Web Application Vulnerability Testing with Nessus

Rik A. Jones, CISSP

rikjones@computer.org
Rik A. Jones

- Web developer since 1995 (16+ years)
- Involved with information security since 2006 (5+ years)
- Senior Information Security Analysts for Dallas County Community College District
- CISSP and GIAC certified
- Member of the Dallas OWASP Leadership Team
- Member of the Dallas Chapter of InfraGard
This is not a sales presentation

I am not affiliated with Tenable or Nessus other than being a knowledgeable and frequent user.

I am here to show you how to use Nessus as a tool, one of many tools I keep in my toolbox.
Introduction to Nessus

Nessus is a multiple platform network and host vulnerability scanner

Server Supported on:
- Window
- Linux
- Mac OS
- UNIX

Clients: Web based and Mobile (IOS, Android)
Introduction to Nessus

Nessus has 2 licensing models (plugin feeds)

- **ProfessionalFeed**
  - Commercial use
  - Access to support portal

- **HomeFeed**
  - No charge
  - Personal use only
  - Some limits to functionality
    - Only 16 IP addresses
    - No compliance/audit checks
    - No scan scheduling
Introduction to Nessus

Nessus Terminology

• **Policy** – Configuration settings for conducting a scan
• **Scan** – Associates a list of IPs and/or domain names with a policy
  • Basic Scan (Run Now)
  • Template
  • Scheduled Template (ProfessionalFeed Only)
    • One time or repeating
• **Report** – The result of a specific instance of a scan
• **Plugin** – A security check, or a scan settings window
• **Plugin Family** – A group of plugins with something in common (e.g. FTP, Web Servers, Cisco)
Introduction to Nessus

Nessus Customization Options

- **Reports Templates** - Coded in XSLT
- **Plugins** - Coded in NASL (Nessus Attack Scripting Language)
- **Audit Files** - Coded in Pseudo-XML [ProfessionalFeed Only]
- **Import/Export** - Nessus & Nessus 2 format coded in XML. Same format for reports and profiles
Logging in to Nessus

By default Nessus runs on port 8834 and can be accessed with any Flash enabled Web Browser.
Basic Navigation

There are four navigation tabs at the top:

- Reports
- Scans
- Policies
- Users
Reports Tab

The Reports tab list the results of scans you have conducted, are currently running or have imported.
Scans Tab

The Scans tab list currently running scans, scan templates and scheduled scans.
Policies Tab

The Policies tab list the scan configurations available for scans
Users Tab

The Users tab list users and allows the addition, deletion or editing of users accounts
Creating a Basic Web Application Scan Policy

- The goal is to create a generic policy for scanning unknown Web applications.
- We will set basic settings that work for most Web Applications.
- When we create an Advanced Web application policy we will add additional settings for a specific Web Application.
Creating a Basic Web Application Scan Policy

Step 1: Go to the Policies Tab and select the default “Web App Test” policy
Creating a Basic Web Application Scan Policy

Step 2: Click on the “Copy” button. This will create a new Policy called “Copy of Web App Test”
Creating a Basic Web Application Scan Policy

Step 3: Select the new policy “Copy of Web App Test”
Creating a Basic Web Application Scan Policy

Step 4: Click on the Edit Button
Creating a Basic Web Application Scan Policy

This will open the Edit Policy screen
Creating a Basic Web Application Scan Policy

Step 5: Change the policy name

![Policy Editing Interface]

- **Name**: My Basic Web App Tests
- **Visibility**: Shared
- **Description**: Empty
Creating a Basic Web Application Scan Policy

Step 6: Uncheck all port scanners except for “TCP Scan” and “Ping Host”
Creating a Basic Web Application Scan Policy

Step 7: Set the Port Scan Range

• default = all common ports listed in the “nessus-services” configuration file
• all = every port (1 - 65,535)
• Specific list (e.g. 80, 443, 8080, 8009)
Creating a Basic Web Application Scan Policy

Step 8: Click on the “Plugins” Side Tab
Creating a Basic Web Application Scan Policy

This should take you to the Plugins selection
Creating a Basic Web Application Scan Policy

Step 9: Click on “Disable All” to disable all plugin families
Creating a Basic Web Application Scan Policy

Step 10: Enable the following plugin families by clicking on the grey dot next to the family name:

- Backdoors
- CGI Abuses
- CGI Abuses : XSS
- Cisco
- Databases
- FTP
- Firewalls
- Gain a shell remotely
- General

- Misc.
- Netware
- Peer-To-Pear File Sharing
- SMTP problems
- Service detection
- Settings
- Web Servers
- Windows
- Windows: Microsoft Bulletins
Creating a Basic Web Application Scan Policy

Step 11: Click on the “Preferences” Side Tab
Creating a Basic Web Application Scan Policy

This should take you to the Preferences section
Creating a Basic Web Application Scan Policy

Step 12: Select “Global variable settings” from the Plugin pull down menu
Creating a Basic Web Application Scan Policy

Step 13: Check the “Probe services on every port” checkbox on “Global variable settings”
Creating a Basic Web Application Scan Policy

Step 14: Check the “Enable CGI scanning” checkbox on “Global variable settings”
Creating a Basic Web Application Scan Policy

Step 15: Check the “Enable experimental scripts” checkbox on “Global variable settings”
Creating a Basic Web Application Scan Policy

Step 16: Check the “Through test (slow)” checkbox on “Global variable settings”
Creating a Basic Web Application Scan Policy

Step 17: Set the “Report Verbosity” pull-down menu to “Verbose” on “Global variable settings”
Creating a Basic Web Application Scan Policy

Step 18: Set the “Report paranoia” pull down menu to “Normal” on “Global variable settings”
Creating a Basic Web Application Scan Policy

Step 19: Select “Login configurations” from the Plugin pull down menu
Creating a Basic Web Application Scan Policy

Step 20: Set the “HTTP account” and “HTTP password” on “Login configurations” to a value that is a common default in your environment.
Creating a Basic Web Application Scan Policy

Step 21: Select “Web Application Test Settings” from the Plugin pull down menu
Creating a Basic Web Application Scan Policy

Step 22: Make sure that the “Enable web application test” checkbox is checked on “Web Application Test Settings”
Creating a Basic Web Application Scan Policy

Step 23: The “Maximum run time” on “Web Application Test Settings” can be left at the default of 60 min. If you see timeouts in the result you may need to increase this value.
Creating a Basic Web Application Scan Policy

Step 24: Check the “Try all HTTP methods” on “Web Application Test Settings”
Creating a Basic Web Application Scan Policy

Step 25: Set the “Combinations of Arguments values” pull-down menu to “some pairs”
Creating a Basic Web Application Scan Policy

Step 26: Check the “HTTP Parameter Pollution” checkbox
Creating a Basic Web Application Scan Policy

Step 27: Set the “Stop at first flaw” pull-down menu to “look for all flaws” or “per parameter”
Creating a Basic Web Application Scan Policy

Step 28: Un-check the “Test embedded web servers” checkbox
Creating a Basic Web Application Scan Policy

Step 29: Select “Web mirroring” from the Plugin pull down menu
Creating a Basic Web Application Scan Policy

Step 30: Make sure that the “Follow dynamic pages” checkbox is checked on “Web mirroring”
Creating a Basic Web Application Scan Policy

Step 31: Select “HTTP login page” from the Plugin pull down menu
Creating a Basic Web Application Scan Policy

Step 32: Check “Automated login page search” checkbox is checked on “HTTP login page”

We will look at the other settings on this page in the Advanced Scan policy section
Creating a Basic Web Application Scan Policy

Step 33: Click on the Submit Button in lower right corner to save your policy
Create Basic Scan Template

Step 1: Click on the “Scan” tab on the top
Create Basic Scan Template

Step 2: Click on the “Add” button
Create Basic Scan Template

This should take you to the interface to create a new scan.
Create Basic Scan Template

Step 3: Name the Scan

**Name**: My Basic Web App Scan
**Type**: Template
**Policy**: My Basic Web App Tests
**Scan Targets**: 192.168.206.134
Create Basic Scan Template

Step 4: Set the scan Type to “Template”
Create Basic Scan Template

Step 5: Select the Basic Web App policy you just created
Create Basic Scan Template

Step 6: Enter your scan target IP, domain name or network range

- single IP address or comma-separated list (e.g., 192.168.0.1, 192.168.206.134)
- IP range (e.g., 192.168.0.1-192.168.0.255)
- subnet with CIDR notation (e.g., 192.168.0.0/24)
- or resolvable host (e.g., www.nessus.org).
Create Basic Scan Template

Step 7: Click on the “Save Template” button to save your scan template
Running Basic Scan Template

Step 1: Select your Basic Scan Template on the Scans Tab
Running Basic Scan Template

Step 2: Click on the Launch Button

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Status</th>
<th>Start Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.206.134 Basic</td>
<td>demo</td>
<td>Template</td>
<td>Never</td>
</tr>
<tr>
<td>192.168.206.134 DVWA</td>
<td>demo</td>
<td>Template</td>
<td>Never</td>
</tr>
<tr>
<td>My Basic Web App Scan</td>
<td>demo</td>
<td>Template</td>
<td>Never</td>
</tr>
</tbody>
</table>
Running Basic Scan Template

“Template was successfully launched” should appear at the top of the screen and a “running copy” of your scan will appear in the list with a progress bar.

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Status</th>
<th>Start Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.206.134 Basic</td>
<td>demo</td>
<td>Template</td>
<td>Never</td>
</tr>
<tr>
<td>192.168.206.134 DVWA</td>
<td>demo</td>
<td>Template</td>
<td>Never</td>
</tr>
<tr>
<td>My Basic Web App Scan</td>
<td>demo</td>
<td>Template</td>
<td>11/11/2012 16:54</td>
</tr>
<tr>
<td>My Basic Web App Scan</td>
<td>demo</td>
<td>Template</td>
<td>Never</td>
</tr>
</tbody>
</table>
Basic Scan Policy Demo

<table>
<thead>
<tr>
<th>Name</th>
<th>Visibility</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Web App Tests 2</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>DVWA Advanced Web App Tests</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>External Network Scan</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
<tr>
<td>Internal Network Scan</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
<tr>
<td>My Basic Web App Tests</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>My Basic Web App Tests 1</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>My DVWA Web App Tests 1</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>My DVWA Web App Tests 2</td>
<td>Private</td>
<td>demo</td>
</tr>
<tr>
<td>Prepare for PCI DSS audits</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
<tr>
<td>Web App Tests</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
</tbody>
</table>
Reviewing the Scan Report

Click on the Reports tab
Reviewing the Scan Report

To open the report double-click on your scan report or select it and click on the Browse button.
Reviewing the Scan Report

The scan report shows a list of IPs or domain names with indication of the number of High, Medium and Low Vulnerabilities and open ports.
Reviewing the Scan Report

Single click on the IP address to drill into each scanned device to get a list of open ports with vulnerability counts.

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>SVC Name</th>
<th>Total</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
<th>Open Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>udp</td>
<td>general</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>tcp</td>
<td>general</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>icmp</td>
<td>general</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>tcp</td>
<td>ssh</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>80</td>
<td>tcp</td>
<td>www</td>
<td>66</td>
<td>13</td>
<td>22</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td>143</td>
<td>tcp</td>
<td>imap?</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>445</td>
<td>tcp</td>
<td>cifs</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8080</td>
<td>tcp</td>
<td>www</td>
<td>23</td>
<td>0</td>
<td>2</td>
<td>19</td>
<td>2</td>
</tr>
</tbody>
</table>
Reviewing the Scan Report

Single click on a port row to drill into the port to get a list of vulnerabilities found

<table>
<thead>
<tr>
<th>Plugin ID</th>
<th>Name</th>
<th>Port</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>24011</td>
<td>WordPress Trackback Charset Decoding SQL Injection</td>
<td>www (80/tcp)</td>
<td>Medium</td>
</tr>
<tr>
<td>55976</td>
<td>Apache HTTP Server Byte Range DoS</td>
<td>www (80/tcp)</td>
<td>High</td>
</tr>
<tr>
<td>47830</td>
<td>CGI Generic Injectable Parameter</td>
<td>www (80/tcp)</td>
<td>Low</td>
</tr>
<tr>
<td>47832</td>
<td>CGI Generic On Site Request Forgery (OSRF)</td>
<td>www (80/tcp)</td>
<td>Medium</td>
</tr>
<tr>
<td>42427</td>
<td>CGI Generic SQL Injection (HTTP Headers)</td>
<td>www (80/tcp)</td>
<td>High</td>
</tr>
<tr>
<td>49067</td>
<td>CGI Generic HTML Injections (quick test)</td>
<td>www (80/tcp)</td>
<td>Medium</td>
</tr>
<tr>
<td>33817</td>
<td>CGI Generic Tests Load Estimation (all tests)</td>
<td>www (80/tcp)</td>
<td>Low</td>
</tr>
<tr>
<td>50494</td>
<td>CGI Generic Path Traversal (quick test)</td>
<td>www (80/tcp)</td>
<td>Medium</td>
</tr>
<tr>
<td>51972</td>
<td>CGI Generic Cross-Site Scripting (Parameters Names)</td>
<td>www (80/tcp)</td>
<td>Medium</td>
</tr>
<tr>
<td>51973</td>
<td>CGI Generic SQL Injection (Parameters Names)</td>
<td>www (80/tcp)</td>
<td>High</td>
</tr>
<tr>
<td>11139</td>
<td>CGI Generic SQL Injection</td>
<td>www (80/tcp)</td>
<td>High</td>
</tr>
</tbody>
</table>
Reviewing the Scan Report

Single click on a vulnerabilities to see the details

**Plugin ID:** 42427  
**Port / Service:** www (80/tcp)  
**Plugin Name:** CGI Generic SQL Injection (HTTP Headers)  
**Severity:** High

**Synopsis:** A CGI application hosted on the remote web server is potentially prone to SQL injection attack.

**Description**
By sending specially crafted HTTP headers to one or more CGI scripts hosted on the remote web server, Nessus was able to cause an error in the underlying database. This error suggests that the CGI script(s) are prone to SQL injection attack.

An attacker may be able to exploit this issue to bypass authentication, read confidential data, modify the remote database, or even take control of the remote operating system.

**Solution**
Modify the affected CGI scripts so that they properly escape arguments.

**See Also**
- [http://www.securiteam.com/securityreviews/5DP0N1P76E.html](http://www.securiteam.com/securityreviews/5DP0N1P76E.html)  
Reviewing the Scan Report

To find a specific vulnerability click on the “Show Filters” button
Reviewing the Scan Report

You have lot of options here. We are going to look for a specific Plugin by ID to check for Timeouts.
Reviewing the Scan Report

Looking at the details of Plugin #39470 will tell you if you need to increase your CGI run time.

<table>
<thead>
<tr>
<th>Plugin ID:</th>
<th>39470</th>
<th>Port / Service:</th>
<th>www (80/tcp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name:</td>
<td>CGI Generic Tests Timeout</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Synopsis:** Some generic CGI attacks ran out of time.

**Description**
Some generic CGI tests ran out of time during the scan. The results may be incomplete.

**Solution**
Run your run scan again with a longer timeout or less ambitious options:

- Combinations of arguments values = 'all combinations' is much slower than 'two pairs' or 'single'.
- Stop at first flaw = 'per port' is quicker.
Downloading Scan Report

To download your scan report select it in the reports list and click on the “Download” button.
or when viewing the report click on the download button. Note that any filters current applied will be applied to the downloaded report.
Downloading Scan Report

Select a Download format

- .nessus & .nessus(v1) can edited and re-imported (XML)
- HTML Detailed or HTML Executive Reports
- RTF
- Custom
### List of hosts

<table>
<thead>
<tr>
<th>Host IP Address</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.206.134</td>
<td>High Severity problem(s) found</td>
</tr>
</tbody>
</table>

#### 192.168.206.134

**Scan Time**
- Start time: Sun Jan 22 01:55:10 2012
- End time: Sun Jan 22 18:26:17 2012

#### Number of vulnerabilities
- Open ports: 4
- High: 13
- Medium: 24
- Low: 64

#### Remote host information
- Operating System: Linux Kernel 2.6 on Ubuntu 10.04 (lucid)
- NetBIOS name: OWASPBWA
- DNS name: 

---

**Synopsis:**
It was possible to obtain traceroute information.
## List of Plugin IDs

The following plugin IDs have problems associated with them. Select the ID to review more detail.

<table>
<thead>
<tr>
<th>PLUGIN ID#</th>
<th># OF ISSUES</th>
<th>PLUGIN NAME</th>
<th>SEVERITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>26968</td>
<td>1</td>
<td>TikiWiki tiki-graph_formula.php f Parameter Arbitrary Command Execution</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>15780</td>
<td>1</td>
<td>phpBB viewtopic.php highlight Parameter SQL Injection</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>17225</td>
<td>1</td>
<td>phpBB &lt;= 2.0.12 Multiple Vulnerabilities</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>13655</td>
<td>1</td>
<td>phpBB &lt; 2.0.9 Multiple Vulnerabilities</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>11938</td>
<td>1</td>
<td>phpBB &lt; 2.0.7 Multiple Script SQL Injection</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>25116</td>
<td>1</td>
<td>myGallery mygallerybrowser.php myPath Parameter Remote File Inclusion</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>48927</td>
<td>1</td>
<td>CGI Generic SQL Injection Detection (potential, 2nd order, 2nd pass)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>51973</td>
<td>1</td>
<td>CGI Generic SQL Injection (Parameters Names)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>42427</td>
<td>1</td>
<td>CGI Generic SQL Injection (HTTP Headers)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>42479</td>
<td>1</td>
<td>CGI Generic SQL Injection (2nd pass)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>11139</td>
<td>1</td>
<td>CGI Generic SQL Injection</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>44967</td>
<td>1</td>
<td>CGI Generic Command Execution (time-based)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>55976</td>
<td>1</td>
<td>Apache HTTP Server Byte Range DoS</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>33821</td>
<td>2</td>
<td>.svn/entries Disclosed via Web Server</td>
<td>Medium Severity problem(s) found</td>
</tr>
<tr>
<td>51846</td>
<td>1</td>
<td>WordPress Trackback Shortcode Denying SQL Injection</td>
<td>Medium Severity problem(s) found</td>
</tr>
</tbody>
</table>
HTML Executive Report

Executive Summary:

TOP 10 HOSTS with ISSUES

<table>
<thead>
<tr>
<th>Hosts</th>
<th>Severity</th>
<th># of Issues</th>
<th>Synopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.206.134</td>
<td>High</td>
<td>1</td>
<td>CGI Generic SQL Injection (2nd pass) A web application is potentially vulnerable to SQL injection.</td>
</tr>
<tr>
<td>13655</td>
<td>High</td>
<td>1</td>
<td>phpBB &lt; 2.0.9 Multiple Vulnerabilities A remote web application is vulnerable to SQL injection. myGallery mygallerybrowser.php myPath Parameter</td>
</tr>
</tbody>
</table>
# HTML Custom Report

**Report:** 192.168.206.134 Basic

**Scan Time:**
- Start Time: Sun Jan 22 01:55:09 2012
- End Time: Sun Jan 22 18:26:19 2012

## List of Hosts

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.206.134</td>
<td>High</td>
<td>Fail</td>
<td>37</td>
<td>13</td>
<td>24</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>OWASPBWA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168.206.134 Basic Totals</td>
<td>1 of 1 IPs Failed</td>
<td>37</td>
<td>13</td>
<td>24</td>
<td>64</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

## Information about the Scan

- The Low Vul. Category includes informational items that may or may not be vulnerabilities; these in some cases require manual checking.
- The PCI Failing column is the sum of the following:
  - High Vul.
  - Medium Vul.
# RTF Report

## NESSUS REPORT

### List of PlugIn IDs

The following plugin IDs have problems associated with them. Select the ID to review more detail.

<table>
<thead>
<tr>
<th>PLUGIN ID#</th>
<th>#</th>
<th>PLUGIN NAME</th>
<th>SEVERITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>55976</td>
<td>1</td>
<td>Apache HTTP Server Byte Range DoS</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>51973</td>
<td>1</td>
<td>CGI Generic SQL Injection (Parameters Names)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>48927</td>
<td>1</td>
<td>CGI Generic SQL Injection Detection (potential, 2nd order, 2nd pass)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>44967</td>
<td>1</td>
<td>CGI Generic Command Execution (time-based)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>42479</td>
<td>1</td>
<td>CGI Generic SQL Injection (2nd pass)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>42427</td>
<td>1</td>
<td>CGI Generic SQL Injection (HTTP Headers)</td>
<td>High Severity problem(s) found</td>
</tr>
<tr>
<td>26068</td>
<td>1</td>
<td>TikiWiki tiki-graph_formula.php f Parameter Arbitrary Command</td>
<td>High Severity problem(s) found</td>
</tr>
</tbody>
</table>
.nessus Export

<?xml version="1.0" ?>
<NessusClientData_v2>
  <Policy>
    <policyName>Basic Web App Tests</policyName>
    <policyComments></policyComments>
    <Preferences>
      <ServerPreferences>
        <preference>
          <name>max_simult_tcp_sessions</name>
          <value>unlimited</value>
        </preference>
        <preference>
          <name>use_mac_addr</name>
          <value>no</value>
        </preference>
        <preference>
          <name>plugin_set</name>
          <value>17803;38808;44943;16058;39500;14325;10702;10777;40886;24698;17312;11769;11234;11985;10569;10447;10830;36088</value>
        </preference>
        <preference>
          <name>TARGET</name>
          <value>192.168.206.134</value>
        </preference>
        <preference>
          <name>throttle_scan</name>
          <value>yes</value>
        </preference>
      </ServerPreferences>
    </Preferences>
  </Policy>
</NessusClientData_v2>
.nessus v1 Export

```xml
<?xml version="1.0" ?>
<NessusClientData>
<Targets>
<Target>
:selected>yes</selected>
:type>hostname</type>
:value>192.168.206.134</value>
</Target>
</Targets>
<Policies>
<Policy>
<policyName>Basic Web App Tests</policyName>
<policyComments></policyComments>
<Preferences>
<ServerPreferences>
<preference>
<name>max_simult_tcp_sessions</name>
<value>unlimited</value>
</preference>
<preference>
<name>use_mac_addr</name>
<value>no</value>
</preference>
<preference>
<name>plugin_set</name>
<value>17803;38808;44943;16058;39500;14325;10702;10777;40886;24698;17312;11769;11234;11985;10569;10447;10830;36088</value>
</preference>
<preference>
<name>TARGET</name>
</preference>
</ServerPreferences>
</Preferences>
</Policy>
</Policies>
</NessusClientData>
```
Creating an Advanced Web Application Scan Policy

- The goal is to create a specific policy for scanning a known Web application.
- This will be based on the Basic Web Application Scan Policy we just created.
- Our target for this example will be the “Damn Venerable Web App” on the “OWASP Broken Web Applications” VMWare image.
Creating an Advanced Web Application Scan Policy

Step 1: Go to the Policies Tab and select the Basic Web Applications policy you just created.
Creating an Advanced Web Application Scan Policy

Step 2: Click on the “Copy” button. This will create a new Policy called “Copy of …”
Creating an Advanced Web Application Scan Policy

Step 3: Select the new policy “Copy of ...”
Creating an Advanced Web Application Scan Policy

Step 4: Click on the Edit Button
Creating an Advanced Web Application Scan Policy

This will open the Edit Policy screen
Creating an Advanced Web Application Scan Policy

Step 5: Change the policy name

![Policy Configuration](image)
Creating an Advanced Web Application Scan Policy

Step 6: Change the Visibility to Private
Creating an Advanced Web Application Scan Policy

Step 6: Uncheck all port scanners. We know what port we want
Creating an Advanced Web Application Scan Policy

Step 7: Set the “Port Scan Range” to only the ports the target Web application is using. In our example we are running on port 80
Creating an Advanced Web Application Scan Policy

Step 8: Select “HTTP login page” from the Plugin pull down menu
Creating an Advanced Web Application Scan Policy

We will need to do some reconnaissance to get the values for these fields.
Creating an Advanced Web Application Scan Policy

Step 9: Find the Login Screen

/dvwa/login.php
Creating an Advanced Web Application Scan Policy

Step 10: Enter the Login page path (not the full URL)

![Login page form fields](image)
Creating an Advanced Web Application Scan Policy

Step 11: View source on the login page to find the “Login Form” (action) and “Login Form Method”
Creating an Advanced Web Application Scan Policy

Step 12: Enter the “Login form” path (not full URL) based on the “action” attribute of the form.
Creating an Advanced Web Application Scan Policy

Step 13: Enter the “Login from method” based on the “method” attribute of the form
Creating an Advanced Web Application Scan Policy

Step 14: Determine the “Login form fields” and values by trapping the login with tamper data or a Web proxy

- **Cookie**: security=high; PHPSESSID=dq3thqopdjvljie5nfibhg5i3...
- **Content-Type**: application/x-www-form-urlencoded
- **Content-Length**: 41
- **POSTDATA**: username=admin&password=admin&Login=Login
Creating an Advanced Web Application Scan Policy

Step 15: Enter the “Login from fields”
Substitute %USER% for the user name
Substitute %PASS% for the password
Creating an Advanced Web Application Scan Policy

Step 16: Uncheck “Automated login page search” since we have told Nessus where the login form is located
Creating an Advanced Web Application Scan Policy

Step 17: Find criteria to confirm login
  • Authenticated page path
  • Text in the page HTML
Creating an Advanced Web Application Scan Policy

Step 18: Enter the “Check authentication on page” path

- Check authentication on page: /dvwa/index.php
- Follow 30x redirections (# of levels): 2
- Authenticated regex: [Ll]ogout
Creating an Advanced Web Application Scan Policy

Step 19: Enter the “Authentication regex.”
This pattern allows the “L” to be case insensitive
Creating an Advanced Web Application Scan Policy

Step 20: Select “Web Application Test Settings” from the Plugin pull down menu.
Creating an Advanced Web Application Scan Policy

Step 21: Increase the “Maximum run time” value. Remember that the Basic Policy timed out.

Maximum run time (min): 240
Creating an Advanced Web Application Scan Policy

Step 22: Select “Web mirroring” from the Plugin pull down menu
Creating an Advanced Web Application Scan Policy

Step 23: Set the Start page to go to the target Web Application

- Number of pages to mirror: 1000
- Maximum depth: 6
- Start page: /dvwa/login.php
- Excluded items regex: logout\.(php|\|phpmyadmin\|WebGoat\|ghost/)
- Follow dynamic pages: ✔
Creating an Advanced Web Application Scan Policy

Step 24: Set the “Exclude Items regex” to avoid logging out or going to places that we don’t want to test.

Number of pages to mirror: 1000
Maximum depth: 6
Start page: /dvwa/login.php
Excluded items regex: logout\.php|/phpmyadmin|/WebGoat|/ghost/
Follow dynamic pages: ✔️
Creating an Advanced Web Application Scan Policy

Step 25: Click on the Submit Button in lower right corner to save your policy
Create Advanced Scan Template

Step 1: Click on the “Scan” tab on the top
Create Advanced Scan Template

Step 2: Click on the “Add” button
Create Advanced Scan Template

This should take you to the interface to create a new scan.
Create Advanced Scan Template

Step 3: Name the Scan

Name: My DVWA Web App Scan
Type: Template
Policy: My DVWA Web App Tests
Scan Targets: 192.168.206.134
Create Advanced Scan Template

Step 4: Set the scan Type to “Template”
Create Advanced Scan Template

Step 5: Select the Advanced Web App policy you just created
Create Advanced Scan Template

Step 6: Enter your scan target IP, domain name or network range

- single IP address or comma-separated list (e.g., 192.168.0.1, 192.168.206.134)
- IP range (e.g., 192.168.0.1-192.168.0.255)
- subnet with CIDR notation (e.g., 192.168.0.0/24)
- or resolvable host (e.g., www.nessus.org).
Create Advanced Scan Template

Step 7: Click on the “Save Template” button to save your scan template
## Advanced Scan Demo

![Nessus Interface](image-url)

### Policies Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Visibility</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Web App Tests 2</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>DVWA Advanced Web App Tests</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>External Network Scan</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
<tr>
<td>Internal Network Scan</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
<tr>
<td>My Basic Web App Tests</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>My Basic Web App Tests 1</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>My Basic Web App Tests 3</td>
<td>Shared</td>
<td>demo</td>
</tr>
<tr>
<td>My DVWA Web App Tests 1</td>
<td>Private</td>
<td>demo</td>
</tr>
<tr>
<td>My DVWA Web App Tests 2</td>
<td>Private</td>
<td>demo</td>
</tr>
<tr>
<td>Prepare for PCI DSS audits</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
<tr>
<td>Web App Tests</td>
<td>Shared</td>
<td>Tenable Policy Distribution Service</td>
</tr>
</tbody>
</table>
Reviewing the Report for OWASP Top Items

**A1 - Injection**
- SQL Injection (CGI abuses) > 11139, 42424, 42426, 42427, 42479, 43160, 51973
- XML Injection (CGI abuses) > 46196
- HTTP Header Injection (CGI abuses: XSS) > 39468, 49067
- Cookie Injection > 44135 (CGI abuses)

**A2 - Cross-Site Scripting (XSS)**
- Cross-Site Scripting (CGI abuses: XSS) > 10815, 39466, 42425, 47831, 46193, 49067, 51972

**A3 - Broken Authentication and Session Management**
- Authentication not over SSL > 26194, 34850
- Is SSL Implement Properly > 15901, 20007, 26928, 35291, 42053, 42873, 42880, 53491, 53360, 56043, 56284, 56984, 57041
Reviewing the Report for OWASP Top Items Cont.

**A4 - Insecure Direct Object References**
- Browsable Web Directories > 40984
- Path Transversal (CGI abuses) > 50494
- Parameters identified for manual testing > 40773, 44134, 47830 *

**A5 - Cross-Site Request Forgery (CSRF)**
- CGI Generic On Site Request Forgery (OSRF) > 47832
- Specific Product checks with known CSRF Vulnerabilities

**A6 - Security Misconfiguration**
- Covered by Nessus Audit Checks in the ProfessionFeed
- Identifies Open ports and services for manual review
- Many checks for default accounts and passwords
Reviewing the Report for OWASP Top Items Cont.

**A9 - Insufficient Transport Layer Protection**
- Authentication not over SSL > 26194, 34850
- Is SSL Implement Properly > 15901, 20007, 26928, 35291, 42053, 42873, 42880, 53491, 53360, 56043, 56284, 56984, 57041
- Secure Cookie Use > 49218, 84832

**A10 - Unvalidated Redirects and Forwards**
- CGI Generic Open Redirection > 47834
Reviewing the Report for 2007 OWASP Top Items

2007 A3 - Malicious File Execution
- Command Execution (CGI abuses) > 39465, 44967

2007 A6 - Information Leakage and Improper Error Handling
- Directory Traversal (CGI abuses) > 39467, 46195, 46194
- File Inclusion (CGI abuses) > 39469, 42056, 42872
- Server Side Includes (CGI abuses) > 42423, 42054
- Error Messages > 40406, 48926, 48927
Other Nessus CGI checks

- Format String (CGI abuses) > 42055
- Cookie Manipulation (CGI abuses) > 44136
- Additional attacks (CGI abuses) > 44134, 47830, 47832, 47834
Resources

Nessus Website
http://www.nessus.org/products/nessus

My Email
rikjones@computer.org