Executive Summary

for

Cybersecurity Risk Assessment

Prepared for: Client’s Consulting Website

April 17, 2021

Prepared by: Paul M. Patterson and Elva R. Lopez
Caltech Cyber Bootcamp

Final Project

Table of Contents

RISK ASSESSMENT

Executive Summary

Our client is a consultant for entertainment industry businesses, associations and nonprofit organizations. Under that banner, she produces promotional videos for studios alongside production companies and works with niche entertainment industry businesses and non-profit organizations developing strategic growth initiatives within very targeted markets.

Her marketing skills were honed at a top entertainment company where she gained experience in everything from developing feature editions to branding, trade shows and special events. Her event production experience began while working at a leading movie studio where she both managed and stage managed various live shows and special events.

Our client contracted with us to perform a risk assessment of her company's website and this write up explains all the details regarding the risk assessment.

STEP 1 – Assessment of Assets

Website

Our client created her consulting business website utilizing iWeb, a template-based WYSIWYG (What You See is What You Get) website creation tool developed by Apple Inc. iWeb has since been discontinued due to end of life after the last update in 2010. Our client's website is currently being hosted by Bluehost.com. WordPress version 5.7 is the application programming interface now being used. The site runs on port 80, unsecured HTTP. The source code for the site can be obtained via FTP server. The site has many external links that point to external resources no longer in existence.

The site uses JavaScript, which has not been updated, and PHP 7.3.27 as the programming languages, MySQL 5.6.41 open-source relational database management system, and HTTPD Apache Version 2.4.46 open-source cross-platform web server software.

There's no data collection from the site and our client has not updated the site in the recent past. Our client plans to take down the site and create a new website following the risk assessment. She plans to utilize the learnings from this risk assessment to create a new website with cyber security as a key objective.

STEP 2 – Active and Passive Reconnaissance

This risk assessment required both active and passive reconnaissance to find as many website vulnerabilities as possible. We began by requesting permission from the website host provider, Bluehost.com, to conduct a penetration test on the website. At first, the technician agreed to the penetration test; however, after asking additional questions, we were only given permission to access the website via FTP server with login credentials. The caveat is that we were not given permission to escalate privileges or gain root access on the website despite having our client's permission to do so.

To begin scanning our client's website for potential vulnerabilities using the assigned FTP server, the following steps were taken - with results of each step given:

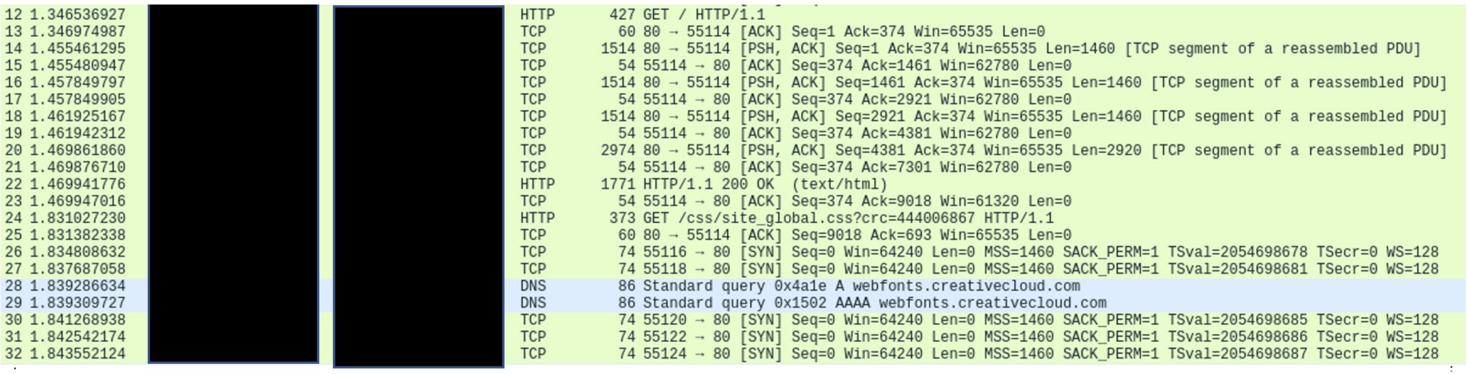
FTP, Admin Authentication

1) Log into FTP site
2) Execute pwd command to determine the present folder/directory in the file structure
3) Execute ls command to determine what is currently in the folder
4) A significant presence of wordpress .php files was noted; this led to a google search for php files that might contain useful information for site penetration.
5) Conduct recon of site source files; focus recon on wp-admin, wp-config.php, wp-includes, wp-login.php, as they appear on first glance to be files of interest. Google search indicated wp-config.php contains login username, a hashed/salted password, the salt for the login, and authorization keys. The backdoor to logging in to the site in a production environment has been established.

   *Note: It is recommended to use Secure File Transfer Protocol (also known as SSH File Transfer Protocol) as FTP on Port 21 was not designed with any deliberate security considerations.

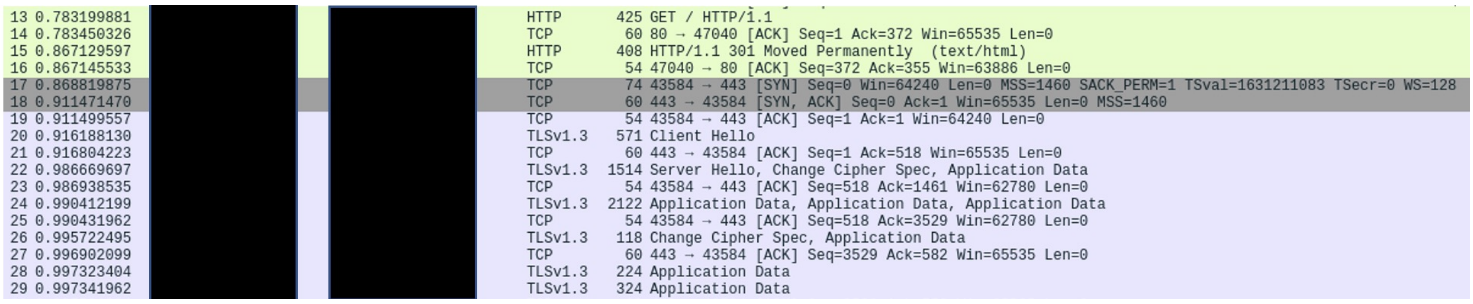
Wireshark Scans

1) Run wireshark while navigating to the target website, and to linked websites that (it was later found out) are being hosted by the target site.
2) Review pcaps from navigation to target website; nothing out of the ordinary was found. The site is running on HTTP, however we later found out that the site has HTTPS capability, but was not being redirected.
3) Review pcaps from navigation to linked websites; TLS 1.2 was noted as the server/client handshake protocol version, with 1.3 being the latest version. Despite known vulnerabilities against TLS 1.2, namely ability to execute a man-in-the-middle attack, very few sites run TLS 1.3 and it is still acceptable for handshake packets to run on TLS 1.2.
4) It was noted that sites hosted by the target site do not contain packets with encrypted Application Data. The data from the sites are pulled directly from source code files. The sites not hosted by the target contain the encrypted Application Data files - reference packets 24, 28, and 29 in Figure 2 below. The owner of the target site confirmed that other sites are being hosted by the target.



```
12 1.346536927 [redacted] [redacted] HTTP       427 GET / HTTP/1.1
13 1.346974987                      TCP         60 80 → 55114 [ACK] Seq=1 Ack=374 Win=65535 Len=0
14 1.455461295                      TCP       1514 80 → 55114 [PSH, ACK] Seq=1 Ack=374 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
15 1.455480947                      TCP         54 55114 → 80 [ACK] Seq=374 Ack=1461 Win=62780 Len=0
16 1.457849797                      TCP       1514 80 → 55114 [PSH, ACK] Seq=1461 Ack=374 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
17 1.457849905                      TCP         54 55114 → 80 [ACK] Seq=374 Ack=2921 Win=62780 Len=0
18 1.461925167                      TCP       1514 80 → 55114 [PSH, ACK] Seq=2921 Ack=374 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
19 1.461942312                      TCP         54 55114 → 80 [ACK] Seq=374 Ack=4381 Win=62780 Len=0
20 1.469861860                      TCP       2974 80 → 55114 [PSH, ACK] Seq=4381 Ack=374 Win=65535 Len=2920 [TCP segment of a reassembled PDU]
21 1.469876710                      TCP         54 55114 → 80 [ACK] Seq=374 Ack=7301 Win=62780 Len=0
22 1.469941776                      HTTP      1771 HTTP/1.1 200 OK  (text/html)
23 1.469947016                      TCP         54 55114 → 80 [ACK] Seq=374 Ack=9018 Win=61320 Len=0
24 1.831027230                      HTTP       373 GET /css/site_global.css?crc=444006867 HTTP/1.1
25 1.831382338                      TCP         60 80 → 55114 [ACK] Seq=9018 Ack=693 Win=65535 Len=0
26 1.834808632                      TCP         74 55116 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2054698678 TSecr=0 WS=128
27 1.837687058                      TCP         74 55118 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2054698681 TSecr=0 WS=128
28 1.839286634                      DNS         86 Standard query 0x4a1e A webfonts.creativecloud.com
29 1.839309727                      DNS         86 Standard query 0x1502 AAAA webfonts.creativecloud.com
30 1.841268938                      TCP         74 55120 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2054698685 TSecr=0 WS=128
31 1.842542174                      TCP         74 55122 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2054698686 TSecr=0 WS=128
32 1.843552124                      TCP         74 55124 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2054698687 TSecr=0 WS=128
```

Figure 1: Snapshot of pcap from navigation to linked/hosted site from target site



```
13 0.783199881 [redacted] [redacted] HTTP       425 GET / HTTP/1.1
14 0.783450326                      TCP         60 80 → 47040 [ACK] Seq=1 Ack=372 Win=65535 Len=0
15 0.867129597                      HTTP       408 HTTP/1.1 301 Moved Permanently  (text/html)
16 0.867145533                      TCP         54 47040 → 80 [ACK] Seq=372 Ack=355 Win=63886 Len=0
17 0.868819875                      TCP         74 43584 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1631211083 TSecr=0 WS=128
18 0.911471470                      TCP         60 443 → 43584 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
19 0.911499557                      TCP         54 43584 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
20 0.916188130                      TLSv1.3    571 Client Hello
21 0.916804223                      TCP         60 443 → 43584 [ACK] Seq=1 Ack=518 Win=65535 Len=0
22 0.986669697                      TLSv1.3   1514 Server Hello, Change Cipher Spec, Application Data
23 0.986938535                      TCP         54 43584 → 443 [ACK] Seq=518 Ack=1461 Win=62780 Len=0
24 0.990412199                      TLSv1.3   2122 Application Data, Application Data, Application Data
25 0.990431962                      TCP         54 43584 → 443 [ACK] Seq=518 Ack=3529 Win=62780 Len=0
26 0.995722495                      TLSv1.3    118 Change Cipher Spec, Application Data
27 0.996902099                      TCP         60 443 → 43584 [ACK] Seq=3529 Ack=582 Win=65535 Len=0
28 0.997323404                      TLSv1.3    224 Application Data
29 0.997341962                      TLSv1.3    324 Application Data
```

Figure 2: Snapshot of pcap from navigation to linked/non-hosted site from target site

Website Inspection

1) The website was run in Mozilla Firefox and inspected as links were clicked in order to look for possibly vulnerable services or files running the site.

Aside from the information gathered above, nothing material was determined by doing inspection of all the client side web links.

This became a purple team exercise and in addition to the FTP scans, we passively researched the website itself. We found several potential vulnerabilities on the site. One of the vulnerabilities we identified without scanning the site is that it runs on port 80, which is unsecured HTTP using WordPress. The website is being hosted by Bluehost.com on a shared server with thousands of other accounts. This is the main reason why scanning the website was not permitted. We used ⌘ Cmd + ⌥ Option + I to inspect the site. We went in search of potential vulnerabilities using this passive reconnaissance method. We also found a static code analysis tool called JSLint (Javascript Lint) to scan the Javascript code; however, the results were less than optimal because Javascript on the website has not been updated for a long time.

We also reviewed our client's information on Netcraft.com to check the site's Netcraft Risk Rating. Fortunately, the rating is 0/10, and a lower risk rating indicates a lower risk. We scanned our client's website on Whois and found personal information such as name, address, phone number in the clear. In addition, we noticed that the Domain Name System Security Extensions (DNSSEC) was unsigned. This signature adds a layer of security to the Domain Name System (DNS) and not having it signed could pose a potential vulnerability.

Following the passive reconnaissance, we needed a way to confirm if the vulnerabilities we had found were in fact vulnerabilities on this particular website. We contacted the web host by utilizing our client's account, with our client's permission, and requested the versions of WordPress, PHP, Javascript, MySQL, HTTPD, and FTP. We were given all of the current versions and a link http://www.*websitename*.com/info.php which has a plethora of information that should not be easily accessed by simply typing it into the browser.

Website Security Vulnerability Scans

In addition, we found a site with a number of free open-source resources to scan websites online for security vulnerabilities: https://geekflare.com/online-scan-website-security-vulnerabilities/. We utilized several of these third-party sites to conduct vulnerability scans of the website on our behalf. The sites include Quttera.com, UpGuard.com, Observatory.Mozilla.org and ImmuniWeb.org. We collected information about Vulnerabilities, Impact, Risk, Recommended Controls, Common Vulnerability Scoring System (CVSS) Score, CVSS Vector, and CIA (Confidentiality, Integrity, Availability). The third-party vulnerability scans yielded twice as many vulnerabilities as we had initially considered. The CVSS scores convey vulnerability severity and help determine urgency and priority of response. For the purpose of this risk assessment, we will be focusing attention on the vulnerabilities with CVSS scores between 7 and 10 as being those with a High or Critical Risk Rating. Eleven of the 24 vulnerabilities found in this research will be discussed in detail in Step 3 – Website Vulnerabilities below. Along with the vulnerabilities are the recommended controls to mitigate these vulnerabilities. Additional vulnerabilities will be included in Step 4 - Risk Assessment Results and these have CVSS scores lower than 6, which implies medium to low risk.

STEP 3 – Website Vulnerabilities

Below are the top 11 vulnerabilities found on our client's website with CVSS scores between 7 and 10:

1) Improper Access Control to /admin/users/ (CVSS v3.0 Score 9.8 – Critical)

- Threat-Source/Vulnerability: A remote attacker can visit the website and create an administrative user account without having to login.
- Recommended Controls: Implement User Access Control mechanism to verify a user's identity and access permissions before they are allowed to access any sensitive data.
- Confidentiality Impact: High – All information is disclosed to the attacker; or some critical information is disclosed.
- Integrity Impact: High – The attacker can modify any non-critical information; or some critical information.
- Availability Impact: High – The affected resource is completely unavailable; or is critical and suffers reduced performance or interruption

2) Arbitrary File Upload in /uploadify/uploadify.php (CVSS v3.0 Score 10 – Critical)

- Threat-Source/Vulnerability: A remote attacker can upload and execute files with .php extension on the web server without having to login. The attacker would be able to upload a web shell to the server, execute arbitrary PHP code, connect to the databases and read information from them, read files and execute commands with privileges.

- Recommended Controls: Edit the web application source code and implement proper validation of the uploaded files based on extensions and MIME type, and restrict access to the vulnerable script as a temporary solution.
- Confidentiality Impact: High – All information is disclosed to the attacker; or some critical information is disclosed.
- Integrity Impact: High – The attacker can modify any non-critical information; or some critical information.
- Availability Impact: High – The affected resource is completely unavailable; or is critical and suffers reduced performance or interruption

3) **XML-RPC Response Parser (CVSS v3.0 Score 9.8 – Critical)**
   - Threat-Source/Vulnerability: An attacker could use an XML-RPC server to target a XML-RPC client and cause it to execute arbitrary code.
   - Recommended Controls: Do not use Apache XML-RPC (aka ws-xmlrpc) and upgrade XML version.
   - Confidentiality Impact: High – All information is disclosed to the attacker; or some critical information is disclosed.
   - Integrity Impact: High – The attacker can modify any non-critical information; or some critical information.
   - Availability Impact: High – The affected resource is completely unavailable; or is critical and suffers reduced performance or interruption

   -

4) **Apache HTTP Server (CVSS v3.0 Score 9.8 – Critical)**
   - Threat-Source/Vulnerability: The Shellshock Vulnerability allows privilege escalation attacks where a non-administrator Unix user could run commands as root.
   - Recommended Control: The original fix for this issue was incorrect; CVE-2014-7169 has been assigned to cover the vulnerability that is still present after the incorrect fix.
   - Confidentiality Impact: High – The attacker has full access to all resources in the impacted system, including highly sensitive information such as encryption keys.
   - Integrity Impact: High – The attacker can modify any/all information on the target system, resulting in a complete loss of integrity.
   - Availability Impact: High – There is a complete loss of availability of the impacted system of information.

5) **SQL Injection in /index.php (CVSS v3.0 Score 8.1 - High)**
   - Threat-Source/Vulnerability: An attacker can potentially use an SQL Injection in the script when the user-supplied input is passed via the HTTP GET parameter without sanitation.
   - Recommended Controls: Properly filter all user-supplied input being processed by the application by developing, testing and deploying corrections for the application's source code.
   - Confidentiality Impact: High – All information is disclosed to the attacker; or some critical information is disclosed.

- Integrity Impact: High – The attacker can modify any non-critical information; or some critical information.
- Availability Impact: High – The affected resource is completely unavailable; or is critical and suffers reduced performance or interruption

6) **Wordpress Database Passwords and Salts in Plain Text (CVSS v2.0 Score 7 – High)**
   - Threat-Source/Vulnerability: An attacker can gain unauthorized access to hosted WordPress website by using login credentials that are in plain text in the wp-config.php file.
   - Recommended Controls: Place the wp-config.php file in a folder that is not accessible via the website.
   - Confidentiality Impact: Complete – There is total information disclosure, resulting in all system files being revealed. The attacker is able to read all of the systems' data (memory, files, etc.).
   - Integrity Impact: Complete – There is a total compromise of system integrity. The entire system is compromised. The attacker is able to modify files in the target system.
   - Availability Impact: None – There is no impact to the availability of the system.

7) **Wordpress authentication keys in Plain Text (CVSS v2.0 Score 7 – High)**
   - Threat-Source/Vulnerability: An attacker can gain unauthorized access to hosted WordPress website by using authentication keys that are in plain text in the wp-config.php file.
   - Recommended Controls: Place the wp-config.php file in a folder that is not accessible via the website.
   - Confidentiality Impact: Complete – There is total information disclosure, resulting in all system files being revealed. The attacker is able to read all of the systems' data (memory, files, etc.).
   - Integrity Impact: Complete – There is a total compromise of system integrity. The entire system is compromised. The attacker is able to modify files in the target system.
   - Availability Impact: None – There is no impact to the availability of the system.

8) **Javascript Node.js Application (CVSS v3.0 Score 7.5 – High)**
   - Threat-Source/Vulnerability: The Node.js application could allow an attacker to trigger a DNS request for a host of their choice to trigger a Denial of Service by getting the application to resolve a DNS record with a large number of responses.
   - Recommended Controls: Upgrade to Node.js version 12.19.1 or higher.
   - Confidentiality Impact: None – No data is accessible to unauthorized users as a result of the exploit.
   - Integrity Impact: None – There is no loss of the integrity of any information.
   - Availability Impact: High – There is a complete loss of availability of the impacted system or information.

9) **HTTP Request Smuggling (CVSS v3.0 Score 7.5 – High)**
   - Threat-Source/Vulnerability: An attacker could leverage specific features of the HTTP/1.1 protocol in order to bypass security protections, conduct phishing attacks, as well as obtain sensitive information from requests other than their own.
   - Recommended Controls: remediation of this vulnerability can be tricky depending on whether you are a frontend or backend project maintainer. To simplify, all remediation points are covered

along with reasoning to which smuggling attack type this will remediate against. In an ideal scenario, all of the points mentioned below should be used to provide a Defense-in-Depth approach solution.

- Confidentiality Impact: None – No data is accessible to unauthorized users as a result of the exploit.
- Integrity Impact: High – The attacker can modify any/all information on the target system, resulting in a complete loss of integrity.
- Availability Impact: None – There is no loss of availability

10) Web Server HTTP 1.1 Header Remote Overflow (CVSS v2.0 Score 7.5 – High)

- Threat-Source/Vulnerability: Arbitrary code may be run on a remote server and an exploit could crash the server.
- Recommended Control: Upgrade the web server or protect it with a filtering reverse proxy.
- Confidentiality Impact: Partial – There is considerable informational disclosure. Access to some files is possible, but the attacker does not have control over what is obtained, or the scope of the loss is constrained.
- Integrity Impact: Partial – Modification of some system files or information is possible, but the attacker does not have control over what can be modified or the scope of what the attacker can affect is limited.
- Availability Impact: Partial – Reduced performance or interruptions in resource availability.

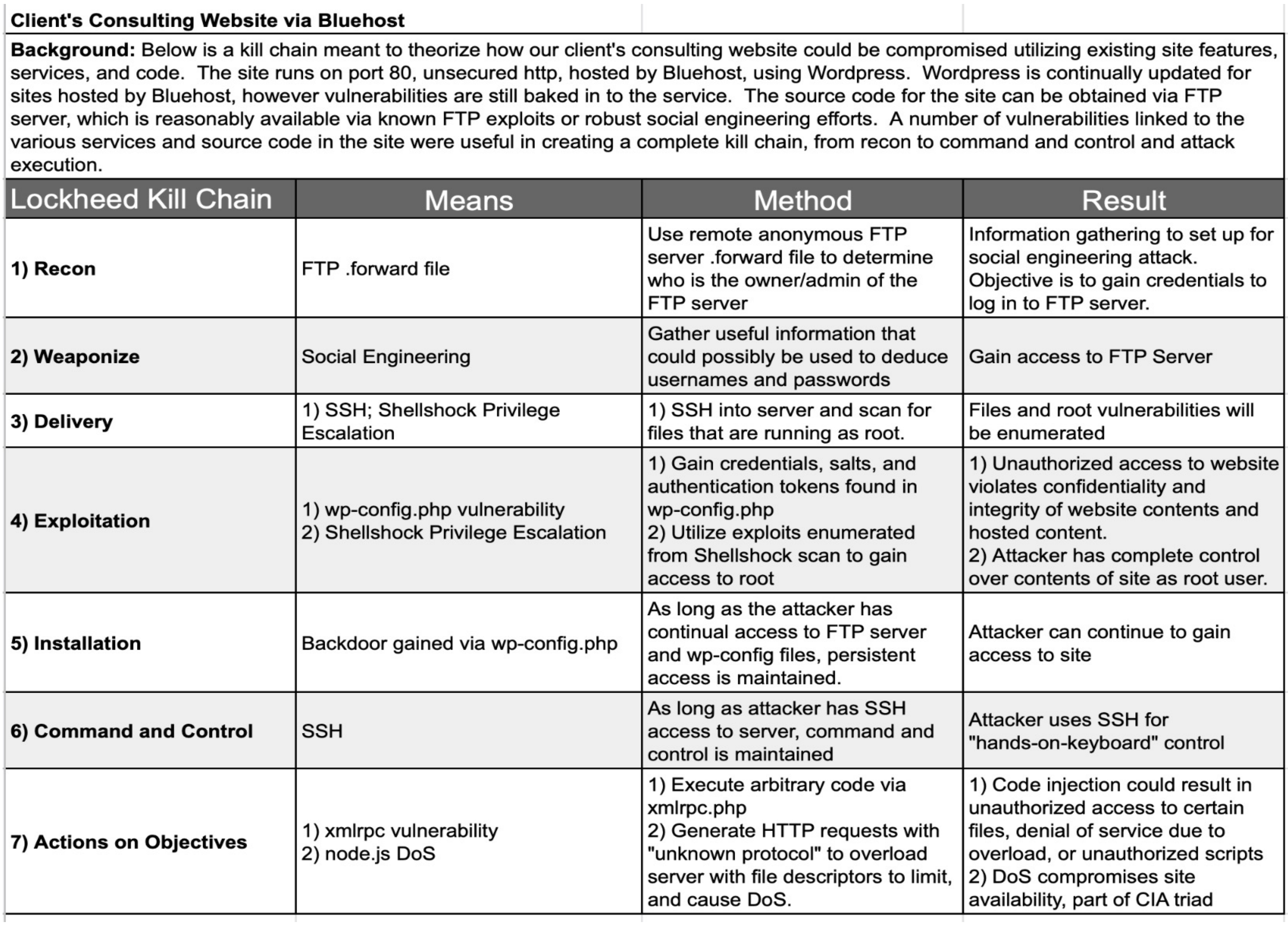
11) PHP Wp-trackback.php (CVSS v3.0 Score 7.5 – High)

- Threat-Source/Vulnerability: The remote web server contains a PHP application affected by SQL injection attacks.
- Recommended Control: Unknown at this time.
- Confidentiality Impact: High – The attacker has full access to all resources in the impacted system, including highly sensitive information such as encryption keys.
- Integrity Impact: High – The attacker can modify any/all information on the target system, resulting in a complete loss of integrity.
- Availability Impact: None – There is no loss of availability.

(CVSS Score Sources: https://www.balbix.com/insights/base-cvss-scores/; https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator)

STEP 4- Risk Assessment Results Spreadsheet
(Refer to Project_Risk_Assessment_Results_04142021.xlsx)

STEP 5- Lockheed Kill Chain



Client's Consulting Website via Bluehost

Background: Below is a kill chain meant to theorize how our client's consulting website could be compromised utilizing existing site features, services, and code. The site runs on port 80, unsecured http, hosted by Bluehost, using Wordpress. Wordpress is continually updated for sites hosted by Bluehost, however vulnerabilities are still baked in to the service. The source code for the site can be obtained via FTP server, which is reasonably available via known FTP exploits or robust social engineering efforts. A number of vulnerabilities linked to the various services and source code in the site were useful in creating a complete kill chain, from recon to command and control and attack execution.

Lockheed Kill Chain	Means	Method	Result
1) Recon	FTP .forward file	Use remote anonymous FTP server .forward file to determine who is the owner/admin of the FTP server	Information gathering to set up for social engineering attack. Objective is to gain credentials to log in to FTP server.
2) Weaponize	Social Engineering	Gather useful information that could possibly be used to deduce usernames and passwords	Gain access to FTP Server
3) Delivery	1) SSH; Shellshock Privilege Escalation	1) SSH into server and scan for files that are running as root.	Files and root vulnerabilities will be enumerated
4) Exploitation	1) wp-config.php vulnerability 2) Shellshock Privilege Escalation	1) Gain credentials, salts, and authentication tokens found in wp-config.php 2) Utilize exploits enumerated from Shellshock scan to gain access to root	1) Unauthorized access to website violates confidentiality and integrity of website contents and hosted content. 2) Attacker has complete control over contents of site as root user.
5) Installation	Backdoor gained via wp-config.php	As long as the attacker has continual access to FTP server and wp-config files, persistent access is maintained.	Attacker can continue to gain access to site
6) Command and Control	SSH	As long as attacker has SSH access to server, command and control is maintained	Attacker uses SSH for "hands-on-keyboard" control
7) Actions on Objectives	1) xmlrpc vulnerability 2) node.js DoS	1) Execute arbitrary code via xmlrpc.php 2) Generate HTTP requests with "unknown protocol" to overload server with file descriptors to limit, and cause DoS.	1) Code injection could result in unauthorized access to certain files, denial of service due to overload, or unauthorized scripts 2) DoS compromises site availability, part of CIA triad

Conclusion

The risk assessment serves as a guide and indicator for improvements regarding cybersecurity for our client’s consulting website. The assessment can help our client with the implementation of a formal cybersecurity plan to improve her security posture as she creates a new website for her business.

The list of vulnerabilities found in this assessment are not exhaustive; therefore, new vulnerabilities are likely to exist and develop over time. Implementing and maintaining security measures is highly recommended and will be an ongoing process.

We recommend to our client that she establish cybersecurity as one of her strategic imperatives to address the risks and controls which we have identified through our research, with a focus on controls for critical and high risk vulnerabilities.