobile Safari",
    "version": "17.4"
}
```

WINDOW DIMENSIONS LOOKING GOOD

```
"wDevicePixelRatio": 3,
"wInnerHeight": 664,
"wInnerWidth": 390,
"wOuterHeight": 664,
"wOuterWidth": 390,
```

05 // THE HUNT FOR BOTS

INTERESTING CASES

VIDEO CARD (?!)

```
"videoCard": [
    "Google Inc. (Google)",
    "ANGLE (Google, Vulkan 1.3.0 (SwiftShader Device (Subzero)
(0x0000C0DE)), SwiftShader driver)"
]
```

JA4: 8daaf6152771 (Google Chrome)

Safari - really?!

The real detection power comes from cross-checking the data from all the sensors

05 // THE HUNT FOR BOTS

EVILGINX PRO BOTGUARD

```
min_ver: '4.0.0'
ja4:
  # if 'allow' is defined, whitelist mode is activated
  allow:
  deny:
    - {b: 'e8f1e7e78f70'}
    - {b: '9dc949149365'} # golang
    - {b: 'cbb2034c60b8'} # golang 1.22
    - {b: 'c7886603b240'} # Python requests 3.10
    - {b: '730fb1b0ac6a'} # Python requests 2.27
    - {b: 'e8a523a41297'} # Googlebot
    - {b: '1ce71f0edbb1'} # Java 8.0
    - {b: '231e334592e8'} # bingbot
    - {b: '2b729b4bf6f3'} # bingbot
    - {b: '76e208dd3e22'} # curl
user_agent:
  # if 'allow' is defined, whitelist mode is activated
  allow:
    - {browser: 'Chrome', version: '>= 120'}
    - {browser: 'Firefox', version: '>= 120.0'}
    - {browser: 'Edge', version: '>= 120.0'}
    - {browser: 'Opera', version: '>= 120.0'}
    - {browser: 'Safari', version: '>= 16.0'}
  deny:
    - {browser: 'Headless'}
```

BREAKDEV

BREAKDEV RED

Contact:

kuba@breakdev.org

Subject:

X33FCON - BREAKDEV

evilginxpro

COMING SOON

(2024)

@mrgretzky

THANK YOU

Questions?

@mrgretzky