Vulnerability Assessment Executive Summary WebPower Application

Primeon performed a DeepSource vulnerability assessment on WebPower Web application between June 1, 2002 and July 1, 2002. During the assessment, Primeon applied its advanced tools, comprehensive application security knowledge base, and security vulnerability assessment methodology for the detection of security issues and exposures within the WebPower code base and the related runtime platform environment. The assessment results show that Webpower is vulnerable for many types of attacks.

This executive summary summarizes a DeepSource Web Application Security Vulnerability Assessment Report for the WebPower application. A DeepSource assessment is an independent review that assesses the architectural design, business logic and security posture of a developed application. Deepsource assessment process provides a true independent look at an application's code design, use of technology components and security flaws. The end result of a DeepSource assessment is a comprehensive easy-to-read actionable report – For each security exposure instance, DeepSource Reports provide the following information:

- Exact exposure location within the code base, including module and actual lines of highlighted code
- Precise technical description of the exposure
- Likely operational impact when exposure is exploited
- Specific remediation guidance to close the exposure

The following section summarizes the security vulnerability findings in the architectural design and application implementation of WebPower.

1 Assessment Findings

1.1 Findings in Architectural Design

We have identified **three (3) vulnerabilities related to architectural design**. The three vulnerabilities are:

- Incomplete access control mechanism
    - Flawed identity management - An attacker may impersonate any WebPower user to access the financial data belonging to this user, such as account banances and financial reports.
    - Flawed entitlement management – An attacker with a guest account may operate privileged functions such as view transaction histories for any user, view profile information including account number, SSN number, address for any user, etc.
- Unsafe input data validation
    - Using the input validation vulnerabilities, an attacker may create a "phishing" attack to obtain account numbers, passwords and other sensitive data of WebPower users.

- Unsafe database query
    - By manipulating the database queries, an attack may damage the integrity of business data stored in the backend database. For example, an attack may create false transactions or remove or alter the financial records of clients.

Understanding these vulnerabilities and conducting recommended remediation activities will not only help WebPower's development team to greatly reduce the security risk level of the WebPower application, it will also help the team maintain a low security risk level during follow-on development efforts.

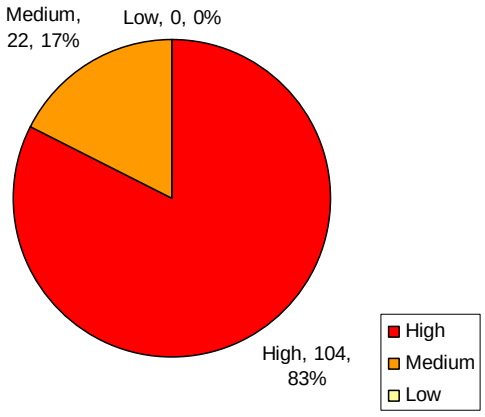
1.2 Findings in Application Implementation

We have also identified **19 vulnerability patterns including 126 instances of vulnerabilities related to application implementation**. These results represent a certain level of business risk to the WebPower application. Table 1 presents the identified vulnerability patterns with the ratings of risk, resource requirement for remediation, and effort to exploit the vulnerability, along with the numbers of instances in each pattern. Table 2 lists the rating definitions for risk, resource requirement and effort to exploit.

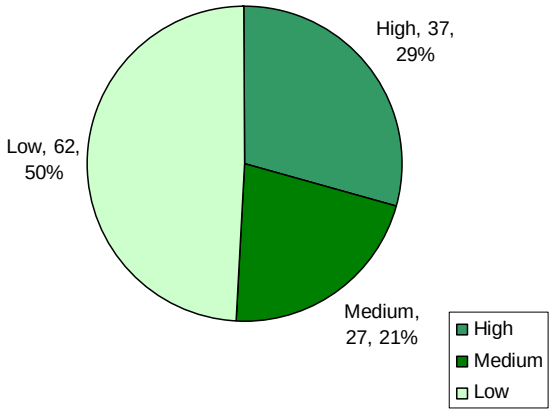
Table 1 – WebPower Vulnerabilities by Pattern (as of July 2002)

No.	Vulnerability Patterns	Risk	Resource Requirement	Effort to Exploit	Number of Instances
	Server/Services – iPlanet Web Server, and WebLogic Application Server				
	Authorization				
1	Testing Files	Medium	Low	High	1
2	Back-Up Files/Temp Files	Medium	Low	High	7
3	Include Files	Medium	Medium	High	11
4	Debug Functions	High	Low	High	1
5	Configuration Files and Template Files	Medium	Low	Medium	2
	Configuration Management				
6	Default Modules	High	Low	Low	10
7	Profiling	Medium	Low	Medium	1
8	Unauthorized access to administration interfaces	High	Medium	Medium	2
	Exception Management				
9	Stack Info Exposure	High	Low	Medium	1
10	Compiler Error Info Exposure	High	Low	Medium	1
	Front-End – Static Web Content, Servlets, JSP Page, and Supporting Java Class				
	Input Validation				
	Cross-Site Scripting (XSS)				
11	XSS without Script Tags	High	High	Low	3
12	XSS with Script Tags in JSP	High	Low	Medium	12
13	XSS with Script Tags in Java	High	Low	Medium	17
	Authorization				
14	Unprotected JSP Views	High	High	Low	18
15	Internal Used Servlet Access Control Flaw	High	Medium	Low	14
	Parameter Manipulation				
16	Request Parameter Manipulation in JSP	High	High	Medium	1
17	Request Parameter Manipulation in Java	High	High	Medium	12

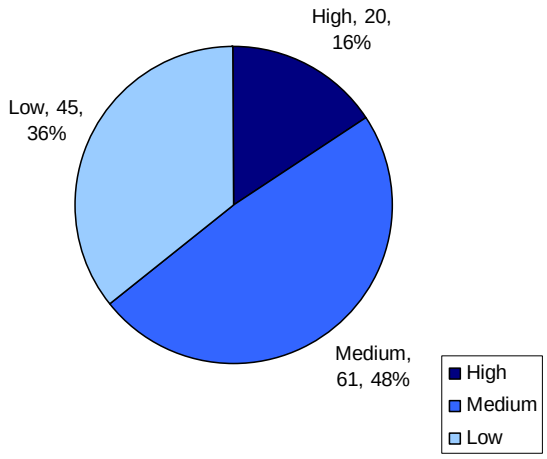
No.	Vulnerability Patterns	Risk	Resource Requirement	Effort to Exploit	Number of Instances
	EJBs				
	Input Validation				
18	SQL Injection	High	Low	Medium	9
	Database				
	Input Validation				
19	Denial of Service	High	High	Medium	3
	Total				**126**



Risk



Resource Requirement



Exploit Effort

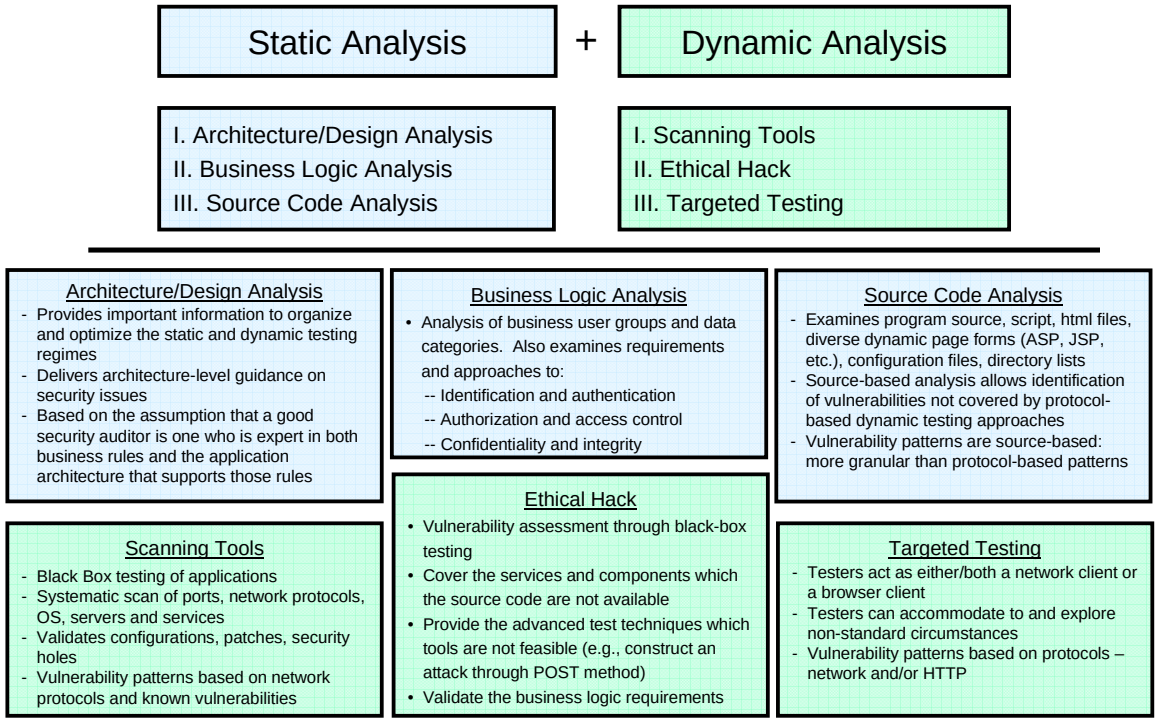
Table 2 – Rating Definition

Rating	Definition of Risk Rating	Definition of Resource Requirement	Definition of Effort to Exploit
High	Deficiency creates a vulnerability that could result in a loss of system control via the compromise of administrative accounts or other system functions.	Recommendation requires the purchase of hardware or software and/or requires significant research / implementation activities	To exploit the weakness requires a high level of expertise and advanced knowledge of programming/application design.
Medium	Deficiency creates an exposure to a larger, but limited loss of confidentiality or integrity, as the result of many user accounts being compromised or restricted system functions being accessed.	Recommendation may require the purchase of hardware or software and/or requires moderate research/implementation activities	Requires medium level of effort. No tools are available but sample code or other similar exploits are known.
Low	Deficiency creates limited exposure to the compromise of user accounts or unauthorized access to data.	Recommendation may require the purchase of minor hardware/software and/or requires minor research/ implementation activities	Easy to exploit with known methods or tools with minimal modification.

2 The DeepSource Assessment Methodology

To substantially solve the application security challenge, an approach is required that goes far beyond the weapons in a hackers' arsenal. To this end, most application security experts agree that the combination of selected tools, accumulated application vulnerability assessment expertise in source code level (security vulnerability knowledge base) and a proven assessment process is considered the best way to identify Web application security exposures.

The following figure presents a high level structure of Primeon DeepSource methodology for application security vulnerability assessments. The assessment is based on the combined static analysis and dynamic analysis to achieve a highly efficient process for identifying security vulnerabilities.



The key differentiation of this security assessment process is its integration of both black-box testing (ethical hacking or penetration testing) and white-box testing (targeted testing). The targeted testing uses the testing plan from an attack tree model and which in turn are generated from the static analysis components.

3 About Primeon

Primeon is the leader in enterprise-wide application security, planning, assessment and pen-testing. Primeon is the only company to offer a complete solution for identifying application exposures, inefficiencies, "malicious code" and subsequent remediation measures in a simple to read actionable output report. Leveraging 10 years of application analysis experience that exceeds over 500 million lines of source code and nearly all computing architectures & languages, Primeon is able to offer the most thorough application security solution on the market. In addition to scanning tool & pen-testing components, Primeon's DeepSource(tm) includes analysis of source code (100% code coverage), comprehensive knowledge base plus accelerated interactive testing that provides unmatched levels of exposure identification and risk mitigation with our fully independent application audits. Founded in 1995 as a premier application assessment company, Primeon's satisfied clients include many of the Fortune 500 and over 50 of the largest financial services enterprises on Wall Street who trust Primeon with their entire application portfolios.