DFS Agent Setup Guide

Introduction

The DFS Agent is a Python-based client application that connects to the DFS server, receives evidence collection tasks, executes them, and transmits the collected evidence back to the server for analysis. This guide will walk you through the process of setting up and using the agent.

System Requirements

- Python 3.6 or later
- Internet connection to reach the DFS server (dfs.dreamgrc.org)
- Administrative/root privileges (required for memory dumps, disk imaging)
- Sufficient storage space for temporary evidence files

Installation

Step 1: Download the Agent

Download the `agent_production.py` script from the distributed collection page in the DFS platform. You can access this by navigating to:

1. Login to your DFS account
2. Navigate to "Distributed Collection"
3. Click on "Agents"
4. Select "Download Agent"

Alternatively, your administrator may provide you with the agent script directly.

Step 2: Transfer to Target System

Transfer the agent script to the target system where you want to collect evidence. This can be done via:

- Secure file transfer (SCP, SFTP)
- USB drive
- Other secure means approved by your organization

Step 3: Create a Dedicated Directory

Create a dedicated directory for the agent on the target system:

```
mkdir -p dfs-agent
cd dfs-agent
cp /path/to/agent_production.py agent.py
```

Agent Registration

Before using the agent, you need to register it with the DFS server.

Option 1: Automatic Registration (Recommended)

Run the agent with the `--register-only` flag:

```
python agent.py --register-only --case-id YOUR_CASE_ID
```

Replace `YOUR_CASE_ID` with the case ID you're working on. This will:

1. Register the agent with the server
2. Generate an Agent ID and API Key
3. Save these credentials to `agent_config.json` in the same directory

Option 2: Manual Configuration

If you already have credentials provided by your administrator, you can create a configuration file manually:

1. Create a file named `agent_config.json` with the following content:

```
{
  "server_url": "https://dfs.dreamgrc.org",
  "agent_id": "YOUR_AGENT_ID",
  "api_key": "YOUR_API_KEY",
  "capabilities": [
    "memory_dump",
    "disk_image",
    "network_capture",
    "artifact_collection"
  ]
}
```

1. Replace `YOUR_AGENT_ID` and `YOUR_API_KEY` with the credentials provided by your administrator.

Running the Agent

Basic Operation

To start the agent with default settings:

```
python agent.py
```

The agent will: 1. Load its configuration from `agent_config.json` 2. Connect to the server 3. Send a heartbeat every 60 seconds 4. Check for pending tasks 5. Process any assigned tasks 6. Transfer collected evidence to the server

Command-Line Options

The agent supports various command-line arguments for customization:

- `--url URL`: Set the server URL (default: https://dfs.dreamgrc.org)
- `--agent-id ID`: Override the agent ID from config
- `--api-key KEY`: Override the API key from config
- `--case-id ID`: Case ID to associate with agent registration
- `--interval SEC`: Check-in interval in seconds (default: 60)
- `--register-only`: Register the agent and exit
- `--debug`: Enable debug logging

Example:

```
python agent.py --interval 30 --debug
```

Task Types

The agent can perform the following types of evidence collection:

1. **Memory Dumps**: Captures RAM contents for analysis
2. **Disk Imaging**: Creates forensic images of disk drives
3. **Network Capture**: Records network traffic
4. **Artifact Collection**: Gathers specific files or system information

Monitoring

The agent logs its activities to: - The console (standard output) - A log file named `dfs_agent.log` in the agent directory

You can monitor this log to track the agent's operation.

Security Considerations

- The agent runs with the permissions of the user that executes it
- For memory dumps and disk imaging, administrative/root privileges are required

- Evidence is temporarily stored on disk during collection and transfer
- All communication with the server is encrypted using HTTPS
- Credentials are stored in `agent_config.json` - secure this file appropriately

Troubleshooting

If the agent fails to connect or operate properly:

1. **Connection Issues**
2. Check network connectivity to dfs.dreamgrc.org
3. Verify that the server URL is correct
4. Check if a firewall is blocking the connection
5. **Authentication Issues**
6. Verify that the agent credentials in `agent_config.json` are correct
7. Try re-registering the agent if authentication fails
8. **Execution Issues**
9. Examine `dfs_agent.log` for detailed error messages
10. Run with the `--debug` flag for more verbose logging
11. Ensure the agent has sufficient permissions for the requested operations
12. **Evidence Transfer Issues**
13. Check available disk space for temporary evidence storage
14. Verify network bandwidth is sufficient for the size of evidence

Advanced Configuration

For advanced users, the agent configuration can be modified for specific environments:

- Change the supported capabilities by editing the `capabilities` list in `agent_config.json`

- Implement proxy support by setting appropriate environment variables
- Customize evidence collection mechanisms based on system architecture

Technical Support

For further assistance: - Contact your DFS administrator - Email support@dreamgrc.org - Call technical support at +1(443)650-8447