

index.html

index.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Мастер установки Сервера</title>
  <!-- Подключаем чистый современный стиль -->
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <style>
    body { background: #f4f6f9; min-height: 100vh; display: flex;
align-items: center; justify-content: center; }
    .wizard-card { background: white; border-radius: 12px; box-
shadow: 0 8px 24px rgba(0,0,0,0.05); width: 650px; min-height: 450px;
display: flex; flex-direction: column; justify-content: space-between;
padding: 40px; }
    .step { display: none; }
    .step.active { display: block; }
    .step-header { border-bottom: 2px solid #efefef; padding-
bottom: 15px; margin-bottom: 25px; }
  </style>
</head>
<body>

<div class="wizard-card">
  <div class="wizard-content">

    <!-- 1.2.1 Окно приветствия -->
    <div id="step-1" class="step active">
      <div class="step-header"><h4>Шаг 1: Приветствие</h4></div>
      <p class="lead">Добро пожаловать в веб-инсталлятор вашего нового
сервера!</p>
      <p>Этот мастер поможет развернуть и настроить операционную систему
без подключения монитора к серверу. Все действия выполняются удаленно через этот
браузер.</p>
    </div>

    <!-- 1.2.2 Выбор языка -->
    <div id="step-2" class="step">
      <div class="step-header"><h4>Шаг 2: Язык системы</h4></div>
      <label class="form-label">Выберите основной язык для
устанавливаемой ОС:</label>
      <select id="sys_lang" class="form-select form-select-lg">
        <option value="ru_RU.UTF-8">Русский (ru_RU)</option>
        <option value="en_US.UTF-8">English (en_US)</option>
      </select>
    </div>

  </div>

</div>
```

```
<!-- 1.2.3 Выбор раскладки клавиатуры -->
<div id="step-3" class="step">
  <div class="step-header"><h4>Шаг 3: Раскладка
клавиатуры</h4></div>
  <label class="form-label">Выберите раскладку клавиатуры по
умолчанию:</label>
  <select id="sys_layout" class="form-select form-select-lg">
    <option value="ru">Русская (ru)</option>
    <option value="us">Английская (us)</option>
  </select>
</div>

<!-- 1.2.4 Выбор часового пояса -->
<div id="step-4" class="step">
  <div class="step-header"><h4>Шаг 4: Часовой пояс</h4></div>
  <label class="form-label">Укажите ваш часовой пояс:</label>
  <select id="sys_timezone" class="form-select form-select-
lg">
    <option value="Europe/Moscow">Москва (GMT+3)</option>
    <option value="Europe/Kaliningrad">Калининград
(GMT+2)</option>
    <option value="Asia/Yekaterinburg">Екатеринбург
(GMT+5)</option>
    <option value="UTC">UTC (Универсальное время)</option>
  </select>
</div>

<!-- 1.2.5 Имя компьютера -->
<div id="step-5" class="step">
  <div class="step-header"><h4>Шаг 5: Сетевое имя
(Hostname)</h4></div>
  <label class="form-label">Введите имя вашего будущего
сервера:</label>
  <input type="text" id="sys_hostname" class="form-control
form-control-lg" value="my-cool-server"
oninput="validateHostname(this)">
  <div class="form-text text-danger">Разрешена только латиница
(a-z), цифры и дефис.</div>
</div>

<!-- 1.2.6 Разбивка диска и RAID -->
<div id="step-6" class="step">
  <div class="step-header"><h4>Шаг 6: Разметка
накопителей</h4></div>
  <label class="form-label">Выберите конфигурацию дисковой
подсистемы:</label>
  <div class="form-check mb-3">
    <input class="form-check-input" type="radio"
name="disk_mode" id="disk_default" value="default" checked>
```

```

        <label class="form-check-label" for="disk_default">
            <strong>Режим по умолчанию:</strong> Система + 25GB
под установку, остальное в резерв под домашнюю папку пользователя
        </label>
    </div>
    <div class="form-check mb-4">
        <input class="form-check-input" type="radio"
name="disk_mode" id="disk_raid" value="raid1">
        <label class="form-check-label" for="disk_raid">
            <strong>Программный RAID-1 (Зеркало):</strong>
Объединить два диска для отказоустойчивости
        </label>
    </div>

    <!-- ВОТ ЭТОТ БЛОК КРИТИЧЕСКИ НЕОБХОДИМ ДЛЯ ИСПРАВЛЕНИЯ
ОШИБКИ 444 -->
    <div id="disks_selection_zone" class="p-3 bg-light border
rounded mb-4" style="display: none;">
        <h6>Доступные физические диски для построения RAID-1
массивов:</h6>
        <div id="disks_list" class="mt-2">
            <div class="text-muted small">Опрос накопителей
контроллером...</div>
        </div>
        <div class="form-text text-muted mt-2">Для активации
зеркалирования (RAID-1) выберите ровно два одинаковых накопителя.</div>
    </div>

    <div class="alert alert-danger mt-4 d-flex align-items-
center">
        <div class="me-3" style="font-size: 24px;">⚠</div>
        <div>
            <strong>ВНИМАНИЕ:</strong> Все существующие данные
(включая Windows или старые ОС) на выбранных накопителях будут
<strong>полностью и безвозвратно удалены</strong>. Разделы будут
переформатированы!
        </div>
    </div>
    <div class="form-check mb-3">
        <input class="form-check-input border-danger"
type="checkbox" id="confirm_erase" onchange="toggleNextButton()">
        <label class="form-check-label text-danger fw-bold"
for="confirm_erase">
            Я понимаю риски. Подтверждаю полное уничтожение данных и
форматирование дисков.
        </label>
    </div>
</div>

```

<!-- 1.2.7 Создание пользователя -->

```

<div id="step-7" class="step">
  <div class="step-header"><h4>Шаг 7: Учетная запись
администратора</h4></div>
  <div class="mb-3">
    <label class="form-label">Имя пользователя (Логин):</label>
    <input type="text" id="username" class="form-control"
value="eva" oninput="validateUsername(this)" placeholder="только
маленькие латинские буквы">
    <div class="form-text text-muted">Используйте
исключительно маленькие английские буквы (a-z).</div>
  </div>
  <div class="row mb-3">
    <div class="col">
      <label class="form-label">Пароль:</label>
      <input type="password" id="password" class="form-
control" oninput="this.value = this.value.replace(/[a-яА-ЯёЁ]/g, '');
checkPasswordStrength(this.value)">
      <div class="form-text text-danger fw-bold"
style="font-size: 13px;">⚠ Ввод строго на АНГЛИЙСКОЙ раскладке! Русский язык
запрещен.</div>
      <div class="progress mt-2" style="height: 6px;">
        <div id="pass_progress" class="progress-bar bg-
danger" style="width: 0%"></div>
      </div>
      <div id="pass_status" class="small text-muted
mt-1">Сложность: слишком слабый</div>
    </div>
    <div class="col">
      <label class="form-label">Подтверждение
пароля:</label>
      <input type="password" id="password_confirm"
class="form-control" oninput="this.value = this.value.replace(/[a-яА-
ЯёЁ]/g, ''); validatePasswordMatch()">
      <div id="match_status" class="small text-danger
mt-1"></div>
    </div>
  </div>

  <div class="p-3 bg-light border rounded mb-3">
    <h6 class="mb-2" style="font-size: 14px;">Критерии
надежности пароля:</h6>
    <ul class="password-rules-list mb-0">
      <li id="rule_len" class="text-danger">☐ Не менее 8
symbols</li>
      <li id="rule_upper" class="text-danger">☐ Минимум
одна заглавная буква (A-Z)</li>
      <li id="rule_number" class="text-danger">☐ Минимум
одна цифра (0-9)</li>
      <li id="rule_symbol" class="text-danger">☐ Минимум
один спецсимвол (@, #, $, !)</li>
    </ul>
  </div>

```

```

    </div>
  </div>

  <!-- НОВЫЙ ШАГ 8: Окно Двухфакторной аутентификации (2FA) -->
  <div id="step-8" class="step">
    <div class="step-header"><h4>Шаг 8: Безопасность аккаунта
(2FA)</h4></div>
    <div class="p-3 bg-light border rounded">
      <h5>Настройка Двухфакторной аутентификации</h5>
      <p class="small text-muted mb-2">Отсканируйте этот QR-код
мобильным приложением перед началом развертывания системы:</p>

      <div class="mb-3 small text-secondary">
        <strong>Поддерживаемые приложения на
телефоне:</strong>
        <ul class="mb-2 mt-1 ps-3">
          <li>Google Authenticator (iOS / Android)</li>
          <li>Yandex Key / Яндекс Ключ</li>
          <li>Microsoft Authenticator</li>
          <li>2FAS / Aegis / Икс-Ключ</li>
        </ul>
        <span class="text-warning fw-bold">⚠
Внимание:</span> Сохраните этот аккаунт в приложении на телефоне. Код
потребуется при каждом входе на сервер.
      </div>

      <div id="qrcode_container" class="text-center py-2">
        
        <!-- Временный вывод для тестирования шага 8 -->
        <div id="qr-debug-zone" class="mt-3 p-3 bg-light border
rounded small text-muted text-start" style="font-family: monospace;">
          <strong>Отладка перед отправкой в API:</strong><br>
          Логин пользователя: <span id="debug-qr-user" class="fw-bold
text-primary">ожидание...</span><br>
          Имя сервера (Host): <span id="debug-qr-host" class="fw-bold
text-primary">ожидание...</span>
        </div>
        <!-- конец Временный вывод для тестирования шага 8 -->

      </div>
    </div>
  </div>
</div>

```

```
<!-- Теперь это ШАГ 9: Процесс установки -->
<div id="step-9" class="step">
  <div class="step-header"><h4>Шаг 9: Выполнение
установки</h4></div>
  <p id="install_status" class="fw-bold text-
primary">Инициализация скриптов разметки дисковых массивов...</p>
  <div class="progress mb-3" style="height: 25px;">
    <div id="install_progress" class="progress-bar
progress-bar-striped progress-bar-animated" role="progressbar"
style="width: 0%;">0%</div>
  </div>
  <div class="text-muted small">Время выполнения операции: <span
id="install_timer" class="fw-bold text-dark">0 сек.</span></div>
</div>

<!-- Теперь это ШАГ 10: Завершение и перезагрузка -->
<div id="step-10" class="step">
  <div class="step-header"><h4 class="text-success">Установка
успешно завершена!</h4></div>
  <p>Операционная система развернута. Сетевые сервисы, SSH-доступ и
конфигурация RAID подготовлены.</p>
  <div class="alert alert-warning text-center p-4">
    <h5>Внимание! Сервер будет отправлен в перезагрузку через:</h5>
    <h1 id="reboot_timer" class="display-3 font-monospace
my-3 text-danger fw-bold">5</h1>
    <p class="mb-0 small fw-bold">Извлеките загрузочную USB-
флешку из разъема, чтобы сервер загрузился с основного диска.</p>
  </div>
</div>

</div>

<!-- Кнопки управления шагами -->
<div class="buttons d-flex justify-content-between mt-4 border-top
pt-3">
  <button id="btn-prev" class="btn btn-outline-secondary px-4"
onclick="changeStep(-1)" disabled>Назад</button>
  <button id="btn-next" class="btn btn-primary px-4"
onclick="changeStep(1)">Далее</button>
</div>
</div>

<script>
let currentStep = 1;
const totalSteps = 10;

// Флаги валидации для учетной записи
let isPasswordValid = false;
let isPasswordMatching = false;
```

```
// Контроль блокировки кнопки "Далее" на Шаг 6 (Диски)
function toggleNextButton() {
    if (currentStep === 6) {
        const confirmed =
document.getElementById('confirm_erase').checked;
        const diskMode =
document.querySelector('input[name="disk_mode"]:checked').value;
        let isDiskSelectionOk = true;

        if (diskMode === 'raid1') {
            const selectedDisksCount =
document.querySelectorAll('.disk-checkbox:checked').length;
            isDiskSelectionOk = (selectedDisksCount === 2); // Строго 2
диска для RAID-1
        }
        document.getElementById('btn-next').disabled = !(confirmed &&
isDiskSelectionOk);
    }
}

function changeStep(direction) {
    // Скрываем блок ошибок перед проверкой
    const errorBlock = document.getElementById('error-message-zone');
    if (errorBlock) errorBlock.style.display = 'none';

    // Подгрузка доступных дисков при переходе с 5 на 6 шаг
    if (currentStep === 5 && direction === 1) {
        loadSystemDisks();
    }

    // ТРИГГЕР: Переход с 7 на 8 шаг (Безопасное обновление QR)
    if (direction === 1 && currentStep === 7) {
        if (!isPasswordValid || !isPasswordMatching) {
            showInlineError('Пароль не соответствует требованиям или не
совпадает!');
            return;
        }
    }

    // Забираем данные и обновляем QR-код
    try {
        const currentHost =
document.getElementById('sys_hostname').value.trim();
        const currentUser =
document.getElementById('username').value.trim();

        const dbgUser = document.getElementById('debug-qr-user');
        const dbgHost = document.getElementById('debug-qr-host');
        if (dbgUser) dbgUser.innerText = currentUser;
        if (dbgHost) dbgHost.innerText = currentHost;
    }
}
```

```
// ИСПРАВЛЕНО: Точный поиск картинки по ID
    const qrImg = document.getElementById('qr-image');
    if (qrImg) {
        qrImg.src =
`api/2fa.php?username=${encodeURIComponent(currentUser)}&hostname=${encodeURIComponent(currentHost)}`;
    }
} catch (e) {
    console.log("Ошибка обновления элементов Шага 8: ", e);
}

}

// Действия при нажатии "Далее" на Шаге 8 (Старт установки)
if (direction === 1 && currentStep === 8) {
    let selectedDisks = [];
    const diskMode =
document.querySelector('input[name="disk_mode"]:checked').value;

    if (diskMode === 'raid1') {
        document.querySelectorAll('.disk-
checkbox:checked').forEach(cb => {
            selectedDisks.push(cb.value);
        });
    } else {
        const firstDiskInput = document.querySelector('.disk-
checkbox');
        selectedDisks.push(firstDiskInput ? firstDiskInput.value :
'sda');
    }

    // Сбор Payload напрямую из полей (СВЕЖИЕ ДАННЫЕ)
    const payload = {
        lang: document.getElementById('sys_lang').value,
        layout: document.getElementById('sys_layout').value,
        timezone: document.getElementById('sys_timezone').value,
        hostname:
document.getElementById('sys_hostname').value.trim(),
        disk_mode: diskMode,
        disks: selectedDisks,
        username: document.getElementById('username').value.trim(),
        password: document.getElementById('password').value.trim()
    };

    // ПРИНУДИТЕЛЬНЫЙ ЗАПУСК ОТПРАВКИ НА СЕРВЕР
    fetch('api/start_install.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(payload)
    })
    .then(res => res.json())
    .then(result => {
```

```
        if (result && result.success) {
document.getElementById('step-8').classList.remove('active');
        currentStep = 9; // <--- Корректный переход
document.getElementById('step-9').classList.add('active');

        document.getElementById('btn-prev').style.display =
'none';
        document.getElementById('btn-next').style.display =
'none';

        startInstallationSimulation();
    } else {
        showInlineError('Ошибка сервера: ' + (result &&
result.message ? result.message : 'Неизвестный ответ'));
    }
})
    .catch(err => {
        showInlineError('Ошибка сети: сервер установки недоступен.');
```

```
    });
    return; // Ждем ответа fetch, блокируем линейный переход
}

// Линейный переход для остальных шагов
document.getElementById(`step-
${currentStep}`).classList.remove('active');
currentStep += direction;
document.getElementById(`step-
${currentStep}`).classList.add('active');

document.getElementById('btn-prev').disabled = (currentStep === 1
|| currentStep >= 9);
document.getElementById('btn-next').disabled = (currentStep >= 9);

if (currentStep === 6) {
    toggleNextButton();
}
}

// Валидаторы ввода с ограничением до 16 символов
function validateHostname(input) {
    input.value = input.value.toLowerCase().replace(/[^a-z0-9-]/g,
'').substring(0, 16);
}

// Изменено: разрешаем латиницу и цифры для соответствия бэкенду
function validateUsername(input) {
    input.value = input.value.toLowerCase().replace(/[^a-z0-9]/g,
```

```
'').substring(0, 16);
}

// Запрет пробелов и ограничение длины в самом поле пароля
const passwordInputField = document.getElementById('password');
if (passwordInputField) {
    passwordInputField.addEventListener('input', function() {
        this.value = this.value.replace(/\s+/g, '').substring(0, 16);
    });
    // Беззвучный запрет вставки
    ['paste', 'drop'].forEach(e => {
        passwordInputField.addEventListener(e, (ev) =>
        ev.preventDefault());
    });
}

// Проверка силы пароля
function checkPasswordStrength(password) {
    const rules = {
        len: password.length >= 8,
        upper: /[A-Z]/.test(password),
        number: /[0-9]/.test(password),
        symbol: /[!@#$%^&*?&#]/.test(password)
    };

    updateRuleVisual('rule_len', rules.len, "Не менее 8 символов");
    updateRuleVisual('rule_upper', rules.upper, "Минимум одна заглавная
    буква (A-Z)");
    updateRuleVisual('rule_number', rules.number, "Минимум одна цифра
    (0-9)");
    updateRuleVisual('rule_symbol', rules.symbol, "Минимум один
    спецсимвол (@, #, $, !)");

    let score = Object.values(rules).filter(Boolean).length;
    const progress = document.getElementById('pass_progress');
    const status = document.getElementById('pass_status');

    if (progress) {
        progress.className = "progress-bar ";
        if (score <= 1) {
            progress.style.width = "25%"; progress.classList.add("bg-
            danger");
            if (status) status.innerText = "Сложность: слишком слабый";
            isPasswordValid = false;
        } else if (score === 2) {
            progress.style.width = "50%"; progress.classList.add("bg-
            warning");
            if (status) status.innerText = "Сложность: средний";
            isPasswordValid = false;
        } else if (score === 3) {
            progress.style.width = "75%"; progress.classList.add("bg-
```

```
info");
    if (status) status.innerText = "Сложность: хороший";
isPasswordValid = false;
    } else if (score === 4) {
        progress.style.width = "100%"; progress.classList.add("bg-
success");
        if (status) status.innerText = "Сложность: отличный
(безопасный)"; isPasswordValid = true;
    }
}
validatePasswordMatch();
}

function updateRuleVisual(elementId, isValid, text) {
    const el = document.getElementById(elementId);
    if (el) {
        el.innerText = (isValid ? "☑ " : "☐ ") + text;
        el.className = isValid ? "text-success fw-bold" : "text-
danger";
    }
}

function validatePasswordMatch() {
    const passInput = document.getElementById('password');
    const confirmInput = document.getElementById('password_confirm');
    const matchStatus = document.getElementById('match_status');

    if (!passInput || !confirmInput || !matchStatus) return;

    const pass = passInput.value;
    const confirm = confirmInput.value;

    if (!confirm) { matchStatus.innerText = ""; isPasswordMatching =
false; return; }

    if (pass === confirm) {
        matchStatus.innerText = "☑ Пароли совпадают";
        matchStatus.className = "small text-success fw-bold";
        isPasswordMatching = true;
    } else {
        matchStatus.innerText = "☐ Пароли не совпадают";
        matchStatus.className = "small text-danger";
        isPasswordMatching = false;
    }
}

// Мониторинг лога бэкенда
function startInstallationSimulation() {
    const progressBar = document.getElementById('install_progress');
    const statusText = document.getElementById('install_status');
    const timerText = document.getElementById('install_timer');
```

```
let elapsedSeconds = 0;

const durationTimer = setInterval(() => {
    elapsedSeconds++;
    if (timerText) timerText.innerText = elapsedSeconds + ' сек.';
}, 1000);

const logWatcher = setInterval(() => {
    fetch('api/get_log.php')
    .then(res => res.json())
    .then(data => {
        if (data && data.success) {
            if (progressBar) {
                progressBar.style.width = data.progress + '%';
                progressBar.innerText = data.progress + '%';
            }
            if (statusText) statusText.innerText = data.status;

            // Переход на финал при 90% или завершении скриптов
            if (data.progress >= 90) {
                clearInterval(logWatcher);
                clearInterval(durationTimer);

                document.getElementById('step-9').classList.remove('active');
                currentStep = 10;
                document.getElementById('step-10').classList.add('active');

                // Полностью скрываем кнопки управления на Шаг 10
                document.getElementById('btn-prev').style.display =
'none';
                document.getElementById('btn-next').style.display =
'none';

                startRebootCountdown();
            }
        }
    })
    .catch(err => {
        if (statusText) statusText.innerText = "Потеря связи с демоном
установки...";
    });
}, 1500);
}

// Финальный экран и мягкий таймер без alert
function startRebootCountdown() {
    let timeLeft = 5;
    const countdownText = document.getElementById('reboot_timer');
    const statusText = document.getElementById('install_status_final');

    if (statusText) {
```

```
        statusText.innerHTML = '<div class="alert alert-success text-center fw-bold">Система установлена! После перезагрузки сервера обновите страницу браузера.</div>';
    }

    const countdown = setInterval(() => {
        timeLeft--;
        if (countdownText) countdownText.innerText = timeLeft;

        if (timeLeft <= 0) {
            clearInterval(countdown);
            fetch('api/reboot_server.php');
        }
    }, 1000);
}

// Вспомогательная функция вывода ошибок на экран (Привязка к контейнеру .buttons)
function showInlineError(message) {
    let errorZone = document.getElementById('error-message-zone');
    if (!errorZone) {
        errorZone = document.createElement('div');
        errorZone.id = 'error-message-zone';
        errorZone.className = 'alert alert-danger my-2 text-center';

        const buttonsContainer = document.querySelector('.buttons');
        const prevBtn = document.getElementById('btn-prev');
        if (buttonsContainer && prevBtn) {
            buttonsContainer.insertBefore(errorZone, prevBtn);
        } else {
            const card = document.querySelector('.wizard-card') || document.body;
            card.appendChild(errorZone);
        }
    }
    errorZone.innerText = message;
    errorZone.style.display = 'block';
}

function loadSystemDisks() {
    const disksList = document.getElementById('disks_list');
    if (!disksList) return;
    disksList.innerHTML = '<div class="spinner-border spinner-border-sm text-primary"></div> Опрашиваю дисковую подсистему...';

    fetch('api/get_disks.php')
        .then(response => response.json())
        .then(data => {
            if (data && data.success && data.disks.length > 0) {
                disksList.innerHTML = '';
                data.disks.forEach(disk => {
```

```
const diskDiv = document.createElement('div');
diskDiv.className = 'form-check mb-2';
diskDiv.innerHTML = `
    <input class="form-check-input disk-checkbox"
type="checkbox" value="${disk.name}" id="disk_${disk.name}">
    <label class="form-check-label"
for="disk_${disk.name}">
        <strong>/dev/${disk.name}</strong> — Размер:
<span class="badge bg-secondary">${disk.size}</span>
    </label>
`;
disksList.appendChild(diskDiv);
});
} else {
    disksList.innerHTML = '<div class="text-danger">⚠ Накопители
не найдены!</div>';
}
})
.catch(error => {
    disksList.innerHTML = '<div class="text-danger">⚠ Ошибка бэкенда
API!</div>';
});
}

// Логика переключения режимов разметки дисков
document.addEventListener('change', function(e) {
    if (e.target && e.target.name === 'disk_mode') {
        const zone = document.getElementById('disks_selection_zone');
        if (zone) zone.style.display = (e.target.value === 'raid1') ?
'block' : 'none';
    }
});

document.addEventListener('click', function(e) {
    if (e.target && e.target.classList.contains('disk-checkbox')) {
        toggleNextButton();
    }
});

// Защита от возврата кнопкой Back браузера
history.pushState(null, null, location.href);
window.addEventListener('popstate', function () {
    history.pushState(null, null, location.href);
    showInlineError('Для навигации используйте кнопки мастера установки.');
```

```
});
</script>

</body>
</html>
```

2fa.php

2fa.php

```
<?php
// 1. Полностью отключаем дисковое кэширование библиотеки QR
define('QR_CACHEABLE', false);

ob_start();

// Жестко привязываем абсолютный путь к qrlib.php через __DIR__
$lib_path = __DIR__ . '/qrlib.php';

if (!file_exists($lib_path)) {
    ob_end_clean();
    header('HTTP/1.1 500 Internal Server Error');
    header('Content-Type: application/json');
    echo json_encode(['error' => 'Файл qrlib.php не найден в каталоге
api/']);
    exit;
}

// Подключаем библиотеку
include_once $lib_path;

// 2. ПРИНИМАЕМ ПАРАМЕТРЫ ИЗ ФОРМЫ (Если пусто — ставим заводские значения)
$company = isset($_GET['hostname']) && !empty($_GET['hostname']) ?
trim($_GET['hostname']) : "Arch-Server";
$user     = isset($_GET['username']) && !empty($_GET['username']) ?
trim($_GET['username']) : "eva";
$secret   = 'SECRETKEY1234567'; // Ваш мастер-ключ

// Формируем ссылку по международному стандарту TOTP с динамическими данными
$otpauth_url = "otpauth://totp/" . rawurlencode($company) . ":" .
rawurlencode($user) . "?secret=" . $secret . "&issuer=" .
rawurlencode($company);

// Стираем любые предупреждения или случайные пробелы до этого момента
if (ob_get_length()) ob_end_clean();

// Отдаем чистый заголовок картинки
header('Content-Type: image/png');
header('Cache-Control: no-store, no-cache, must-revalidate, max-
age=0');
header('Pragma: no-cache');

// 3. ГЕНЕРИРУЕМ QR-КОД НАПРЯМУЮ В ПАМЯТЬ (БЕЗ КЭША НА ДИСКЕ)
// Пятый параметр (размер пикселя) = 5, шестой параметр (размер рамки) = 2, седьмой
(кэш) = false
```

```
QRcode::png($otpauth_url, false, QR_ECLEVEL_L, 5, 2, false);
exit;
```

disk_prepare.sh

disk_prepare.sh

```
#!/bin/bash
# Финальный эталонный скрипт разметки (api/disk_prepare.sh)
cd "$(dirname "$0")" CONFIG_FILE="install_config.txt"
LOG_FILE="/tmp/install.log" log_msg() { echo "$1"
>> "$LOG_FILE"; echo "$1"; }
# 1. Сбрасываем лог для get_log.php
echo "[PROGRESS] 0" > "$LOG_FILE" echo "[INFO] Инициализация разметки
ДИСКОВЫХ МАССИВОВ..." >>
"$LOG_FILE" if [ ! -f "$CONFIG_FILE" ]; then
    log_msg "[ERROR] Конфигурация install_config.txt не найдена!" exit 1
fi
# 2. Чтение всех переменных в память и экспорт для дочерних процессов
set -a source "$CONFIG_FILE" set +a
# 3. Безопасное уничтожение файла конфигурации на диске
/usr/bin/shred -u "$CONFIG_FILE"
# 4. Начало разметки дисков
echo "[PROGRESS] 10" > "$LOG_FILE" echo "[INFO] Очистка старых таблиц
разделов..." >> "$LOG_FILE" IFS=' '
read -r -a DISKS_ARRAY <<< "$SELECTED_DISKS" TARGET_DISKS=$(printf "
/dev/%s" "${DISKS_ARRAY[@]}")
# 5. Стираем старые сигнатуры ФС
for disk in "${DISKS_ARRAY[@]"; do /usr/bin/wipefs -a "/dev/$disk" >>
"$LOG_FILE" 2>&1 done echo
"[PROGRESS] 25" > "$LOG_FILE" echo "[INFO] Создание файловой системы
Btrfs..." >> "$LOG_FILE"
# 6. Форматирование в Btrfs (RAID-1 или Single)
if [ "$DISK_MODE" == "raid1" ]; then /usr/bin/mkfs.btrfs -f -d raid1 -m
raid1 $TARGET_DISKS >>
"$LOG_FILE" 2>&1
else /usr/bin/mkfs.btrfs -f $TARGET_DISKS >> "$LOG_FILE" 2>&1 fi echo
"[PROGRESS] 40" > "$LOG_FILE" echo
"[INFO] Нарезка системных подтомов @ и @home..." >> "$LOG_FILE"
# 7. Монтирование и создание подтомов
FIRST_DISK="/dev/${DISKS_ARRAY[0]}" /usr/bin/mount "$FIRST_DISK" /mnt
/usr/bin/btrfs subvolume create
/mnt/@ /usr/bin/btrfs subvolume create /mnt/@home /usr/bin/umount /mnt
# 8. Передача эстафеты скрипту установки ядра Arch Linux
echo "[PROGRESS] 50" > "$LOG_FILE" echo "[INFO] Разметка дисков успешно
завершена. Запуск pacstrap..." >>
```

```
"$LOG_FILE" sudo -E ./system_install.sh >> "$LOG_FILE" 2>&1
```

get_disks.php

get_disks.php

```
<?php
header('Content-Type: application/json');

// Передаем параметры lsblk напрямую в бинарник
$command = 'sudo /usr/bin/lsblk -d -n -o NAME,SIZE,MODEL';
exec($command, $output, $return_var);

$disks = [];

if ($return_var === 0) {
    foreach ($output as $line) {
        // Очищаем лишние пробелы и разносим элементы по переменным одной командой
        $line = preg_replace('/\s+/', ' ', trim($line));
        list($name, $size, $model) = explode(' ', $line, 3) + [null, null, 'Unknown Drive'];

        // Фильтруем виртуальные CD-ROM приводы и LiveCD образы
        if (strpos($name, 'sr') === 0 || strpos($name, 'loop') === 0 || strpos($name, 'airootfs') === 0) {
            continue;
        }

        $disks[] = [
            'name' => $name,
            'size' => $size,
            'model' => $model
        ];
    }
}

echo json_encode(['success' => true, 'disks' => $disks],
JSON_UNESCAPED_UNICODE);
?>
```

get_log.php

get_log.php

```
<?php
header('Content-Type: application/json');

$logFile = 'install_log.txt';

if (file_exists($logFile)) {
    // Считываем строки из файла логов
    $lines = file($logFile);
    if (!empty($lines)) {
        $lastLine = trim(end($lines));

        // Разделяем строку по символу "|" (например: "25% | Сборка массива
        Btrfs...")
        $parts = explode('|', $lastLine, 2);

        if (count($parts) === 2) {
            echo json_encode([
                'success' => true,
                'progress' => (int)trim($parts[0]),
                'status' => trim($parts[1])
            ], JSON_UNESCAPED_UNICODE);
            exit;
        }
    }
}

// Если Bash-скрипт еще не успел создать лог-файл, отдаем стартовый статус
echo json_encode(['success' => true, 'progress' => 0, 'status' =>
'Запуск фонового процесса разметки дисков...']);
?>
```

start_install.php

start_install.php

```
<?php
header('Content-Type: application/json');
ini_set('display_errors', 0);

$inputData = file_get_contents('php://input');
$data = json_decode($inputData, true);
```

```
if (!$data) {
    echo json_encode(['success' => false, 'message' => 'Пустой запрос']);
    exit;
}

// Жесткая очистка строк
$clean_hostname = preg_replace('/[^a-zA-Z0-9_\-\./]/', '',
substr(trim($data['hostname']), 0, 16));
$clean_username = preg_replace('/[^a-zA-Z0-9_\-\./]/', '',
substr(trim($data['username']), 0, 16));
$clean_password = preg_replace('/[^a-zA-Z0-9_\-\./]/', '',
substr(trim($data['password']), 0, 16));

// Пишем конфиг с нуля (всегда новые данные)
$configContent = "SYS_LANG=" . $data['lang'] . "\n";
$configContent .= "SYS_LAYOUT=" . $data['layout'] . "\n";
$configContent .= "SYS_TIMEZONE=" . $data['timezone'] . "\n";
$configContent .= "SYS_HOSTNAME=" . $clean_hostname . "\n";
$configContent .= "DISK_MODE=" . $data['disk_mode'] . "\n";
$configContent .= "SELECTED_DISKS=" . implode(' ', $data['disks']) .
"\n";
$configContent .= "SYS_USER=" . $clean_username . "\n";
$configContent .= "SYS_PASS=" . $clean_password . "\n";

$target_file = __DIR__ . '/install_config.txt';

// Перезаписываем файл актуальными данными
if (file_put_contents($target_file, $configContent) !== false) {
    chmod($target_file, 0600);

    // Передаем управление ОДНОМУ управляющему скрипту
    exec('sudo /srv/http/installer/api/disk_prepare.sh > /dev/null 2>&1
&');

    echo json_encode(['success' => true, 'message' => 'Установка успешно
запущена.']);
} else {
    echo json_encode(['success' => false, 'message' => 'Ошибка записи
конфигурации']);
}
exit;
```

system_install.sh

[system_install.sh](#)

```
#!/bin/bash
# Скорректированный скрипт (api/system_install.sh)
cd "$(dirname "$0")"
LOG_FILE="/tmp/install.log"
log_msg() { echo "$1" >> "$LOG_FILE"; echo "$1"; }

# Монтирование Btrfs
IFS=' ' read -r -a DISKS_ARRAY <<< "$SELECTED_DISKS"
FIRST_DISK="/dev/${DISKS_ARRAY[0]}"
/usr/bin/mount -o noatime,compress=zstd,subvol=@ "$FIRST_DISK" /mnt
/usr/bin/mkdir -p /mnt/home
/usr/bin/mount -o noatime,compress=zstd,subvol=@home "$FIRST_DISK"
/mnt/home

log_msg "[PROGRESS] 60"
# Pacstrap с микрокодами из исходного кода
/usr/bin/pacstrap -K /mnt base linux linux-firmware btrfs-progs amd-
ucode intel-ucode >> "$LOG_FILE" 2>&1

if [ $? -ne 0 ]; then
    log_msg "[ERROR] Ошибка установки"
    exit 1
fi

log_msg "[PROGRESS] 75"
/usr/bin/genfstab -U /mnt >> /mnt/etc/fstab 2>&1
log_msg "[PROGRESS] 85"
log_msg "[INFO] Базовая система готова"
```

validate_input.php

[index.php](#)

```
<?php
// Функция полной очистки строк
function sanitize_and_validate($data, $max_length = 16) {
    // 1. Запрещаем пробелы, табы, переносы строк
    $data = preg_replace('/\s+/', '', $data);

    // 2. Обрезаем строго до 16 символов (п. 7)
    $data = mb_substr($data, 0, $max_length, 'UTF-8');

    // 3. Вырезаем любые HTML-теги, скрипты, конструкции типа <?php (п. 4, 5, 6, 8)
    $data = strip_tags($data);

    // 4. Оставляем ТОЛЬКО буквы (латиница), цифры и знаки
```

подчеркивания/дефиса/точки

// Это полностью уничтожает любые попытки SQL-инъекций (типа `OR 1=1`) и XSS-атак

```
$data = preg_replace('/[^a-zA-Z0-9_\-\.\!]/', '', $data);

return $data;
}
```

// Черные списки запрещенных слов (п. 4.1 и 4.2)

```
$forbidden_usernames = ['admin', 'administrator', 'test', 'root',
'user', 'guest'];
$forbidden_passwords = ['qwerty', 'abcdef', '123456', 'iamnumber1',
'password'];
```

// Пример проверки пришедших POST-данных (встраивается в `api/start_install.php`)

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $input_user = isset($_POST['username']) ? $_POST['username'] : '';
    $input_pass = isset($_POST['password']) ? $_POST['password'] : '';

    // Чистим данные
    $clean_user = sanitize_and_validate($input_user, 16);
    $clean_pass = sanitize_and_validate($input_pass, 16);

    // Проверяем по массивам запретов (вхождение подстроки)
    foreach ($forbidden_usernames as $bad_user) {
        if (strpos(strtolower($clean_user), $bad_user) !== false ||
empty($clean_user)) {
            echo json_encode(['error' => 'Недопустимое имя
пользователя!']);
            exit;
        }
    }

    foreach ($forbidden_passwords as $bad_pass) {
        if (strpos(strtolower($clean_pass), $bad_pass) !== false ||
empty($clean_pass)) {
            echo json_encode(['error' => 'Слишком простой или запрещенный
пароль!']);
            exit;
        }
    }
}
```

From:

<https://wwoss.ru/> - **worldwide open-source software**

Permanent link:

https://wwoss.ru/doku.php?id=tmp_full&rev=1779009488

Last update: **2026/05/17 12:18**

