Атака Golden *MSA, бессмысленная и беспощадная

Александр Родченко

Летом этого года исследователи из Semperis Security раскрыли критическую уязвимость в Windows Server 2025, которая позволяет атакующему с правами домен-админа легко подобрать пароли ко всем учетным записям типа dMSA и gMSA в домене. В этой статье мы разберём, как устроена такая атака и как её детектировать.

Что такое *MSA

MSA – *Managed Service Account*. Общее название типов учетных записей. Технология, которая и породила семейство *MSA-аккаунтов, заключается в автоматической ротации пароля сервисной учётки. При этом пароль в открытом виде доступен, что называется, «не только лишь всем». Подтипы – **gMSA** и **dMSA**.

KDS – *Group Key Distribution Service* (компонент AD). Отвечает за генерацию/распределение материалов, из которых DC как-то получает пароли MSA.

gMSA – *Group Managed Service Account*. Классический «управляемый» сервис-аккаунт (Windows Server 2012+). Может использоваться несколькими хостами (группа хостов). Пароль вычисляется DC (что называется, в рантайме) и при необходимости отдаётся уполномоченным субъектам через атрибут **msDS-ManagedPassword** (как BLOB). Идентификатор версии/цикла (пароль же меняется автоматически, вот и нужна сущность для понимания, какой сейчас он конкретно будет) хранится в **msDS-ManagedPasswordId**.

Для журналирования событий, связанных с dMSA, можно использовать:

- весь спектр журналов, связанных со входом учётки,
- Applications and Services Logs → Microsoft → Windows → Security-Netlogon → Operational → 9000/9001/9002 (тут, тут и тут - это ошибки разного рода при установке gMSA):
- Windows Logs → Directory Service → 2946 / 2947 («выдача пароля gMSA успешна» / «попытка получить пароль gMSA неудачна»)

dMSA – *Delegated Managed Service Account* (Windows Server 2025). Эволюция gMSA с привязкой аутентификации к конкретным разрешённым компьютерам (делегированная модель логона). У донских казаков и dMSA много больше общего, чем буква “д” вначале: казаки не выдаются с Дона, как и пароли dMSA – с контроллера домена. Запомните это сравнение.

Технология dMSA работает поверх Kerberos и замещает билеты «уполномоченных» аккаунтов. Когда «старый» (он же уполномоченный) сервис пытается войти, как обычно, под самим собой, то DC возвращает KRB-ERROR с полем KERB-SUPERSEDED-BY-USER, и клиент **повторяет логон уже как dMSA** с чудодейственно (нет) взявшимся у него билетом для dMSA-учётки. Пароль при этом **не покидает контроллер домена.**

Для мониторинга dMSA можно смотреть соответствующие логи, а также обычные события, связанные со входом по Kerberos и системной службы:

- На КД
  - **Security → 4768** (*Kerberos Authentication Service*): выдача **TGT** для gMSA$ учётки
  - **Security → 4769 / 4770** (*Kerberos Service Ticket Operations*): выдача/продление **TGS**
- На хосте-потребителе (где крутится служба)
  - **Security → 4624 (Logon Type = 5)**: локальный вход под gMSA$ при старте службы/задачи. По этому событию удобно биндить время ротации пароля к факту нового логина.

Естественно, сразу возникает идея для детектирующей логики: dMSA может только в Kerberos, если аккаунт dMSA использует NTLM – это очень странно.

Как рождается пароль (оба типа MSA)

На самом деле, такая сущность как пароль есть для обоих типов УЗ. Даже если нам нужен только билет, то сначала на домен-контроллере вычисляется пароль, хешируется и формируется билет. А как создаётся пароль для *MSA-учёток? Ещё раз: он не записан где-то, а создаётся всякий раз, когда его зачем-то надо.

На домен контроллере есть **KDS root key** – это корневой (мастер) ключ KDS для домена/леса. Объекты лежат в Configuration NC: CN=Master Root Keys, CN=Group Key Distribution Service, CN=Services, <Configuration NC>. Обратим внимание на:

- **msKds-ProvRootKey (class)** – это класс объектов KDS-корневых ключей. Это «контейнеры» крипто-материалов KDS.
- **msKds-RootKeyData (attr)** – это главный секрет внутри msKds-ProvRootKey. Бинарные данные корневого ключа KDS (то самое, что атакующим нужно прочитать, но это я забегаю вперёд).
- **msKds-KDFAlgorithmID, msKds-KDFParam, msKds-SecretAgreementParam** – это параметры алгоритма и ввода в функцию генерации и вывода паролей. Это тоже атрибуты.
- И, кстати, **KDF** – собственно Key Derivation Function. Алгоритм, который, используя KDS-материалы (см. выше) + временную функцию + контекст (SID/имена домена и т. д.), детерминировано выводит пароль MSA.

На этом месте у SOC-аналитика должен сработать рефлекс: вижу новые атрибуты и классы в статье – добавляю их в SACL и мониторю событие 4662 (доступ к объекту каталога) + 5136 (изменение объекта каталога). И это правильная привычка.

А теперь посмотрим, как работает функция вывода пароля KDF. Мы сказали, что ей нужны: KDS-материалы + временная функция + контекст пользователя. И понятно, где что брать – кроме таинственной **временной функции**.

Это на самом деле не текущее время, а идентификатор временного отрезка («эпоха»), на котором в последний раз вычислялся пароль (условно, время разбивается на 10-часовые интервалы и вообще-то этот счётчик тикает циклически и всего бывает 1024 интервала). Это лежит в атрибуте ManagedPasswordId для текущей учётки, и он актуализируется по использованию. А если атрибута нет, то алгоритм тоже не растеряется, возьмёт время создания учётки и вычислит для него «эпоху».

Звучит как-то брутабельно, не правда ли? – есть достаточно постоянные ключи и контекст пользователя, а время для генерации пароля тикает +10-часовым интервалом… Именно это и эксплуатируется в данной атаке: можно предсказать практически все пароли.

Кто может воссоздать пароль

Хорошая новость: провести такую атаку может не каждый. Вспомним про **KDS root key** – и заметим, что механизм AdminSDHolder ни к классу, ни к его атрибутам не относятся. Это отдельные объекты конфигурации с собственным «жёстким» DACL. По умолчанию только **Enterprise Admins/Domain Admins/SYSTEM** способны читать и объект, и значение **msKds-RootKeyData**, а обычные пользователи/компьютеры – совсем нет.

Атрибут ещё и «конфиденциальный»: **msKds-RootKeyData** помечен Search-Flags = 0x00000280 — это:

- **0x80** – атрибут **Confidential** (требует *Read Property* **и** *Control Access* на сам атрибут/сет свойств для чтения значения);
- **0x200** – включён в **Filtered Attribute Set** (не реплицируется на RODC).

Иными словами, даже если у субъекта есть обычное чтение объекта, то для чтения значения msKds-RootKeyData нужны расширенные права именно на этот атрибут. Поэтому, чтобы провернуть атаку Golden @MSA, нужно быть домен-админом или системой на КД (не на RODC). Ну или если кто-то нарочно не переписал права.

Резюме атаки

Golden *MSA – это детерминированное восстановление пароля MSA. По сути — перебор малых временных индексов эпохи ManagedPasswordId (всего 1024 комбинации) при наличии KDS root key. Это больше оффлайн-воссоздание того, что делает DC, чем классическая удалённая эксплуатация. Работает и против gMSA, и против dMSA (Server 2025).

Вопрос	**Суть**
Что нужно атакующему?	Доступ к **KDS root key** (объект класса msKds-ProvRootKey в *Configuration NC*). Без него пароль не восстановить.
Где «мастер-секрет»?	Атрибут **msKds-RootKeyData** у msKds-ProvRootKey.
В чём «перебор»?	В ManagedPasswordId два временных индекса имеют по 32 значения (**~1024** вариантов), поэтому нужную эпоху легко подобрать. (Semperis)
Что дальше?	По KDS-материалам и эпохе DC (или инструмент злоумышленника) детерминированно выводит пароль: для gMSA он обычно выдаётся как BLOB msDS-ManagedPassword, для dMSA – используется KDC для билетов.
Кто вообще «видит» пароль легитимно?	У **gMSA** – авторизованные субъекты через msDS-ManagedPassword (Directory Service события 2946/2947 фиксируют выдачу/ошибку). У **dMSA** пароль наружу не отдаётся – клиент живёт на Kerberos-билетах.
Требуется ли изначальная привилегия?	Да: чтение msKds-RootKeyData по умолчанию доступно только EA/DA/SYSTEM на DC. Это **пост-DA** техника.

Советы по детектированию

- Вешайте **SACL** на контейнер «CN=Master Root Keys,*» и фиксируйте **Security 4662** чтение msKds-RootKeyData (любой не-DC субъект = алерт). Ограничьте/проверьте DACL до EA/DA/SYSTEM.
- Следите за Directory Service **2946/2947** на DC (успех/неудача получения msDS-ManagedPassword). Аномальные частоты/источники – алерт + расследование (стоит заранее определить, какие системы **должны** запрашивать пароли gMSA, а какие нет).
- Коррелируйте **4768/4769** (Kerberos) по MSA-учеткам и **4624 (Type 5)** на хостах-потребителях (нестандартные хосты).