External Vulnerability Assessment
Client: Business Solutions
January 23rd 2014
Table of Contents

1.0 - Executive Summary .................................................................................................................................................. 2
  1.1 Summary of Results .................................................................................................................................................. 2
2.0 - Findings .................................................................................................................................................................... 3
  2.1 Server Vulnerabilities ................................................................................................................................................ 3
  2.2 Cryptographic Vulnerabilities .................................................................................................................................... 4
  2.3 Web Application Vulnerabilities ............................................................................................................................... 6
3.0 - Recommendations ...................................................................................................................................................... 9
4.0 - Conclusion .................................................................................................................................................................. 10
Appendix A: Risk Rating Scale ........................................................................................................................................... 10
Appendix B: Definitions ...................................................................................................................................................... 11
1.0 - Executive Summary

Acumen Innovations was contracted by Business Solutions to perform an external vulnerability assessment. The purpose of the assessment was to determine the vulnerabilities an external attacker could exploit to gain access to the web server without any previous knowledge of the organization. The assessment method focused on minimizing the server’s load and evading detection mechanisms. For sensitive areas of the application, manual testing was required since discovery of previously unknown vulnerabilities could have caused a denial of service.

The scope of the assessment included all services hosted on the server, such as the companies’ web application, FTP, Remote Connection Servers, and cryptographic communications. Social media was also investigated for IT personnel in order to identify information leakage. False positives were eliminated where possible by manually interacting with the application in areas where this could be done passively.

1.1 - Summary of Results

A scan of the server revealed Web, FTP, and Remote Access Services. Through version identification it was discovered that the web server was running an outdated application which is susceptible to multiple high and critical risk vulnerabilities that may enable full system compromise. Likewise, the FTP server was running an outdated service with several low and medium vulnerabilities that could allow an attacker to read and write sensitive files within the server.

Analysis of the cryptographic protocols used to secure communication resulted in the discovery of several low risk vulnerabilities in which an attacker could break the encryption and recover short amounts of text, such as a session cookie. This could lead to session hijacking and compromise administrative users.

Multiple critical vulnerabilities were found in the web application. These vulnerabilities included SQL injection attacks, Stored Cross Site Scripting (XSS) and Local file inclusion, all of which could provide an attacker with administrative control of the server and access to confidential information. While network and web application layer firewalls were encountered, these were easily bypassed due to improper configuration.

Social media profiles of the IT personnel revealed information which could be leveraged by an attacker in order to reset login credentials and gain access to sensitive areas of the web application.

Risk Assessment concluded that the system’s security status was **critical**. This is due to many possible attack vectors; some of the vulnerabilities require minimal skill in order to exploit. To reduce the attack surface, a multi-pronged approach is needed. In the short-term, all the services should be updated to the latest versions and firewall rules reconfigured to mitigate attacks. A thorough review of the system and firewall logs must be conducted to identify previously undiscovered intrusions, and a review of the web application’s source code must carried out in order to eliminate false positives and discover additional vulnerabilities (if present). A thorough review of the Organization’s Security Management Process and Policy is strongly recommended, including the patch management process which should adhere to guidelines outlines in NIST SP 800-40.
2.0 - Findings

The following is a detailed list of the vulnerabilities encountered during the assessment. Sample attacks have also been included which may be used to verify the existence of the vulnerability. Please note, some of these vulnerabilities were not verified since doing so could lead to system compromise or a Denial of Service which was outside the scope of the assessment.

In accordance with NIST SP 800-30, as well our internal risk rating system, vulnerabilities are ranked based on complexity required to run a successful attack and the impact it would have on the system. For more information please refer to Appendix A: Risk Rating Scale.

2.1 - Server services

The following vulnerabilities were identified using a port scan to identify open ports and banner grabbing to fingerprint the listening services.

2.1.1 Outdated Web server – IIS 6.0

Rating: High

Description: Web server running on port 80 was identified as IIS 6.0, which is an outdated and vulnerable service.

Impact: Microsoft’s Internet Information Services version 6.0 suffers from multiple vulnerabilities that will compromise system availability, confidentiality and integrity. The vulnerabilities may allow a remote unauthenticated user to execute code remotely via various buffer overflow vulnerabilities. In addition, this server is vulnerable to a powerful DoS attack in which a single user can send malformed connection requests which tie up system resources, and therefore enable them to render the system inoperable using minimal power and bandwidth.

Mitigation: Update to the latest version of IIS which, at the time of this writing, is 8.0. In addition, a thorough review of the organization’s patch management procedure is required to ensure systems are patched and updated as soon as updates become available. For critical systems, patching and updating should first be done in a cloned, separate and isolated system to measure impact the updates may have on functionality before being applied to the live production systems.

2.1.2 Vulnerable FTP Server – Pro FTPD 1.3.4rc3

Rating: Moderate

Description: FTP server running on port 21 leaks version information via header. Examination of this particular version reveals vulnerabilities in auxiliary modules and in the main daemon.
Impact: Pro FTPD 1.3.4rc3 suffers from vulnerabilities in the mod_copy module which allows remote unauthenticated users to read and write to arbitrary files, which affects system confidentiality and integrity. In addition, lack of input sanitation allows users to inject code into authentication logs, which can then be executed leveraging the mentioned vulnerability.

Mitigation: If the mod_copy module is enabled, it should be disabled immediately. Due to the multiple vulnerabilities found in other versions of ProFTPD, it is recommended that a change is made to the latest version of VSFTPD instead. Finally, a review of the patch management procedure is needed to ensure patches are applied as they become available.

2.1.3 Directory Listing Enabled

Rating: Low

Description: Directory listing is enabled in the web server.

Impact: Directory listing aids attackers in reconnaissance by listing all the contents of the current folder. While this vulnerability may not have immediate direct security concerns, the information contained within the folder, such as .htaccess files coupled with permission misconfigurations could be used by an attacker to advance an attack.

Mitigation: Disable directory listing.

2.1.4 Network Firewall Misconfiguration

Rating: Low

Description: Network Firewall rules allow connections to a Remote Desktop Server from any IP.

Impact: This vulnerability may potentially allow restricted users to brute force a login which will provide access to the listening service and, depending on network configuration, the organization’s internal network.

Mitigation: If access to a Remote Desktop Server is crucial for normal operations, firewalls should be configured with a whitelist which only allows specific IP addresses to connect to the listening system. This will help prevent unauthorized access, reduce the attack surface and keep the service hidden from unauthorized users.

2.2 Cryptography

Review of the application’s HTTPS communications and certificates revealed several vulnerabilities which may allow attackers to compromise message integrity, and, in some cases, lead to session hijacking. While these vulnerabilities have a low-moderate rating due to the complexity of the attacks (see Appendix A for the complete Risk Rating Scale), from a cryptographic perspective they have a risk rating of high since encryption is completely broken.
2.2.1 - SSL Redirect

Rating: Moderate

Description: This *downgrade attack* enables forwarding of traffic intended for port 443 to port 80, forcing plain text connections.

Impact: This type of *MITM* attack allows an attacker to redirect HTTPS traffic to a HTTP port. Since the server accepts connections on both of these ports, it enables the attacker to view all intended encrypted communication in plaintext and therefore completely break both integrity and confidentiality. Unlike other forms of MITM attacks on HTTPS such as certificate forgery, this attack is usually transparent and completely unknown to the user.

Mitigation: Employ the *HTTP Strict Transport Security (HSTS)* security policy mechanism which allows web servers to declare that the web browser should only interact using secure https connections and never the http protocol, thus ensuring prevention of downgrade attacks such as this one.

2.2.2 - Padding Oracle On Downgraded Legacy Encryption (POODLE)

Rating: Moderate

Description: The web server supports SSL 3.0, which allows an attacker to perform a MITM attack and hijack the encryption negotiation in order to force both the browser and server to encrypt all traffic using the old and broken SSL 3.0 protocol.

Impact: Using the POODLE vulnerability, an attacker can force both client and server to use a weak encryption mechanism which allows for 1 byte of plaintext to be recovered in as little as 256 requests. While this may not seem like a significant attack, a 16 byte session cookie could be recovered in as little as 4096 requests which could lead to *session hijacking* of a privileged account.

Mitigation: Since the vulnerability stems from the server’s support of SSL 3.0, disabling this protocol is strongly recommended. However, if support is necessary due to backwards compatibility issues, then *TLS_FALLBACK_SCSV* must be implemented in the server to make the downgrade attack impossible.

2.2.3 - TLS 1.0 Support, but not TLS 1.2

Rating: Low

Description: Currently the server supports TLS 1.0 but not the latest version 1.2. Like SSL 2 and 3, TLS 1.0 has been broken and currently has many attacks open against it.

Impact: In addition to some implementations being vulnerable to POODLE, the *IV* used for *Cipher Block Chaining (CBC)* mode is predictable, as the IV for the next record is the last cipher text block of
the current cipher. This means that it is not *CPA secure*, and attacks like *BEAST* are able to recover some plaintext. When TLS is implemented on IMAP, the attacker will know part of the plaintext being transmitted, and timing attacks may sometimes be carried out.

**Mitigation:** Disable TLS 1.0 and upgrade support to TLS 1.2.

### 2.2.4 - RC4 Support

**Rating:** Low

**Description:** This server accepts the RC4 cipher.

**Impact:** This cipher takes a variable size seed and expands it to 2048 bits. The biggest issue with RC4 occurs when the IV’s used as a seed are related. Perhaps the best example of the weakness of this cipher can be seen when analyzing the wireless WEP encryption which has long been completely broken and also uses RC4. In both cases, weaknesses in the scheme rose from similarities in the IVs used. It is also the slowest as far as performance, making many other encryption standards more practical. This vulnerability affects message integrity and confidentiality.

**Mitigation:** Disable RC4 support.

### 2.3 Web Application

The Web application was analyzed using both automated scans and manual parameter fuzzing where required. Several high risk and critical vulnerabilities were encountered which could allow a remote attacker to compromise the server. Although an application level firewall was encountered, it was bypassed using some common obfuscation techniques.

Locations of the vulnerable services are indicated below, along with sample payloads where applicable.

#### 2.3.1 - SQL Injection

**Rating:** Critical

**Location:** [www.businesssolutions.com/blog/getpost.php?uid=1](http://www.businesssolutions.com/blog/getpost.php?uid=1)

**Description:** The parameter *uid* in the *getpost.php* service suffers from an SQL injection vulnerability. The vulnerability was manually validated using passive techniques. The flaw can be triggered via GET requests, with the payloads *uid=(3-2), (5-4), and (0+1)*, all which returned the same page as *uid=1*.

**Impact:** Depending on the privileges granted to the database user, injection of sql queries may lead to confidential information disclosure and even server compromise via uploaded shells. There are
many open source tools which will automate the process of finding and actively exploiting sql injection flaws, which means anyone with minimal skill can exploit this vulnerability.

Mitigation: SQL injection is a matter of improper input sanitation. A thorough code review of the web services is strongly recommended and invalid character stripping of user supplied input required. Furthermore, application layer firewalls must be properly configured to prevent these and obfuscation attempts.

### 2.3.2 - Cross Site Scripting (XSS)

**Rating:** *Critical*

**Location:** [www.businesssolutions.com/blog/postcomment.php](http://www.businesssolutions.com/blog/postcomment.php)

**Description:** The postcomment.php service suffers from a stored cross site scripting vulnerability which is triggered when a user injects malicious html such as the following:

```html
<IMG SRC=javascript:alert('XSS')>
```

In this case, the application firewall detected the malicious payload and dropped the request. Using hex of encoding, it was possible to bypass the firewall and successfully post

```html
<IMG SRC=&#x6A&#x61&#x76&#x61&#x73&#x72&#x74&#x28&#x27&#x58&#x53&#x27&#x29>
```

**Impact:** While this specific payload only produced a pop up window, consider the following payload:

```html
```

This payload would send the cookies of all users who visited the comments page to an attacker. If this was a logged in system administrator, this would result in a session hijacking where the attack could escalate further.

**Mitigation:** Like SQL injection, cross site scripting vulnerabilities arise from lack of user input sanitation. All user input should be properly filtered to ensure no invalid characters are injected into the page. A code review is needed to determine if further vulnerabilities exist.

### 2.3.3 - Local File Inclusion

**Rating:** *High*


**Description:** The getreport.pdf service allows users to read files in the server through the pdf parameter. This vulnerability was discovered by analyzing the file structure and using the payload getreport.pdf?pdf=./../images/home.jpg%00
Impact: Local file inclusion vulnerabilities can allow a malicious attacker to read sensitive files within the server. A malicious user would use a payload such as `../../../etc/passwd` which may return the passwd file in an apache system and allow him to enumerate users.

Mitigation: To prevent LFI vulnerabilities, user input must never be passed to any filesystem/framework API. Code review is needed, and the application layer firewall must have its rule set reviewed.

2.3.4 Session ID Prediction

Rating: Moderate

Location: www.businesssolutions.com/admin/login.php

Description: By utilizing a proxy, it was discovered that the session id generated after every login attempt are a base64 encoded string composed of username/timestamp/ipadress. Specifically, the string generated when attempting to login with the username acumen was YWN1bWVuLzEwOjM0OjIxLzE3OC4yMzQuMTUuNjI=.

Impact: Predictable session IDs can allow an attacker to bypass authentication and gain access to sensitive content and even escalate privileges. If default usernames are used in the system, such as root or admin, it is possible for an attacker to brute force a time stamp since the IP address can be obtained from something like an email header from the victim.

Mitigation: Session IDs must be generated at random and properly encoded using strong hashing algorithms. It is recommended that the whole session management procedure is reviewed in order to identify additional vulnerabilities.

2.3.5 - Weak password recovery procedure

Rating: Moderate

Location: www.businesssolutions.com/recovery.php

Description: The password recovery procedure consists of validating a username and answering a secret question followed by a new password. The answers to secret questions in some cases appeared to be easily researched through the use of social media, as was the case for a system administrator.

Impact: While password recovery is a necessity for all applications that require logins, these also extend an attacker’s attack surface if not implemented securely. In this case, with a little research, the answers to some of the secret questions were available through public social media. This could potentially allow an attacker to correctly answer the question, and change the password of the targeted user.
Mitigation: Two factor authentication is strongly recommended when changing account details. This ensures that the correct user will be alerted via a preregistered phone number or email, and would require their authorization to successfully carry out the attack. Avoiding questions such as “What is your pet’s name” is also necessary as these are easily determined through social engineering attacks.

3.0 - Recommendations

Due to the nature of several high and critical vulnerabilities exposed during the vulnerability assessment, it is strongly recommended that a multi-pronged approach is taken to secure the assessed system. Resources must be allocated to ensure compliance with the following suggestions is done in a timely manner.

Security policy review – A well written corporate security policy is the bedrock of a corporation’s information security strategy. This policy must be kept up to date with an organization’s changing IT structure, and should be modified as the system evolves. In this case, a thorough review is needed in order to identify missing components and update the policy to reflect the current ecosystem.

Patch management process – A Patch management process must be added to the security policy if it is non-existing. Several vulnerabilities identified were due to outdated software for which updates are readily available, indicating a lack of awareness amongst the IT team or a lack of a patch management process. Operating a patch management program as per the guidelines in NIST SP800-40 is required.


Review firewall rules – Although different firewalls were encountered at both the network and application layer, faulty configuration of these allowed them to be easily defeated. Review of the firewall rule set must be updated immediately to include encoded strings and block access to certain ports (RDP).

Review of system and firewall logs – Due to the severity of the encountered vulnerabilities, it is possible that a system breach may have occurred. Review of all firewall and system logs is highly recommended to detect such intrusions. If an intrusion is detected, an incident response must be initiated.

Internal Vulnerability Assessment – Vulnerabilities in the external part of the system leads us to believe that there may be multiple vulnerabilities behind the firewalls and on the internal system network. It is recommended that a vulnerability assessment be conducted internally as well in order to identify and mitigate vulnerabilities which may allow an attacker to continue the attack to critical systems once the outside perimeter is breached.

Regular vulnerability assessments – Regular system wide vulnerability assessments should be part of every corporate security policy in order to detect and mitigate risks in the shortest amount of time. New attacks and software vulnerabilities are discovered almost every day, and assessments by experienced security consultants coupled with a comprehensive security policy ensures that your system is always kept up to date and safe from these threats.
4.0 - Conclusion

An external vulnerability assessment was conducted on Business Solutions to examine the risk level of the system from an outsider’s perspective. Due to multiple high and critical security vulnerabilities, the system’s risk level is **critical** requiring immediate software patching and code review. Vulnerabilities were found in outdated software on the server, in the cryptographic protocols used to secure communications, and in the web application hosted on the server. If exploited, these vulnerabilities could lead to full system compromise resulting in complete integrity, confidentiality, and availability loss of the system’s information and resources.

Many vulnerabilities found require minimal skill to exploit and can even be done using automated tools, which leads us to believe this would be a very attractive target for attackers. Because of the system’s risk, it is strongly recommended that a review of both firewall and system logs be conducted to detect previous undetected intrusion attempts.

Current corporate Information Security Policies are lacking various processes such as a regular patching strategy and periodic full system vulnerability scans. It is strongly recommended that the policy be updated and these processes included. Sufficient resources must be made available in order to mitigate these vulnerabilities in the shortest time possible.
Appendix A: Risk Rating Scale

In accordance to our internal risk assessment scale which follows the guidelines set forth in NIST SP800-30, vulnerabilities are categorized using the following rating system:

Likelihood of threat event occurrence

The following definitions are used when describing the likelihood of adversarial threat occurrence:

Critical – An adversary is almost certain to exploit the vulnerability and initiate the threat event. Minimal skill is required and/or automated tools are readily available to exploit.

High – An adversary is highly likely to exploit the vulnerability and initiate the threat event. Minimal skill is required.

Medium – An adversary is somewhat likely to exploit the vulnerability and initiate the threat event. Some skill required and/or favorable circumstances required.

Low – An adversary is unlikely to exploit the vulnerability and initiate the threat event. High Level of skill and determination is required.

Likelihood threat event results in adverse impacts

The following definitions are used when describing the severity of vulnerability exploitation having adverse impacts on the system:

Critical – If a vulnerability is exploited, it is certain it will have adverse impact on the system.

High – If a vulnerability is exploited, it is highly likely to have adverse impacts on the system.

Medium – If a vulnerability is exploited, it is somewhat likely to have adverse impacts on the system.

Low – If a vulnerability is exploited, it is unlikely to have adverse impacts on the system.
Figure 1.0 Assessment Scale – Overall Likelihood

<table>
<thead>
<tr>
<th>Likelihood of Threat Event Occurrence</th>
<th>Likelihood Threat Events Result in Adverse Impacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Moderate, Medium, High, Critical</td>
</tr>
<tr>
<td>Medium</td>
<td>Low, Moderate, Moderate, High</td>
</tr>
<tr>
<td>High</td>
<td>Moderate, Medium, High, Critical</td>
</tr>
</tbody>
</table>

- **Low** – Manage by routine process
- **Moderate** – Mark as Priority
- **Critical** – Execute mitigation strategy immediately
- **High** – Immediate Attention Required
Appendix B: Definitions

**Session Cookie** – A Session cookie is used by a web page to correctly recognize users. A Session cookie is often generated for logged in users so that they are not forced to re-authenticate while navigating through the website.

**Banner Grabbing** - Banner grabbing is an enumeration technique to determine services and versions running on a server by establishing a connection with the target in order to receive a banner which may leak information.

**Buffer Overflow** – A Buffer Overflow occurs when a program that is writing data to a buffer writes past its boundaries and leaks into adjacent memory locations. If exploited correctly, this can lead to code execution on the affected system.

**Denial of Service (DoS)** – A Denial of Service attack occurs when a malicious user sends malformed requests to the target in order to tie up its resources and render the system unable to respond to legitimate traffic.

**.htaccess** - .htaccess is a file found on many Linux web servers which sets permissions and restrictions to different areas of the site. These files will over write global server configurations for the directory they are in and all subdirectories.

**Attack surface** – An attack surface is the sum of different attack vectors where unauthorized users can carry out an attack.

**Whitelist** – A whitelist in firewall configurations specifies IP addresses to be let through the firewall to interact with the system behind.

**Session Hijacking** – A session hijacking attack occurs when a flaw in the web session control mechanism is exploited in order to take over an authenticated account. This often happens when session cookies are predictable or stolen via other attacks.

**Downgrade Attack** – A downgrade attack occurs when the encryption negotiation between the client and server is forced to use an older, less favorable encryption standard or eliminate encryption altogether.

**Man In The Middle (MITM)** – A man in the middle attack occurs when the attacker is able to place himself in between the client and server communications, which allows them to view and modify all traffic between the two parties.

**Initialization Vector (IV)** – An initialization vector (IV) is an arbitrary number that can be used along with an encryption key in order to better protect the cipher text by randomizing the result in such a way that if two plain texts are encrypted using the same key, the resulting cipher texts will be different and therefore leak no information to the attacker.
Cipher block chaining (CBC) – Cipher block chaining is a mode of operation for a block cipher in which a sequenced set of bits are encrypted as a block with the cipher key being applied to the entire block.

Chosen Plaintext Attack (CPA) – A chosen plaintext attack is an attack model which presumes the attacker can obtain the cipher texts for arbitrary plaintexts. The goal of the attack is to gain information which reduces the security of the encryption scheme.

Browser Exploit Against SSL/TLS (BEAST) - BEAST is an attack on cryptographic communications using TLS 1.0 and SSL where the initialization vector can be predicted by an active MITM attacker. This vulnerability enables the attacker to recover small fragments of plaintext such as credentials or session cookies.

Internet Message Access Protocol – IMAP allows a two way synchronization between an email client and a mail account.

Timing attacks – A timing attack is a side channel attack in which the attacker may send many requests to a server and study the response times in order to gain information about the cryptographic algorithms.

SQL injection – SQL injection is a code injection technique used to attack SQL database applications with malicious SQL queries with the intent of compromising the database and/or system.

Stored Cross Site Scripting (XSS) – Cross Site Scripting occurs when an attacker is able to inject client-side script into web pages. In the Stored or Persistent case, the attacker is able to inject the script and have it saved by the server. This way, the server will supply the malicious script every time a user visits the vulnerable page.