Кибератака, которая пишет себя сама

Алексей ИВАНОВ,
заместитель технического директора по вопросам инноваций «АйТи Бастион»

Еще недавно разговор об искусственном интеллекте в арсенале злоумышленников почти всегда приходил к одному выводу: ИИ лишь ускоряет рутину. Нейросеть может написать фишинговое письмо, помочь с текстом легенды или, к примеру, с шаблоном кода. Однако все эти практики – уже прошлое. Развитие механизма ИИ-агентов привело к тому, что нейросети все больше внедряются не в подготовительный этап атаки, а в ее механику. Туда, где раньше все держалось на заранее прописанном сценарии и ручном управлении оператора. Как из помощника хакера нейросеть превращается в непосредственную угрозу?

Под ИИ-агентами понимаются не просто языковые модели, генерирующие текст по запросу, а системы, способные самостоятельно планировать последовательность действий, выполнять их, оценивать результат и корректировать стратегию без участия человека. Это принципиально иной уровень автономности по сравнению с привычным использованием ИИ как инструмента поддержки.

Сегодня наиболее широкое распространение получают сценарии, в которых искусственный интеллект используется прежде всего в качестве усиления стандартных схем мошенничества. В первую очередь, это генерация и подделка аудио- и видеоконтента, применяемого в атаках социальной инженерии – от простых телефонных звонков до полноценной имитации видеокоммуникаций. Технологический уровень таких подделок пока нельзя назвать однородным, однако эффективность подобного рода мошенничества заключается, скорее, не в качестве исполнения, а в снижении порога входа. Инструменты становятся дешевле, проще и доступнее для широкого круга злоумышленников. Однако в подобных сценариях ИИ не меняет архитектуру атаки, оставаясь лишь вспомогательным компонентом.

Разница в механизме работы ИИ-агента и ИИ-помощника принципиальна. Когда модель работает в качестве копирайтера или помогает программисту, итоговый вредоносный инструмент остается вполне привычным. Это бинарный файл, строгий набор функций и понятная логика действий вируса. Однако если модель встроена в цикл исполнения, поведение вредоноса становится словно живым: действия анализируются и выбираются по месту: с учетом окружения, конфигурации системы и результата предыдущих шагов. Фактически, мы наблюдаем переход от ИИ-контента к ИИ-управлению: от заблаговременной генерации до генерации момента и выполнения «прямо сейчас» в контексте конкретной среды.

В классической модели вредоносное ПО представляет собой законченный артефакт: файл с фиксированной логикой, заранее определенным набором функций и предсказуемой последовательностью шагов. Даже при сложной обфускации или использовании загрузчиков поведение такого образца в целом остается детерминированным. В случае внедрения языковой модели непосредственно в цикл исполнения вредонос превращается из статического объекта в изменяемый, адаптивный и контекстно-зависимый процесс. Его поведение формируется не только кодом, но и результатами рассуждений модели в каждый конкретный момент времени.

Один из наиболее показательных случаев такого сдвига – prompt-driven ransomware: класс «умного» вредоносного программного обеспечения для вымогательства. Он примечателен тем, что здесь наблюдается изменение самого способа описания поведения атаки. Часть логики теперь выражается в виде запросов к языковой модели, а не «зашивается» в код. По сути, промпт становится исполняемой спецификацией: что именно искать, что считать ценным, а что игнорировать, как действовать дальше. И определяется это не набором жестких процедур, а текстовой инструкцией и ответом модели в момент запуска. Используя автоматизированное логическое рассуждение, синтез кода

и контекстно-зависимое принятие решений, мы видим новую угрозу, которая эксплуатирует большие языковые модели (LLM) для автономного планирования, адаптации и выполнения полного жизненного цикла атаки вымогательского ПО. Иными словами, Ransomware 3.0.

Пока что Ransomware 3.0 – не устоявшийся академический термин, а лишь предложенное обозначение эволюционного развития концепции программ вымогателей. Если следовать этой условности, можно обозначить и Ransomware 1.0 как простые, автономные программы начала 2000-х и 2010-х гг. В них жестко зашита логика, предполагающая примитивное шифрование или блокировку экрана с отсутствием инфраструктуры управления. В научных статьях нулевых слово ransomware используется без версионирования – деление на версии появилось позже, в обзорных и сравнительных работах. Оно позволяет зафиксировать смену архитектурных парадигм и, как следствие, понять, почему методы защиты, эффективно работавшие против предыдущих поколений вымогателей, оказываются малоприменимыми к новым моделям атак. Речь не о косметических изменениях, а о принципиально разных способах организации вредоносного поведения.

Ransomware 2.0 – это также не термин из одной статьи, а общее обозначение, которое постепенно закрепилось в отчетах и аналитике индустрии ИБ во второй половине 2010-х. Под второй версией понимают переход к операторским кампаниям, Ransomware-as-a-Service, ручному управлению, предварительной разведке, расширению присутствия в атакуемой системе и двойному вымогательству – сочетанию шифрования и угроз публикации похищенной информации. Цифра 2.0 используется задним числом как удобная рамка для отличия этих кампаний от ранних автономных образцов. В отличие от первых двух версий, термин Ransomware 3.0 был впервые введен авторами Корнеллского университета.[1] Нумерация здесь задается для обозначения архитектурной эволюции – новое поколение отличается концептуально, а не только по тактикам. Вымогатель рассматривается как LLM-оркестрируемый процесс (термин, введенный статьей), а не статический бинарный код. Для средств защиты это означает серьезный вызов.

Если в предыдущих поколениях можно было опираться на сигнатуры, характерные участки кода или устойчивые поведенческие цепочки, то в случае LLM-оркестрируемого процесса таких якорей становится значительно меньше. Поведение может варьироваться от запуска к запуску, а этапы атаки – перестраиваться или повторяться в произвольном порядке в зависимости от промежуточных результатов. Мы наблюдаем не новые приемы обфускации или шифрования, а новый архитектурный сдвиг к динамически собираемому процессу. Это значительно сокращает возможности статического анализа и сигнатурного детектирования. Этапы атаки уже не обязательно следуют линейной последовательности. LLM может возвращаться к разведке, менять приоритеты и перестраивать план в процессе взлома.

Доказательство концепции Ransomware 3.0 появилось летом 2025 г. При анализе нетипичного образца вымогательского ПО исследователи ESET обнаружили PromptLock, в котором отсутствовала жестко зашитая логика выполнения атаки. При реверс-инжиниринге выяснилось, что образец в рантайме обращается к локально развернутой языковой модели (gpt-oss:20b) через API Ollama. На основе текстовых запросов модель генерировала Lua-скрипты, которые сразу же исполнялись и использовались для перечисления файлов, их эксфильтрации и шифрования. Таким образом, вредоносный код в привычном понимании выполнял лишь роль минимального загрузчика и интерфейса взаимодействия с моделью, тогда как ключевые решения – какие файлы обрабатывать, какие действия выполнять и в какой последовательности – принимались в процессе работы, на основании ответов LLM. Ключевые элементы поведения формировались динамически, а не были заранее реализованы в коде. ESET классифицировала PromptLock как proof-of-concept, указав на его экспериментальный характер и отсутствие признаков зрелой преступной кампании. При этом исследователи подчеркнули принципиальную новизну подхода: PromptLock стал первым задокументированным примером ransomware, где логика вымогательства генерируется языковой моделью во время выполнения.

Следует подчеркнуть, что подобный подход пока остается скорее разовым исследовательским экспериментом, чем массово применяемой техникой. Тем не менее, сам факт его успешной реализации демонстрирует направление эволюции атакующих инструментов и позволяет заранее оценить потенциальные риски для инфраструктур, в которых языковые модели начинают играть роль полноценного вычислительного компонента.

Формально мы не вступаем в эру новых, практически неотразимых угроз. На текущий момент PromptLock – фактически единственный задокументированный случай реализации концепции Ransomware 3.0. Остальные описанные случаи носят экспериментальный характер либо относятся к вспомогательным инструментам написания вредоносного кода (например MalTerminal, GPROMPTFLUX, PROMPTSTEAL и др.). Слабость концепции PromptLock заключается в том, что он предполагает наличие Ollama – модель уже должна быть развернута локально в атакуемой инфраструктуре. Вредоносное

[1] *Альтернативное предложение для печатной версии: «В отличие от первых двух версий, термин Ransomware 3.0 был впервые введен авторами Корнеллского университета в статье Ransomware 3.0: Self-Composing and LLM-Orchestrated (https://arxiv.org/abs/2508.20444)»*

ПО не загружает модель по сети и не разворачивает ее с нуля. Это попросту нереалистично – загрузка модели требует длительного времени и гигабайтов трафика в придачу.

На практике подобное поведение будет легко детектироваться уже существующими средствами защиты (EDR/XDR, PAM), как и повышение требований к CPU/GPU и памяти. Даже без специализированных механизмов анализа ИИ-компонентов попытки использования локально развернутой языковой модели в контексте вредоносной активности сопровождаются характерными признаками: нетипичными цепочками процессов, выполнением динамически сгенерированных скриптов, а также обращением к чувствительным ресурсам файловой системы. В этом смысле prompt-driven ransomware не выходит за рамки наблюдаемого для классических атак – меняется не столько факт выполнения действий, сколько способ принятия решений.

В случае мошеннических сценариев, основанных на подделке голоса или видео, ключевую роль будут играть решения, ориентированные на контроль доверия и подтверждение подлинности действий. Это и системы многофакторной аутентификации, поведенческого анализа, и механизмы обязательной верификации критически значимых операций, к примеру, денежных переводов внутри организации, вне зависимости от убедительности входящего сигнала. Существенную роль в противодействии подобным сценариям играют не столько техническая защита, сколько организационные меры, инициированные самими компаниями. Сотрудникам необходимо понимать, что убедительная визуальная или голосовая имитация отныне не является надежным признаком подлинности. Речь идет о формировании навыков распознавания неочевидных признаков ИИ – несоответствий в контексте и нетипичных просьб, даже если обращение выглядит как сообщение от руководителя в мессенджере в формате видеосообщения.

В то же время локальные LLM-модели получают широкое распространение, компании видят все больше пользы от их использования в своих процессах, а мы наблюдаем рост количества публикаций о способах развертывания и применения инференсов моделей в закрытых контурах. Часто такие развертывания выполняются без соблюдения даже минимальных требований безопасности, шифрования трафика или установки токенов. Следовательно, попыток эксплуатировать такие уязвимости через концепции prompt-driven ransomware в будущем станет больше.

При этом на уровне публичного регулирования искусственный интеллект сегодня чаще рассматривается только в контексте потенциального вреда в социально значимых областях, таких как здравоохранение, образование или судебная практика. Одна из недавних инициатив – создание рамочного законопроекта по регулированию ИИ в «чувствительных» сферах: обсуждаются маркировка ИИ-контента и определение ответственных за ошибки нейросетей. Однако вопросы использования ИИ в преступных схемах, мошенничестве и кибератаках остаются в значительной степени вне фокуса формализованного регулирования и описываются лишь через уже существующие классы угроз. К примеру, в банке данных угроз безопасности информации ФСТЭК России ИИ фигурирует как фактор, усиливающий известные классы атак, – социальную инженерию, мошенничество и несанкционированный доступ. Акцент по-прежнему остается на последствиях и типах нарушений, тогда как архитектурные изменения, связанные с автономным принятием решений, пока находятся за рамками взора регуляторов.

В этом смысле Ransomware 3.0 стоит рассматривать не как очередную вариацию вредоносного ПО, а как симптом более глубокого сдвига. Языковые модели постепенно перестают быть вспомогательным инструментом и начинают выступать полноценным элементом вычислительной среды, влияющим на логику принятия решений внутри атакующих процессов. Для специалистов по информационной безопасности это означает необходимость пересмотра привычных моделей угроз: объектом анализа становится не только код, но и контекст, в котором этот код взаимодействует с моделью. И хотя сегодня подобные атаки остаются редкими и экспериментальными, их появление задает направление эволюции – от статических артефактов к адаптивным и самонастраивающимся процессам, где граница между программой и ее поведением постепенно размывается. ■