Name: Abdurrahman Qureshi

Roll No: 242466

Practical No: 9

Date Of Performance: 15/09/2025

Aim: To implement and demonstrate keylogger attacks

Lab Outcome: A comprehensive theoretical understanding of what keyloggers are, how they are deployed, the mechanisms they use to capture data, and the best practices for detecting, removing, and preventing them.

Theory:

What is a Keylogger?

A keylogger, or keystroke logger, is a form of surveillance technology used to monitor and record every keystroke typed on a specific computer's keyboard. While there are legitimate uses for keyloggers (e.g., parental control, corporate monitoring), they are most commonly associated with malicious activities. Cybercriminals deploy keyloggers to surreptitiously steal sensitive information such as usernames, passwords, credit card numbers, private messages, and other confidential data.

Types of Keyloggers

As detailed in the research material, keyloggers can be broadly categorized based on their method of operation:

- Software-Based Keyloggers: These are the most common type. They are programs installed on a victim's device that run silently in the

background, capturing keystrokes and often sending the logged data to a remote attacker.

- Hardware-Based Keyloggers: These are physical devices, such as a small USB
- dongle that connects between the keyboard and the computer, which intercept and store keystrokes locally. They require physical access to the machine for both installation and retrieval of data.
- Kernel-Level Keyloggers: These are highly sophisticated and operate at the core (kernel) of the operating system. Their deep integration makes them extremely difficult for standard antivirus programs to detect.
- Remote Access Trojan (RAT) Keyloggers: This type is a component of a larger malware package (a trojan). The keylogging functionality is one of many malicious capabilities, which can also include taking screenshots, accessing the webcam, and stealing files.

Common Installation Vectors

Keyloggers are installed on a system without the user's consent through various deceptive methods:

- Phishing & Malicious Attachments: Victims are tricked into opening an infected email attachment that silently installs the keylogger.
- Drive-By Downloads: Visiting a compromised website can trigger an automatic download and installation by exploiting browser vulnerabilities.
- Cracked or Fake Software: Illegitimate copies of software or "free" utilities are often bundled with malware, including keyloggers.
- Social Engineering: Attackers convince a user to voluntarily install a program that has a hidden keylogging component.

CODE:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Keylogger Demo</title>
        <style>                                                box-sizing: border-
            * {                                box;
                                                               margin: 0;
```

```
            padding: 0;
            font-family: "Segoe
UI", Tahoma, Geneva, Verdana, sans-
serif;
        }
        body {
            background-color:
#333;
            min-height: 100vh;
            display: flex;
            justify-content:
center;
            align-items: center;
            padding: 20px;
            color: #fff;
        }
        .container {
            border-radius: 15px;
            box-shadow: 0 10px
30px rgba(0, 0, 0, 0.2);
            width: 90%;
            max-width: 700px;
            padding: 30px;
            text-align: center;
        }
        h1 {
            margin-bottom: 20px;
            font-size: 2.5rem;
        }
        .description {
            margin-bottom: 25px;
            font-size: 1.1rem;
            line-height: 1.6;
        }
        .warning {
            background-color: red;
            padding: 15px;
            margin-bottom: 25px;
            text-align: left;
            border-radius: 4px;
        }
        .input-area {
            margin-bottom: 25px;
        }
        textarea {
            width: 100%;
            height: 150px;
            padding: 15px;
            border-radius: 8px;
            font-size: 1rem;
            resize: vertical;
            transition: border-
color 0.3s;
        }
        textarea:focus {
            outline: none;
        }
        .controls {
            display: flex;
            justify-content:
center;
            gap: 15px;
            margin-bottom: 25px;
            flex-wrap: wrap;
        }
        button {
            padding: 12px 25px;
            border: none;
            border-radius: 50px;
            font-size: 1rem;
            font-weight: 600;
            cursor: pointer;
            transition: all 0.3s
ease;
        }
        .start-btn {
            background-color:
#2ecc71;
            color: white;
        }
        .start-btn:hover {
            background-color:
#27ae60;
            transform:
translateY(-2px);
        }
        .stop-btn {
            background-color:
#e74c3c;
            color: white;
        }
        .stop-btn:hover {
            background-color:
#c0392b;
            transform:
translateY(-2px);
        }
        .download-btn {
            background-color:
#3498db;
            color: white;
        }
        .download-btn:hover {
            background-color:
#2980b9;
            transform:
translateY(-2px);
        }
        .clear-btn {
            background-color:
#f39c12;
            color: white;
```

```
            }
            .clear-btn:hover {
                background-color:
#d35400;
                transform:
translateY(-2px);
            }
            .status {
                margin-top: 20px;
                padding: 10px;
                border-radius: 8px;
                font-weight: 500;
            }
            .active {
                background-color:
#d4edda;
                color: #155724;
            }
            .inactive {
                background-color:
#f8d7da;
                color: #721c24;
            }
            .log-container {
                margin-top: 30px;
                text-align: left;
            }
            .log-title {
                font-size: 1.2rem;
                margin-bottom: 10px;
            }
            #keyLog {
                border: 1px solid
#ddd;
                border-radius: 8px;
                padding: 15px;
                height: 150px;
                overflow-y: auto;
                font-family:
monospace;
                white-space: pre-wrap;
            }
            footer {
                margin-top: 30px;
                font-size: 0.9rem;
                color: #7f8c8d;
            }
        </style>
    </head>
    <body>
        <div class="container">
            <h1>JavaScript Keylogger Demo</h1>
            <p class="description">
                This is a demonstration of keylogging functionality using
                JavaScript. Type in the text area below and see the logged
                keystrokes.
            </p>

            <div class="warning">
                <strong>Warning:</strong> This is for educational purposes only.
                Always ensure you have proper authorization before implementing
                any form of logging. Unauthorized use may violate privacy laws.
            </div>

            <div class="input-area">
                <textarea
                    id="userInput"
                    placeholder="Start typing here..."
                ></textarea>
            </div>

            <div class="controls">
                <button id="startBtn" class="start-btn">Start Logging</button>
                <button id="stopBtn" class="stop-btn">Stop Logging</button>
                <button id="downloadBtn" class="download-btn">
                    Download Log
                </button>
                <button id="clearBtn" class="clear-btn">Clear Log</button>
            </div>
```

```
            <div id="status" class="status inactive">Logging: Inactive</div>

            <div class="log-container">
                <div class="log-title">Keystroke Log:</div>
                <div id="keyLog"></div>
            </div>

            <footer>
                <p>
                    This keylogger runs entirely in your browser. No data is
                    sent to any server.
                </p>
            </footer>
        </div>

        <script>
            document.addEventListener("DOMContentLoaded", function () {
                const userInput = document.getElementById("userInput");
                const keyLog = document.getElementById("keyLog");
                const startBtn = document.getElementById("startBtn");
                const stopBtn = document.getElementById("stopBtn");
                const downloadBtn = document.getElementById("downloadBtn");
                const clearBtn = document.getElementById("clearBtn");
                const status = document.getElementById("status");

                let isLogging = false;
                let logData = "";
                let logEntries = [];


function handleKeyEvent(event) {
                    if (!isLogging)
return;
                    const key =
event.key;
                    const keyCode =
event.keyCode || event.which;
                    const timestamp =
new Date().toLocaleTimeString();

                    if (
                        [
                            "Shift",
                            "Control",
                            "Alt",
                            "Meta",
                            "CapsLock"
,
                        ].includes(key
)
                    ) {
                        return;
                    }

                    let logEntry = "";
                                        if (key === " ") {
                                            logEntry =
                                        "[Space]";
                                        } else if (key ===
                                        "Enter") {
                                            logEntry =
                                        "[Enter]\n";
                                        } else if (key ===
                                        "Tab") {
                                            logEntry =
                                        "[Tab]";
                                            event.preventD
                                        efault();
                                        } else if (key ===
                                        "Backspace") {
                                            logEntry =
                                        "[Backspace]";
                                        } else if (key ===
                                        "Escape") {
                                            logEntry =
                                        "[Escape]";
                                        } else if (key ===
                                        "ArrowUp") {
                                            logEntry =
                                        "[ArrowUp]";
```

```
                    } else if (key ===
"ArrowDown") {
                        logEntry =
"[ArrowDown]";
                    } else if (key ===
"ArrowLeft") {
                        logEntry =
"[ArrowLeft]";
                    } else if (key ===
"ArrowRight") {
                        logEntry =
"[ArrowRight]";
                    } else if
(key.length > 1) {
                        logEntry =
`[${key}]`;
                    } else {
                        if
(event.shiftKey && key >= "a" && key
<= "z") {
                            logEntry =
key.toUpperCase();
                        } else {
                            logEntry =
key;
                        }
                    }

                    logData +=
logEntry;
                    logEntries.push({
                        key: key,
                        keyCode:
keyCode,
                        timestamp:
timestamp,
                        special:
logEntry !== key,
                    });

                    updateLogDisplay()
;
                }

                function updateLogDisplay() {
                    keyLog.textContent = logData;
                    keyLog.scrollTop = keyLog.scrollHeight;
                }

                startBtn.addEventListener("click", function () {
                    isLogging = true;
                    status.textContent = "Logging: Active";
                    status.className = "status active";
                    userInput.focus();
                });

                stopBtn.addEventListener("click", function () {
                    isLogging = false;
                    status.textContent = "Logging: Inactive";
                    status.className = "status inactive";
                });

                downloadBtn.addEventListener("click", function () {
                    if (!logData) {
                        alert("No data to download");
                        return;
                    }

                    const blob = new Blob([logData], { type: "text/plain" });
                    const url = URL.createObjectURL(blob);
                    const a = document.createElement("a");
                    a.href = url;
                    a.download = "keylog.txt";
                    document.body.appendChild(a);
                    a.click();
                    document.body.removeChild(a);
                    URL.revokeObjectURL(url);
```

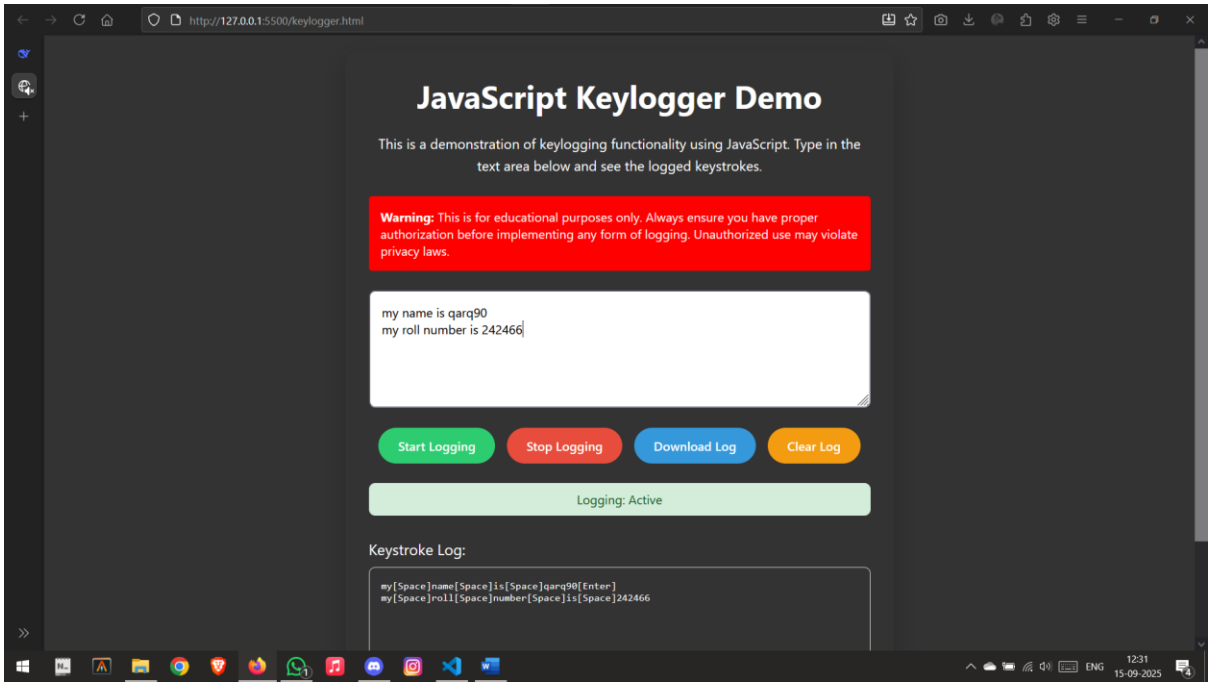
```
            });

            clearBtn.addEventListener("click", function () {
                logData = "";
                logEntries = [];
                updateLogDisplay();
            });

            userInput.addEventListener("keydown", handleKeyEvent);

            updateLogDisplay();
        });
    </script>
</body>
</html>
```

OUPTUT:



Performance (7M)	Journal (3M)	Lab Ethics (2M)	Attendance (3M)	Total (15M)	Faculty Signature