Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Защита разработчиков от ИИ-скама: наш инженерный контур

Одноразовая песочница, Workspace Trust без автозадач, ASR-профиль и FIDO-ключи

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Тестовое задание от «работодателя» открывается на боевой машине: в ~/.ssh лежат ключи к проду, в .env — строки подключения к БД, в браузере живут сессии CRM и почты. Любой автозапуск в момент открытия папки или установки зависимостей превращает одну рабочую станцию в точку входа во всю инфраструктуру. Мы закрываем это конфигурацией парка, а не инструктажем.

Почему это важно бизнесу

- Ключ разработчика открывает прод: одна станция даёт доступ к серверам, репозиториям и облачным консолям компании.
- Простой после инцидента считается днями: ротация ключей и токенов на 20-50 учёток останавливает релизы и поддержку.
- Утечка .env и cookie задевает персональные данные клиентов — это 152-ФЗ и уведомление регулятора, а не только ИТ.
- Переписка по офферам идёт вне контроля ИТ: в компании до 50 человек нет ни службы безопасности, ни фильтра входящих.
- Без централизованных политик защита держится на дисциплине одного человека и ломается на первом интересном оффере.

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

off

task.allowAutomaticTasks: задачи runOn folderOpen не стартуют и не спрашивают

code.visualstudio.com, Tasks

1.69+

версии VS Code с ADMX-шаблонами; ключ Software\Policies\Microsoft\VSCode

code.visualstudio.com, Enterprise policies

9e6c4e1f…

GUID ASR-правила против кражи учётных данных из LSASS; режима Warn у него нет

learn.microsoft.com, ASR rules reference

Disable

Networking в нашем .wsb-профиле; ProtectedClient Enable, ReadOnly true на маппингах

learn.microsoft.com, Windows Sandbox .wsb

7 дней

min-release-age=7 в .npmrc: ставим только версии, опубликованные неделю назад

docs.npmjs.com, npm CLI v11 config

NRT

Continuous (NRT) для детекта по DeviceFileEvents: одна таблица, без join и union

learn.microsoft.com, Defender XDR detections

РЕАЛИЗАЦИЯ 1 ИЗ 3

Одноразовая песочница под любое внешнее «тестовое задание»

Что настраиваем

Веб-студия 18 PM и продуктовая команда 34 PM: Windows 11 Pro на большинстве машин, часть разработчиков на Ubuntu 24.04

Как мы это делаем

1. Готовим .wsb-профиль: Networking Disable, vGPU Disable, ClipboardRedirection Disable, ProtectedClient Enable, MemoryInMB 4096 при минимуме песочницы 2048 МБ.
2. Для Linux и маков держим шаблон QEMU/KVM с чистым Debian и снапшотом base: клон поднимается за 30-40 секунд, сеть — отдельный VLAN без маршрута в LAN.
3. Репозиторий попадает внутрь только через MappedFolder с ReadOnly true; маппинг отрабатывает до LogonCommand, который запускает скрипт первичного осмотра.
4. Осмотр до открытия в редакторе: git clone --no-checkout, затем git ls-tree -r --name-only HEAD — смотрим .vscode/, скрипты сборки, Makefile, compose-файлы.
5. Сброс: закрытие песочницы уничтожает содержимое, VM откатываем на снапшот; ключи, профили браузера и токены внутрь не пробрасываем никогда.

РЕЗУЛЬТАТ

Открытие чужого проекта перестаёт быть событием безопасности: сработавший автозапуск живёт в среде без сети, без буфера обмена и без ключей и исчезает вместе с окном. Разработчик не ждёт согласований — регламент выполняется за минуту, поэтому его действительно соблюдают.

КЛЮЧЕВОЙ НЮАНС

Песочница работает только тогда, когда она быстрее, чем «открою по-быстрому на рабочей». Мы кладём готовый .wsb и пункт «Открыть в песочнице» в контекстное меню — иначе правило обходят в первый же день.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Workspace Trust и политики VS Code централизованно, а не «попросили»

Что настраиваем

Домен на 30-50 PM, Windows 11 23H2/24H2, стабильная ветка VS Code, плюс несколько Linux-станций вне домена

Как мы это делаем

1. Раскладываем vscode.admx/adml в Central Store SYSVOL; политики читаются из Software\Policies\Microsoft\VSCode, сверка — Developer: Policy Diagnostics.
2. Политикой AllowedExtensions (extensions.allowed) фиксируем белый список, UpdateMode и ExtensionsAutoUpdate — редактор не тянет произвольный код.
3. Настроек Workspace Trust в ADMX нет: доставляем %APPDATA%\Code\User\settings.json логон-скриптом GPO, на Linux — Ansible раз в сутки, идемпотентно.
4. Фиксируем security.workspace.trust.enabled true, startupPrompt once, banner always, untrustedFiles prompt, emptyWindow false, task.allowAutomaticTasks off.
5. Раз в месяц выгружаем фактические settings.json и список доверенных папок: доверие наследуется на подпапки, дерево незаметно разрастается.

РЕЗУЛЬТАТ

Незнакомая папка открывается в Restricted Mode: задачи, отладчик, терминал и недоверенные расширения не работают до явного подтверждения. Класс «код выполнился, потому что я просто открыл проект» закрывается конфигурацией парка, а не памятью разработчика.

КЛЮЧЕВОЙ НЮАНС

ADMX для VS Code покрывает расширения, обновления, телеметрию и чат, но не Workspace Trust и не автозадачи. Кто ждёт, что всё закроет одна групповая политика, получает дыру: эти настройки доставляются файлом.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Зависимости, секреты и ключи: где мы обрываем цепочку

Что настраиваем

Команды 5-15 разработчиков на Node.js и Python, репозитории в GitHub, станции под Defender for Endpoint

Как мы это делаем

1. В .npmrc проекта и пользователя: ignore-scripts=true, audit-level=high, min-release-age=7; установка только npm ci и только внутри контейнера сборки.
2. После установки прогоняем npm audit signatures — проверка подписей реестра и provenance-аттестаций; расхождение считаем стоп-фактором для сборки.
3. Python ставим только в hash-checking mode: --require-hashes с жёстким пиннингом версий и --only-binary :all:, чтобы не исполнялся код сборки sdist.
4. ASR-профиль: 5beb7efe (обфусцированные скрипты), d3e037e1 (JS/VBScript → загруженный exe), 01443614 (prevalence/age) — две недели Audit, затем Block.
5. SSH переводим на ed25519-sk с -O resident -O verify-required, на серверах PubkeyAuthOptions verify-required: копия файла ключа бесполезна без токена.

РЕЗУЛЬТАТ

Недобросовестный пакет не получает ни автозапуска при установке, ни свежей публикации без карантина, ни пригодного к переносу приватного ключа. Компрометация сводится к одной изолированной среде, а восстановление — к отзыву короткоживущих токенов вместо аврала на всём парке.

КЛЮЧЕВОЙ НЮАНС

ignore-scripts убирает lifecycle-скрипты, но не выполнение кода при импорте модуля или сборке из исходников. Поэтому изоляция среды остаётся первым рубежом, а контроль зависимостей — вторым, и никогда наоборот.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Ставка только на ADMX для VS Code**
  В шаблоне нет политик для security.workspace.trust.* и task.allowAutomaticTasks: эти значения доставляются файлом settings.json.

- ✕ **startupPrompt по умолчанию never**
  Папка тихо открывается в Restricted Mode, только с баннером. Ставим startupPrompt once и banner always, иначе статус доверия не замечают.

- ✕ **Доверие наследуется на подпапки**
  Доверив каталог проектов целиком, вы доверяете всему, что туда склонируют. Папку под внешние задания держим вне доверенного дерева.

- ✕ **ASR на LSASS: шум и отсутствие Warn**
  Правило не поддерживает Warn, даёт много аудит-событий и не требуется там, где уже включены LSA protection и Credential Guard.

- ✕ **Песочница с сетью по умолчанию**
  По умолчанию Networking включён: гость подключается через виртуальный коммутатор хоста. Для чужого кода ставим Networking Disable.

- ✕ **Mapped folder примонтирован на запись**
  Изменения в папке с правом записи остаются на хосте после уничтожения песочницы. Для внешних репозиториев только ReadOnly true.

- ✕ **Resident-ключ без verify-required**
  Без него подпись требует касания, но не PIN, а сервер не проверит условие, пока в sshd_config нет PubkeyAuthOptions verify-required.

- ✕ **Токены и push protection по умолчанию**
  Политика организации допускает fine-grained токены до 366 дней, а push protection для приватных репозиториев включают отдельно.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- settings.json с Workspace Trust и task.allowAutomaticTasks off на всех машинах
- Внешние репозитории открываются только в Windows Sandbox или одноразовой VM
- ignore-scripts=true в пользовательском .npmrc, установка строго через npm ci
- MFA и уникальные пароли на GitHub, почте и CRM у всех разработчиков

НОРМАЛЬНО

- ADMX VS Code в Central Store, AllowedExtensions белым списком
- Профиль ASR две недели в Audit, затем перевод согласованных правил в Block
- pip --require-hashes и --only-binary :all: в CI и на рабочих станциях
- FIDO-ключи ed25519-sk для SSH и подписи коммитов у всех, кто ходит в прод

ХОРОШО

- Custom detection в Defender XDR на доступ к каталогам ключей, Continuous (NRT)
- Секреты в хранилище и короткоживущие токены вместо .env на локальном диске
- Внутреннее зеркало npm/PyPI с карантином min-release-age и проверкой подписей
- Ежеквартальный аудит политик и учебный «фейковый оффер» на процесс найма

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Знаете ли вы, на скольких машинах реально применены Workspace Trust и task.allowAutomaticTasks off?
- [ ] Есть ли готовая одноразовая среда под внешний репозиторий и запускается ли она меньше чем за минуту?
- [ ] Отключены ли lifecycle-скрипты npm и запрещены ли сборки из sdist на рабочих станциях разработчиков?
- [ ] Проверяется ли состав репозитория до открытия: .vscode/, скрипты сборки, Makefile, compose-файлы?
- [ ] Развёрнут ли профиль ASR и знаете ли вы, какое правило сейчас в Audit, а какое уже в Block?
- [ ] Лежат ли прод-ключи SSH файлом на диске или их приватная часть не покидает аппаратный токен?
- [ ] Ограничен ли предельный срок жизни токенов и включена ли защита от пуша секретов в организации?
- [ ] Известно ли, за сколько часов вы отзовёте и перевыпустите все ключи и токены после инцидента?
- [ ] Есть ли алерт на чтение каталогов с ключами и облачными профилями и кто реагирует на него ночью?
- [ ] Понимают ли разработчики, что делать с подозрительным заданием, кроме как просто закрыть окно?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Разворачиваем .wsb-профили и шаблон одноразовой VM, выносим запуск в контекстное меню — за один выезд.
- Готовим ADMX для VS Code, доставку settings.json через GPO и Ansible-роль для Linux-станций.
- Собираем профиль ASR в Intune или GPO, ведём его от Audit до Block и разбираем ложные срабатывания.
- Переводим SSH, подпись коммитов и вход в репозитории на FIDO-ключи, выносим секреты из .env в хранилище.
- Настраиваем детекты на доступ к ключам и регламент реакции: кто, что и в каком порядке отзывает.

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Workspace Trust и Restricted Mode: раздел Editing / Workspaces (code.visualstudio.com — 2026)

02 Tasks: автозадачи runOn folderOpen и task.allowAutomaticTasks (code.visualstudio.com — 2026)

03 Enterprise policies: ADMX с версии 1.69, Policy Diagnostics (code.visualstudio.com — 1.69+)

04 Attack surface reduction rules reference: GUID, режимы, зависимости (learn.microsoft.com — 2026)

05 Use and configure Windows Sandbox: параметры .wsb-файла (learn.microsoft.com — 2026)

06 Defender XDR custom detection rules: частоты и Continuous (NRT) (learn.microsoft.com — 2026)

07 npm CLI config: ignore-scripts, min-release-age, audit signatures (docs.npmjs.com — v11)

08 Secure installs: hash-checking mode, --only-binary :all: (pip.pypa.io — 26.1)

09 ssh-keygen(1) и sshd_config(5): ed25519-sk, PubkeyAuthOptions (man.openbsd.org — 2026)

10 Наш шаблон dev-baseline: settings.json, .wsb, профиль ASR (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.