ENCRYPTED KEYLOGGER AND REVERSE SHELL

נעם כרמון ומנחם רוזנטל

WHAT IS KEYSTROKE LOGGING?

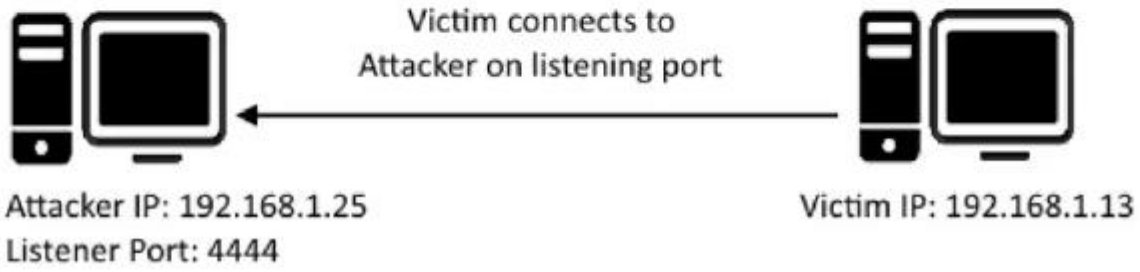
קיילוגר זה תוכנה שרצה ברקע ושומרת את כל הקלט של המקלדת.
לאחר שנקלט המידע מהמקלדת המידע נשמר במחשב או נשלח מיד לתוקף.
התוקף יכול לאחר מכן לעבור על המידע ולחפש סיסמאות, מידע אישי או כל דבר אחר שיכול להיות שמיש.

REVERSE SHELL

קשה לפרוץ לתוך מחשב או רשת.
לכן, המטרה היא לגרום לנתקף להתחבר אלינו. כך שאם אנחנו מצליחים להשחיל קוד זדוני במחשב של הנתקף אז הוא יפנה אלינו מעצמו ויאפשר לנו גישה מבפנים.
לאחר מכן אנחנו יכולים לשלוט מרחוק.

Reverse Shell

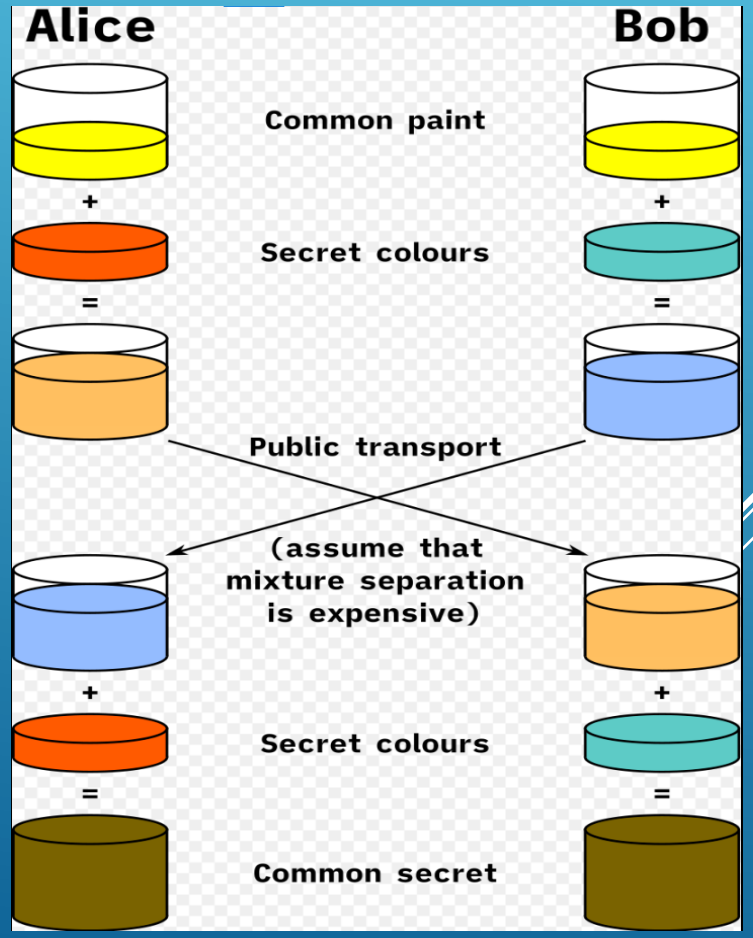
A reverse shell is a type of shell in which the target machine communicates back to the attacking machine. The attacking machine has a listener port on which it receives the connection, which by using, code or command execution is achieved.



ENCRYPTION

על מנת שהמתקפה שלנו תהיה 'מאובטחת' אנחנו רוצים להצפין את המידע כך שגם אם יבינו שיש מתקפה או שבמקרה מאזינים לקו, לא יוכלו לדעת איזה פקודות הריצו או איזה מידע הצלחנו לגנוב.

Diffie–Hellman



MULTIPLE KEYS

הסרבר פותח סוקט שיכול להתחבר אליו כמה נתקפים, כלומר קבוצה של נתקפים. מול כל אחד יש החלפת מפתחות כך שיש מפתח ייחודי להצפנת התקשורת עם כל נתקף. במקרה כזה גם אם יצליחו איך שהוא לפצח מפתח ספציפי, זה לא פוגע בתקשורת עם כל שאר הנתקפים.

Packet fragmentation

כדי להישאר חשאים ושלא ישימו לב שהם נתקפים, שלחנו את הפקטות בחלקים על מנת לא ליצור עומס ברשת.

עוד יתרון, אם מישהו מאזין לרשת יהיה הרבה יותר קשה לפענח את ההצפנה כשהוא מקבל רק חלקי הודעות.

מבנה הקוד

server

הסרבר פותח סוקט שיכולים להתחבר אליו כמה משתמשים (נתקפים), מול כל אחד הוא מבצע החלפת מפתחות.
הסרבר מקבל את המידע שמוזרם אליו מהנתקף ובמקביל יכול לבצע פקודות דרך המחשב השני.

```
def send_command_to_client(client_address, command):...

def reverse_shell():...

def delete_client(client_port):...

# this function creates a Diffie-Hellman key pair
def create_dh_key():...

# initialize the handshake with the client
def handshake(client_socket, client_address):...

def get_key_logger_pic(data, client_port):...

def get_key_logger_data(data, client_port):...

# this function handles the client
def handle_client(client_socket, client_address):...

def main():...
```

מבנה הקוד

client

הקליינט מתחבר לסוקט של הסרבר ומתחיל להזרים לו את המידע שנקלט מהמחשב.
במקביל הוא מריץ את הפקודות שהוא מקבל.

```
# Create a Diffie-Hellman client object
def create_dh_key(param_bytes):...

def initialize_handshake(client_socket):...

def key_pressed(key):...

def key_logger(client_socket, shared_key):...

def main():...
```

מבנה הקוד

protocol

בפרוטוקול אנחנו מצפינים ומפענחים את ההודעות עם המפתח, ובנוסף אנחנו שולחים את ההודעות בחלקים.

```
def create_msg(type_msg, message, shared_key):...

def send_message(my_socket, type_msg, message, shared_key):...

def send_message_in_parts(my_socket, message):...

def get_type(my_socket, shared_key):...

def get_msg_in_parts(my_socket, length, shared_key, picture=False):...

def get_msg(my_socket, shared_key, picture=False):...

def decode_picture(message, shared_key):...

# this function decode the message from the client
def decode_message(message, shared_key):...

# this function encrypt the message to the client
def encode_message(message, shared_key):...

# this function decode the message from the client
def decode_message_with_unpadding(message, shared_key):...
```

שליחת הפרמטרים

Wireshark · Packet 49 · Adapter for loopback traffic capture

```
> Frame 49: 289 bytes on wire (2312 bits), 289 bytes captured (2312 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 8822, Dst Port: 65454, Seq: 1, Ack: 1, Len: 245
v Data (245 bytes)
    Data: 2d2d2d2d2d424547494e20444820504152414d45544552532d2d2d2d2d0a4d494748416f…
    [Length: 245]
```

```
0000  02 00 00 00 45 00 01 1d  6f d5 40 00 80 06 00 00   ····E··· o·@·····
0010  7f 00 00 01 7f 00 00 01  22 76 ff ae d7 f9 0e a5   ····· "v·····
0020  42 98 23 9c 50 18 27 f9  9f 19 00 00 2d 2d 2d 2d   B·#·P·'· ····----
0030  2d 42 45 47 49 4e 20 44  48 20 50 41 52 41 4d 45   -BEGIN D H PARAME
0040  54 45 52 53 2d 2d 2d 2d  2d 0a 4d 49 47 48 41 6f   TERS---- ··MIGHAo
0050  47 42 41 50 33 56 31 78  67 61 43 56 6e 6a 34 78   GBAP3V1x gaCVnj4x
0060  35 31 4d 34 6d 55 56 74  61 79 49 36 65 4c 6a 41   51M4mUVt ayI6eLjA
0070  55 64 43 45 56 4e 63 47  7a 57 69 61 56 39 70 65   UdCEVNcG zWiaV9pe
0080  58 6b 45 6f 4e 4f 39 48  2f 6e 0a 6e 34 75 42 6e   XkEoNO9H /n·n4uBn
0090  47 78 47 75 32 47 72 4c  49 30 70 79 34 55 5a 51   GxGu2GrL I0py4UZQ
00a0  77 53 73 6b 7a 2f 45 4b  38 36 2b 6b 69 5a 4b 55   wSskz/EK 86+kiZKU
00b0  6a 71 76 49 6a 4b 69 59  53 34 72 6f 4c 42 61 44   jqvIjKiY S4roLBaD
00c0  69 65 75 4f 37 63 76 54  2b 79 4e 0a 65 50 70 2b   ieuO7cvT +yN·ePp+
00d0  38 48 45 2f 35 32 49 7a  39 68 78 39 70 72 6a 70   8HE/52Iz 9hx9prjp
00e0  59 7a 6c 65 77 71 4e 62  6d 71 6c 76 6d 55 65 35   YzlewqNb mqlvmUe5
00f0  4a 79 37 6c 6f 4e 65 79  57 6c 4f 49 6c 5a 30 58   Jy7loNey WlOIlZ0X
0100  41 67 45 43 0a 2d 2d 2d  2d 2d 45 4e 44 20 44 48   AgEC···· ··END DH
0110  20 50 41 52 41 4d 45 54  45 52 53 2d 2d 2d 2d 2d    PARAMET ERS-----
```

☑ Show packet bytes

Server public key

Wireshark · Packet 51 · Adapter for loopback traffic capture

```
> Frame 51: 491 bytes on wire (3928 bits), 491 bytes captured (3928 bits) on interface \Device\NPF_Loopback, id
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 8822, Dst Port: 65454, Seq: 246, Ack: 1, Len: 447
v Data (447 bytes)
    Data: 2d2d2d2d2d424547494e205055424c4943204b45592d2d2d2d2d0a4d494942487a43426c…
    [Length: 447]
```

```
0010  7f 00 00 01 7f 00 00 01  22 76 ff ae d7 f9 0f 9a   -------- "v------
0020  42 98 23 9c 50 18 27 f9  66 9b 00 00 2d 2d 2d 2d   B·#·P·'· f···----
0030  2d 42 45 47 49 4e 20 50  55 42 4c 49 43 20 4b 45   -BEGIN P UBLIC KE
0040  59 2d 2d 2d 2d 2d 0a 4d  49 49 42 48 7a 43 42 6c   Y-----·M IIBHzCBl
0050  51 59 4a 4b 6f 5a 49 68  76 63 4e 41 51 4d 42 4d   QYJKoZIh vcNAQMBM
0060  49 47 48 41 6f 47 42 41  50 33 56 31 78 67 61 43   IGHAoGBA P3V1xgaC
0070  56 6e 6a 34 78 35 31 4d  34 6d 55 56 74 61 79 49   Vnj4x51M 4mUVtayI
0080  36 65 4c 6a 41 55 64 0a  43 45 56 4e 63 47 7a 57   6eLjAUd· CEVNcGzW
0090  69 61 56 39 70 65 58 6b  45 6f 4e 4f 39 48 2f 6e   iaV9peXk EoNO9H/n
00a0  6e 34 75 42 6e 47 78 47  75 32 47 72 4c 49 30 70   n4uBnGxG u2GrLI0p
00b0  79 34 55 5a 51 77 53 73  6b 7a 2f 45 4b 38 36 2b   y4UZQwSs kz/EK86+
00c0  6b 69 5a 4b 55 6a 71 76  0a 49 6a 4b 69 59 53 34   kiZKUjqv ·IjKiYS4
00d0  72 6f 4c 42 61 44 69 65  75 4f 37 63 76 54 2b 79   roLBaDie uO7cvT+y
00e0  4e 65 50 70 2b 38 48 45  2f 35 32 49 7a 39 68 78   NePp+8HE /52Iz9hx
00f0  39 70 72 6a 70 59 7a 6c  65 77 71 4e 62 6d 71 6c   9prjpYzl ewqNbmql
0100  76 6d 55 65 35 4a 79 37  6c 0a 6f 4e 65 79 57 6c   vmUe5Jy7 l·oNeyWl
```

☑ Show packet bytes

client public key

Wireshark · Packet 53 · Adapter for loopback traffic capture

```
> Frame 53: 495 bytes on wire (3960 bits), 495 bytes captured (3960 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 65454, Dst Port: 8822, Seq: 1, Ack: 693, Len: 451
v Data (451 bytes)
    Data: 2d2d2d2d2d424547494e205055424c4943204b45592d2d2d2d2d0a4d494942494443426c…
    [Length: 451]
```

```
0000  02 00 00 00 45 00 01 eb  6f d9 40 00 80 06 00 00   ····E··· o·@·····
0010  7f 00 00 01 7f 00 00 01  ff ae 22 76 42 98 23 9c   ........ ··"vB·#·
0020  d7 f9 11 59 50 18 27 f6  5c 22 00 00 2d 2d 2d 2d   ···YP·'· \"··----
0030  2d 42 45 47 49 4e 20 50  55 42 4c 49 43 20 4b 45   -BEGIN P UBLIC KE
0040  59 2d 2d 2d 2d 2d 0a 4d  49 49 42 49 44 43 42 6c   Y-----·M IIBIDCBl
0050  51 59 4a 4b 6f 5a 49 68  76 63 4e 41 51 4d 42 4d   QYJKoZIh vcNAQMBM
0060  49 47 48 41 6f 47 42 41  50 33 56 31 78 67 61 43   IGHAoGBA P3V1xgaC
0070  56 6e 6a 34 78 35 31 4d  34 6d 55 56 74 61 79 49   Vnj4x51M 4mUVtayI
0080  36 65 4c 6a 41 55 64 0a  43 45 56 4e 63 47 7a 57   6eLjAUd· CEVNcGzW
0090  69 61 56 39 70 65 58 6b  45 6f 4e 4f 39 48 2f 6e   iaV9peXk EoNO9H/n
00a0  6e 34 75 42 6e 47 78 47  75 32 47 72 4c 49 30 70   n4uBnGxG u2GrLI0p
00b0  79 34 55 5a 51 77 53 73  6b 7a 2f 45 4b 38 36 2b   y4UZQwSs kz/EK86+
00c0  6b 69 5a 4b 55 6a 71 76  0a 49 6a 4b 69 59 53 34   kiZKUjqv ·IjKiYS4
00d0  72 6f 4c 42 61 44 69 65  75 4f 37 63 76 54 2b 79   roLBaDie uO7cvT+y
00e0  4e 65 50 70 2b 38 48 45  2f 35 32 49 7a 39 68 78   NePp+8HE /52Iz9hx
00f0  39 70 72 6a 70 59 7a 6c  65 77 71 4e 62 6d 71 6c   9prjpYzl ewqNbmql
0100  76 6d 55 65 35 4a 79 37  6c 0a 6f 4e 65 79 57 6c   vmUe5Jy7 l·oNeyWl
0110  4f 49 6c 5a 30 58 41 67  45 43 41 34 47 46 41 41   OIlZ0XAg ECA4GFAA
```

☑ Show packet bytes

Encrypted packet

Wireshark · Packet 39 · Adapter for loopback traffic capture

```
> Frame 39: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 65496, Dst Port: 8822, Seq: 448, Ack: 693, Len: 64
v Data (64 bytes)
    Data: 8dcb5240d855888896560fd4eff3def083ebcd60bc5a3fc71a174cac1ed631a9d97bb982…
    [Length: 64]
```

```
0000  02 00 00 00 45 00 00 68  cc 6c 40 00 80 06 00 00   ····E··h ·l@·····
0010  7f 00 00 01 7f 00 00 01  ff d8 22 76 9e c3 f6 61   ........ ··"v···a
0020  5c e4 f4 32 50 18 27 f6  c6 c3 00 00 8d cb 52 40   \··2P·'· ......R@
0030  d8 55 88 88 96 56 0f d4  ef f3 de f0 83 eb cd 60   ·U···V·· .......`
0040  bc 5a 3f c7 1a 17 4c ac  1e d6 31 a9 d9 7b b9 82   ·Z?···L· ··1··{··
0050  e5 37 24 72 69 8b a6 4a  18 ea db 2d c6 23 2f eb   ·7$ri··J ···-·#/·
0060  54 cd 51 12 83 fa b2 66  3b dd 50 d6               T·Q····f ;·P·
```