Web Application Penetration Test and Vulnerability Assessment, and External Network Penetration Test

Prepared by:
Prepared for:
Submission Date: 02/23/2017

Table of Contents:

Executive Summary........3
Assessment Objectives........3
Security Posture Analysis........4
Positive Observations........4
Areas of Risk........4
Recommendations........4
Risk Analysis........5
DREAD Scoring Criteria........5
Composite Risk Categories........5
Remediation Effort Key........6
Security Risk Findings........6
Moderate Risk Findings and Remediation Guidance........7
Weak Web Transport Layer Security........7
Missing Secure Attribute in SSL Session Cookie........11
Samba Server Allows NULL Session Enumeration........15
TCP Timestamps Enabled........18
Low Risk Findings and Remediation Guidance........21
Outdated Version of Nginx........21
Appendix A: Rapid7 Contacts........23
Appendix B: Engagement Scope Overview........24
Rules of Engagement and Assumptions........24
Accounts........24
Scope Targets........24
Appendix C: DREAD Overview........25
Appendix D: Manual Application Testing and OWASP Testing Methodology........26

Executive Summary

Rapid7 Global Services conducted a web application vulnerability assessment and penetration test and external network penetration test of security posture during the week of February 20th, 2017. This test was designed to provide an independent, point-in-time, assessment of XXX's organizational vulnerabilities.

Rapid7 worked with □□ to structure a penetration test that can identify vulnerabilities associated with the 'click2park.co.uk/ □' application and hosting infrastructure. □ provided Rapid7 with authenticators for both a privileged and non-privileged user. Rapid7 discovered weak Transport Layer Security (TLS) in use on the network, and Secure Sockets Layer (SSL) cookies missing the 'Secure' attribute. Rapid7 also found a Samba server that allowed user enumeration through NULL sessions, and TCP timestamps enabled on packet headers.

Based on the testing performed by Rapid7, the overall risk to the click2park application is 'Moderate'. will need to follow the recommendations in this report to improve the application's security posture.

Assessment Objectives

- Document and demonstrate likely attack vectors.
- Quantify the impact of successful attacks through active exploitation.
- Identify specific vulnerabilities that can be remediated to improve security.
- Recommend ways to improve 's overall security posture.

Risk Summary

The following chart provides a summary of □'s risk ratings:

CRITICAL	SEVERE	MODERATE	LOW
0	0	4	1

Table 1: Risk Summary

's overall risk rating is: **MODERATE**

Risk ratings are based on the vulnerabilities and technical risks observed during this assessment, including:

- The ease with which attacks can be executed.
- The impact of the executed attacks on information security.
- The organization's ability to detect and react to executed attacks.
- A comparison of 's security posture against other organizations of similar size.

Security Posture Analysis

Rapid7 identified positive observations and areas of risk related to ’s overall security posture. Each risk discovered during the assessment will have to be remediated based on recommendations in the Risk Findings and Remediation Guidance sections of this report.

Positive Observations

- XXX exposed a limited attack surface to the Internet.
- The application used anti-Cross-Site Request Forgery (CSRF) tokens, which prevented Rapid7 from executing CSRF attacks.
- The application encoded input and output, preventing Rapid7 from breaking out of the user context and executing scripts within the client browser.

Areas of Risk

- Rapid7 found weak ciphers in use on the network, which a malicious actor could use to decrypt network traffic.
- The application did not set the ‘Secure’ flag on server response headers, which allows information to be transmitted in cleartext. A malicious actor could use this to intercept sensitive session information passed over HTTP.
- The application’s Samba server allowed NULL sessions, which exposed sensitive click2park infrastructure information.
- TCP Timestamps were enabled on the application, which a malicious actor could use to determine server uptime and which patches are applied to the server.
- The application’s Nginx server was not running the most up to date version. The version, 1.10.0, could allow a malicious actor to execute Denial of Service (DoS) or privilege escalation attacks.

Recommendations

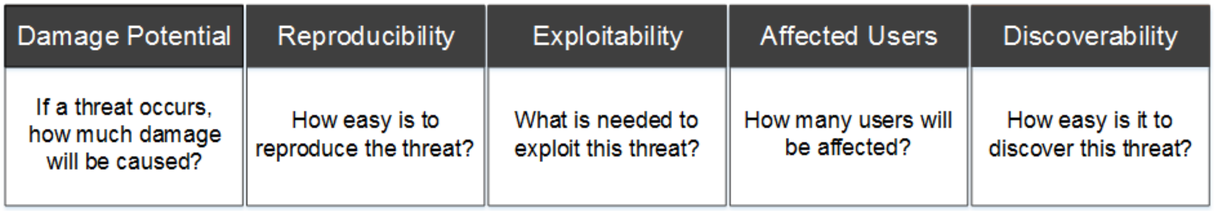
- Configure the application’s web and email servers with strong encryption ciphers, and use certificates with strong hashing algorithms that are signed by a trusted, 3rd party Certificate Authority.
- Set the ‘Secure’ flag with the ‘sessionid’ cookie value in the application’s response headers.
- Configure the Samba server to disallow NULL session.
- Disable TCP Timestamps on the click2park web server.
- Update the Nginx server to the latest available version.

The general security of XXX’s security posture will improve once these recommendations are implemented.

Risk Analysis

Each area of risk is analyzed using the DREAD framework. This framework is adaptive, allowing these risk findings to be rated based on the context of the affected environment. For example, a vulnerability that affects a non-critical system located in a heavily protected subnet has a lower risk score than a critical system affected by the same issue. The following charts describe how the DREAD framework is applied when calculating technical risk as well as the remediation efforts associated with each finding. See Appendix C: DREAD Overview for a detailed explanation of the framework.

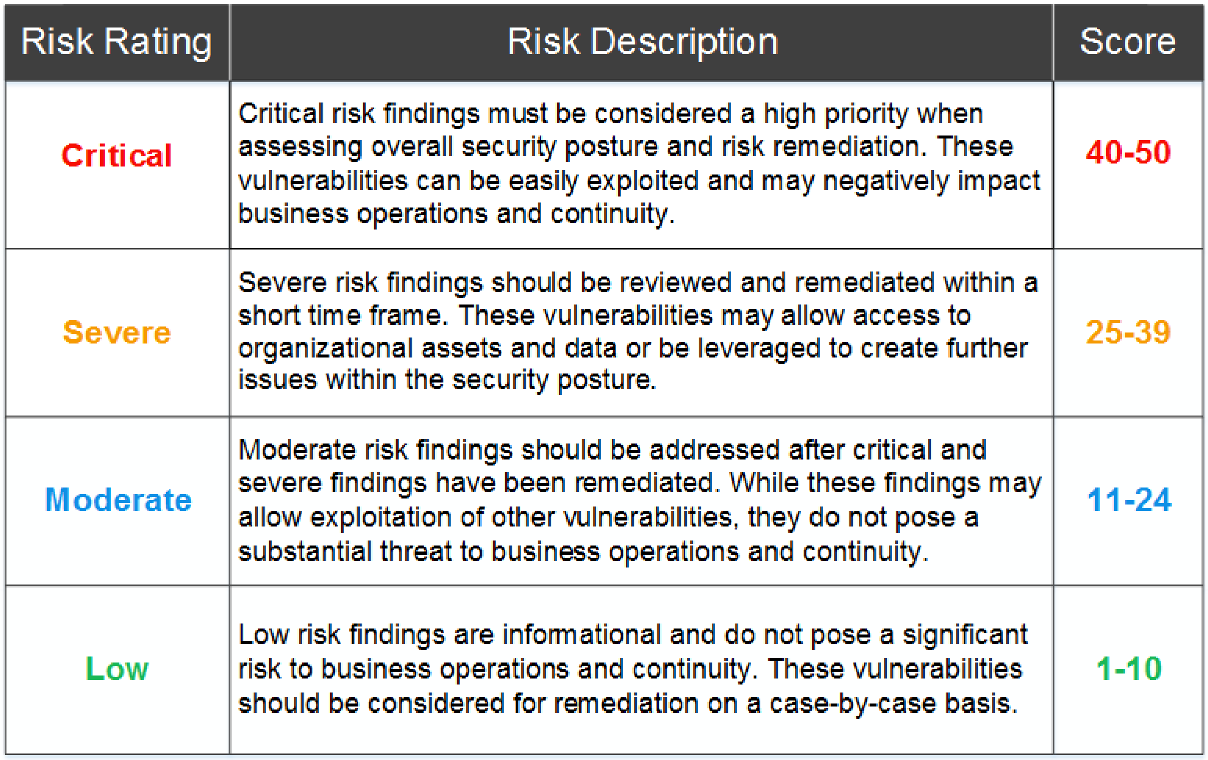
DREAD Scoring Criteria



Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability
If a threat occurs, how much damage will be caused?	How easy is to reproduce the threat?	What is needed to exploit this threat?	How many users will be affected?	How easy is it to discover this threat?

Table 2: DREAD Scoring Criteria

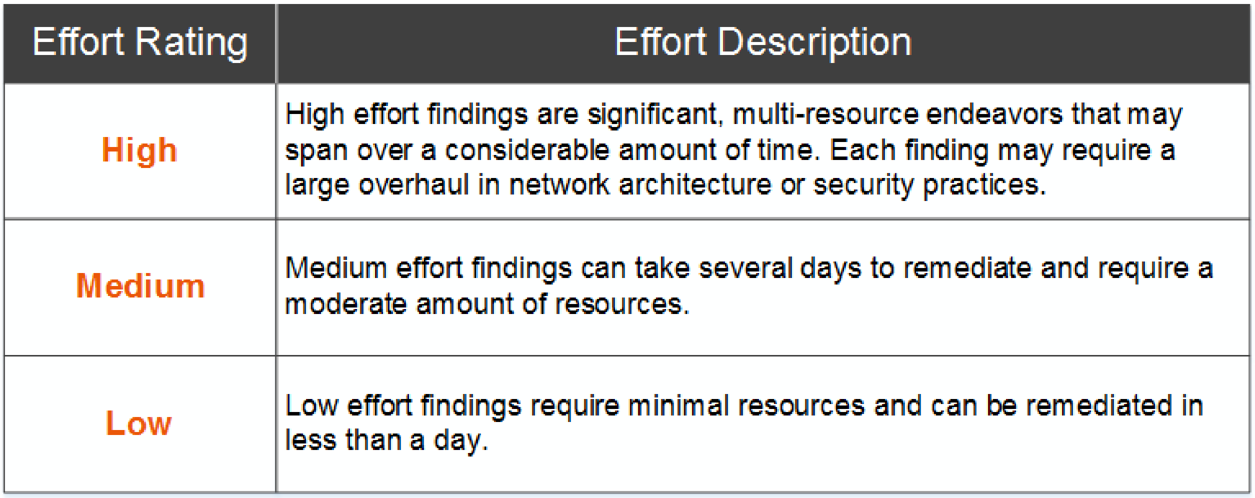
Composite Risk Categories



Risk Rating	Risk Description	Score
Critical	Critical risk findings must be considered a high priority when assessing overall security posture and risk remediation. These vulnerabilities can be easily exploited and may negatively impact business operations and continuity.	40-50
Severe	Severe risk findings should be reviewed and remediated within a short time frame. These vulnerabilities may allow access to organizational assets and data or be leveraged to create further issues within the security posture.	25-39
Moderate	Moderate risk findings should be addressed after critical and severe findings have been remediated. While these findings may allow exploitation of other vulnerabilities, they do not pose a substantial threat to business operations and continuity.	11-24
Low	Low risk findings are informational and do not pose a significant risk to business operations and continuity. These vulnerabilities should be considered for remediation on a case-by-case basis.	1-10

Table 3: Risk Categories

Remediation Effort Key



Effort Rating	Effort Description
High	High effort findings are significant, multi-resource endeavors that may span over a considerable amount of time. Each finding may require a large overhaul in network architecture or security practices.
Medium	Medium effort findings can take several days to remediate and require a moderate amount of resources.
Low	Low effort findings require minimal resources and can be remediated in less than a day.

Table 4: Remediation Effort

Security Risk Findings

Security Risk Finding	Risk Score	Effort
CRITICAL	**(40-50)**	
SEVERE	**(25-39)**	
MODERATE	**(11-24)**	
Weak Web Transport Layer Security	21	LOW
Missing Secure Attribute in SSL Session Cookie	21	LOW
Samba Server Allows NULL Session Enumeration	14	LOW
TCP Timestamps Enabled	13	LOW
LOW	**(1-10)**	
Outdated Software Versions	8	LOW

Table 5: Risk Summary

Moderate Risk Findings and Remediation Guidance

Weak Web Transport Layer Security

DREAD Score Summary

Risk Rating: MODERATE						
Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total	Remediation Effort
4	3	3	4	7	21	LOW

Status

CONFIRMED

Summary of Finding

Rapid7 identified web services utilizing weak encryption standards for the security of traffic transmitted from the web service to the end-user. Transport Layer Security (TLS) and its predecessor Secure Socket Layer (SSL) are both cryptographic protocols used to provide encryption for information communicated over the Internet. In addition to the cryptographic protocols, these connections also utilize an array of cipher suites. Cipher suites are the cryptographic algorithms used to encrypt the communicated traffic.

In both cases there have been protocol versions and cipher suites deemed cryptographically insecure. Most notable are the following:

- SSL Version 2
- SSL Version 3
- Any cipher suite utilizing antiquated algorithms such as RC4
- Any cipher suite utilizing less than 128 bits of encryption
- Any NULL or EXPORT ciphers

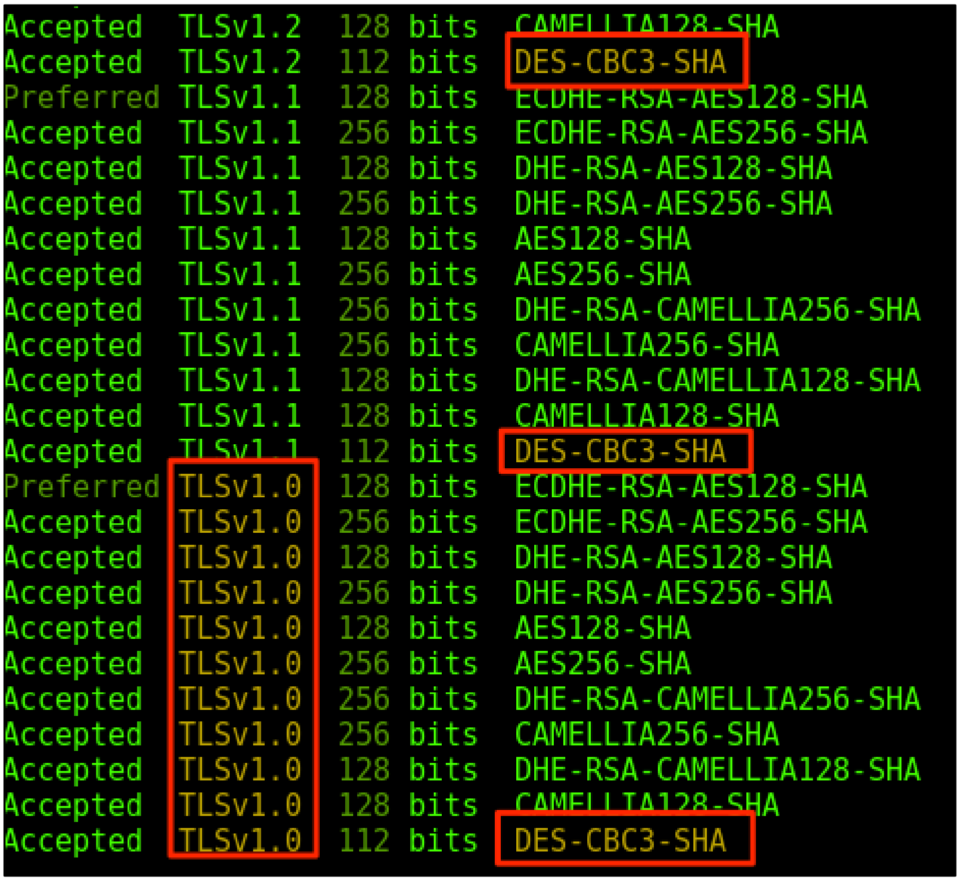
Although the web services are configured to accept stronger protocols and cipher suites, a malicious actor could leverage a web service still supporting the above weak protocols and cipher suites to potentially downgrade a user's connection to the weak protocol or cipher suite during a Man-in-the-Middle attack. This downgrade would greatly increase the malicious actor's chances of decrypting the secure web communication. The most common of these attacks can occur if a user is utilizing a public Wi-Fi connection being controlled by a malicious actor.

In addition to being cryptographically insecure, many of the antiquated protocols and cipher suites also suffer from Denial of Service (DoS) vulnerabilities. A malicious actor could leverage vulnerabilities to waste XXX resources or cause a disruption to services.

It should be noted this risk of weak protocols and cipher suites is increased when the data being communicated over the web connection contains sensitive information such as a user's password, credit card number, bank account number, or other personally identifiable information.

Proof of Concept

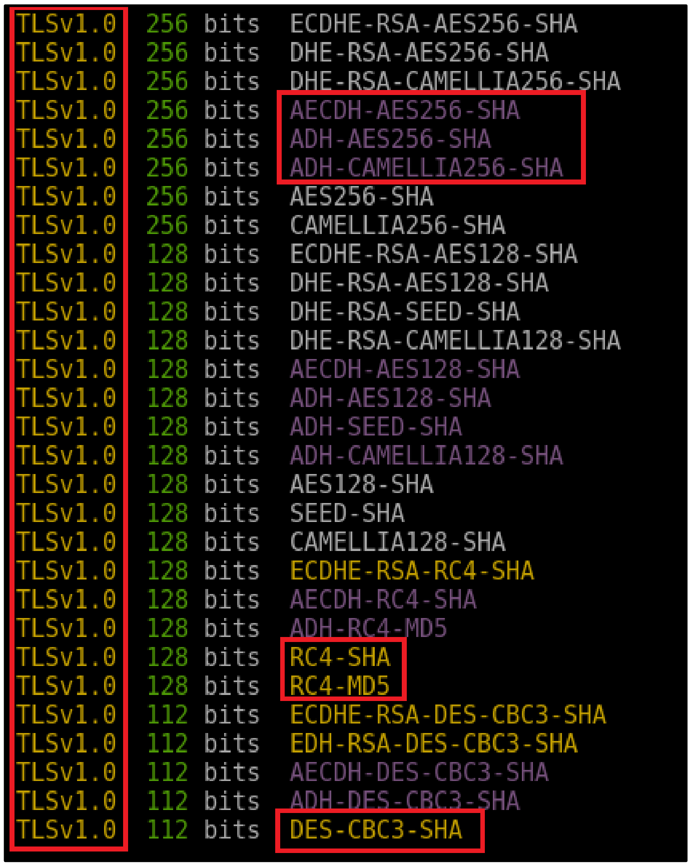
Rapid7 used SSLScan to audit the encryption protocols and ciphers in use on the network, and found weak 3DES ciphers and the TLS 1.0 protocol, as shown in Figure 1:



```
Accepted  TLSv1.2  128 bits  CAMELLIA128-SHA
Accepted  TLSv1.2  112 bits  DES-CBC3-SHA
Preferred TLSv1.1  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.1  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.1  128 bits  DHE-RSA-AES128-SHA
Accepted  TLSv1.1  256 bits  DHE-RSA-AES256-SHA
Accepted  TLSv1.1  128 bits  AES128-SHA
Accepted  TLSv1.1  256 bits  AES256-SHA
Accepted  TLSv1.1  256 bits  DHE-RSA-CAMELLIA256-SHA
Accepted  TLSv1.1  256 bits  CAMELLIA256-SHA
Accepted  TLSv1.1  128 bits  DHE-RSA-CAMELLIA128-SHA
Accepted  TLSv1.1  128 bits  CAMELLIA128-SHA
Accepted  TLSv1.1  112 bits  DES-CBC3-SHA
Preferred TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.0  128 bits  DHE-RSA-AES128-SHA
Accepted  TLSv1.0  256 bits  DHE-RSA-AES256-SHA
Accepted  TLSv1.0  128 bits  AES128-SHA
Accepted  TLSv1.0  256 bits  AES256-SHA
Accepted  TLSv1.0  256 bits  DHE-RSA-CAMELLIA256-SHA
Accepted  TLSv1.0  256 bits  CAMELLIA256-SHA
Accepted  TLSv1.0  128 bits  DHE-RSA-CAMELLIA128-SHA
Accepted  TLSv1.0  128 bits  CAMELLIA128-SHA
Accepted  TLSv1.0  112 bits  DES-CBC3-SHA
```

Figure 1: SSLScan Output for https://click2park.co.uk

Rapid7 also discovered a Simple Mail Transfer Protocol (SMTP) server using STARTTLS for email encryption. The encryption ciphers for this email server were weak, used a weak certificate hashing algorithm, and used a self-signed certificate, as shown in Figure 2:



```
TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA
TLSv1.0  256 bits  DHE-RSA-AES256-SHA
TLSv1.0  256 bits  DHE-RSA-CAMELLIA256-SHA
TLSv1.0  256 bits  AECDH-AES256-SHA
TLSv1.0  256 bits  ADH-AES256-SHA
TLSv1.0  256 bits  ADH-CAMELLIA256-SHA
TLSv1.0  256 bits  AES256-SHA
TLSv1.0  256 bits  CAMELLIA256-SHA
TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA
TLSv1.0  128 bits  DHE-RSA-AES128-SHA
TLSv1.0  128 bits  DHE-RSA-SEED-SHA
TLSv1.0  128 bits  DHE-RSA-CAMELLIA128-SHA
TLSv1.0  128 bits  AECDH-AES128-SHA
TLSv1.0  128 bits  ADH-AES128-SHA
TLSv1.0  128 bits  ADH-SEED-SHA
TLSv1.0  128 bits  ADH-CAMELLIA128-SHA
TLSv1.0  128 bits  AES128-SHA
TLSv1.0  128 bits  SEED-SHA
TLSv1.0  128 bits  CAMELLIA128-SHA
TLSv1.0  128 bits  ECDHE-RSA-RC4-SHA
TLSv1.0  128 bits  AECDH-RC4-SHA
TLSv1.0  128 bits  ADH-RC4-MD5
TLSv1.0  128 bits  RC4-SHA
TLSv1.0  128 bits  RC4-MD5
TLSv1.0  112 bits  ECDHE-RSA-DES-CBC3-SHA
TLSv1.0  112 bits  EDH-RSA-DES-CBC3-SHA
TLSv1.0  112 bits  AECDH-DES-CBC3-SHA
TLSv1.0  112 bits  ADH-DES-CBC3-SHA
TLSv1.0  112 bits  DES-CBC3-SHA
```

Figure 2: SSLScan Output for click2park.co.uk:25

Affected Hosts

Weak Ciphers (ANON, RC4, 3DES)

- click2park.co.uk:443 (3DES)
- click2park.co.uk:25 (ANON, RC4, 3DES)

TLS 1.0

- click2park.co.uk:443
- click2park.co.uk:25

Recommendations

Rapid7 recommends all web services be configured with only known secure protocols and cipher suites enabled. In addition, XXX should disable support for any weak protocols or cipher suites as it could be leveraged in a downgrade attack. Rapid7 recommends the following protocols and cipher suites:

- TLS 1.2.
- Elliptic Curve Diffie-Hellman (ECDH) with Advanced Encryption Standard (AES) with 128 bits of encryption and above.
- RSA with AES with 128 bits of encryption and above.

References

- http://projects.webappsec.org/w/page/13246945/Insufficient%20Transport%20Layer%20Protection
- https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet
- https://cipherli.st/

Missing Secure Attribute in SSL Session Cookie

DREAD Score Summary

Risk Rating: MODERATE						
Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total	Remediation Effort
3	4	3	5	6	21	LOW

Status

CONFIRMED

Summary of Finding

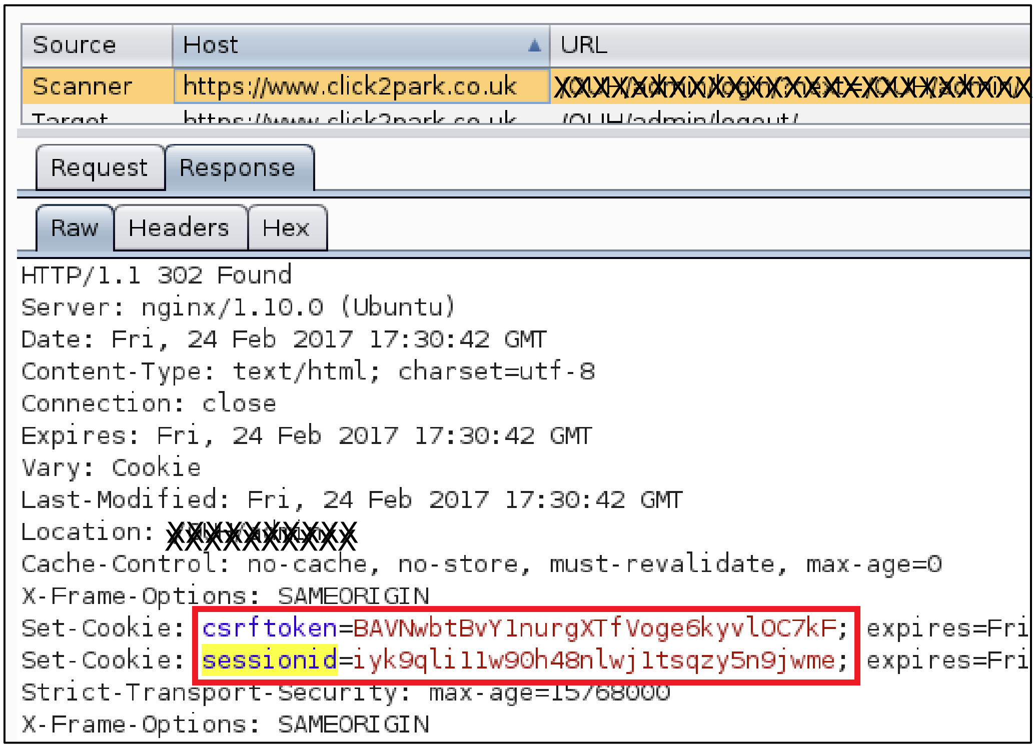
The purpose of setting the secure flag in a cookie is to prevent the unintentional transmission of that cookie over unencrypted communication channels. Request for Comments (RFC) states that, if the cookie does not have the secure attribute assigned to it, the client can pass the cookie to the server over non-secure channels.

Setting the secure flag prevents the cookie from being easily intercepted by a malicious actor monitoring network traffic. If the secure flag is not set, then the cookie will be transmitted as cleartext when the user visits any HTTP URLs within the cookie's scope.

Rapid7 detected that the referenced web application set a cookie without the ‘secure’ attribute during an encrypted session.

Proof of Concept

Rapid7 discovered that the cookie values ‘sessionid’ and ‘csrftoken’ were used for the application’s session management, and the ‘Secure’ value was not set in the server’s response headers, as shown in Figure 3:

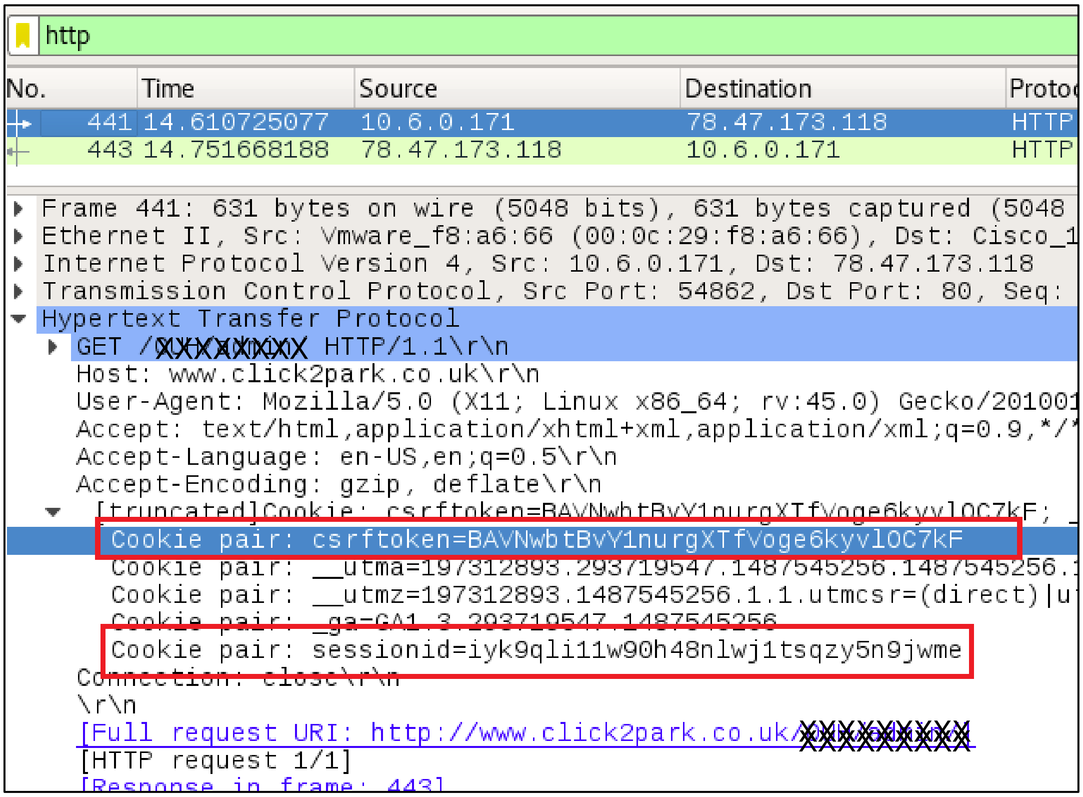


Source	Host	URL
Scanner	https://www.click2park.co.uk	[redacted]
Target	https://www.click2park.co.uk	/OUH/admin/logout/

```
HTTP/1.1 302 Found
Server: nginx/1.10.0 (Ubuntu)
Date: Fri, 24 Feb 2017 17:30:42 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Expires: Fri, 24 Feb 2017 17:30:42 GMT
Vary: Cookie
Last-Modified: Fri, 24 Feb 2017 17:30:42 GMT
Location: [redacted]
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
X-Frame-Options: SAMEORIGIN
Set-Cookie: csrftoken=BAVNwbtBvY1nurgXTfVoge6kyvlOC7kF; expires=Fri
Set-Cookie: sessionid=iyk9qli11w90h48nlwj1tsqzy5n9jwme; expires=Fri
Strict-Transport-Security: max-age=15768000
X-Frame-Options: SAMEORIGIN
```

Figure 3: Server Response Missing Secure Flag

Rapid7 captured the ‘sessionid’ cookie with Wireshark and found sensitive cookie header information, as shown in Figure 4:



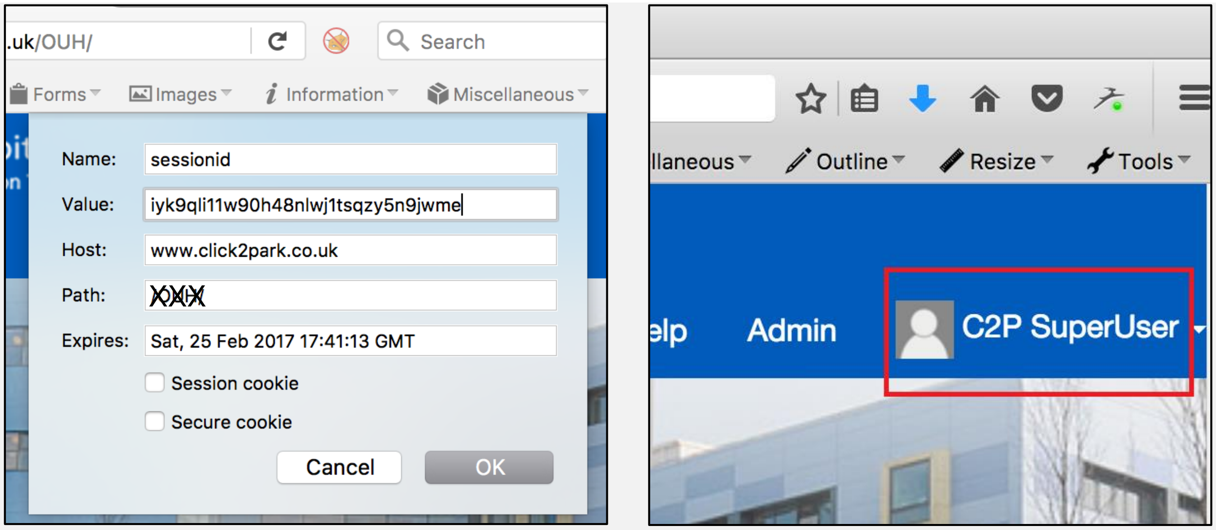
```
http

No.  Time             Source          Destination     Protoc
 441 14.610725077     10.6.0.171      78.47.173.118   HTTP
 443 14.751668188     78.47.173.118   10.6.0.171      HTTP

▶ Frame 441: 631 bytes on wire (5048 bits), 631 bytes captured (5048
▶ Ethernet II, Src: Vmware_f8:a6:66 (00:0c:29:f8:a6:66), Dst: Cisco_1
▶ Internet Protocol Version 4, Src: 10.6.0.171, Dst: 78.47.173.118
▶ Transmission Control Protocol, Src Port: 54862, Dst Port: 80, Seq:
▼ Hypertext Transfer Protocol
  ▶ GET /[redacted] HTTP/1.1\r\n
    Host: www.click2park.co.uk\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/201001
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
  ▼ [truncated]Cookie: csrftoken=BAVNwbtBvY1nurgXTfVoge6kyvlOC7kF; _
      Cookie pair: csrftoken=BAVNwbtBvY1nurgXTfVoge6kyvlOC7kF
      Cookie pair: __utma=197312893.293719547.1487545256.1487545256.1
      Cookie pair: __utmz=197312893.1487545256.1.1.utmcsr=(direct)|u
      Cookie pair: _ga=GA1.3.293719547.1487545256
      Cookie pair: sessionid=iyk9qli11w90h48nlwj1tsqzy5n9jwme
    Connection: close\r\n
    \r\n
    [Full request URI: http://www.click2park.co.uk/[redacted]]
    [HTTP request 1/1]
    [Response in frame: 443]
```

Figure 4: Captured Sensitive Session Information over HTTP

Rapid7 replayed the ‘csrftoken’ and ‘sessionid’ tokens, hijacking the session of the ‘C2P SuperUser’ account, as shown in Figure 5:



Name: sessionid

Value: iyk9qli11w90h48nlwj1tsqzy5n9jwme

Host: www.click2park.co.uk

Path: [redacted]

Expires: Sat, 25 Feb 2017 17:41:13 GMT

☐ Session cookie

☐ Secure cookie

Cancel OK

Figure 5: Session Information Replayed

Affected Hosts

- https://click2park.co.uk/XXXX/ (‘sessionid’ cookie)

Recommendations

It is best practice that any cookies sent (set-cookie) over an encrypted connection to the server be set with the ‘secure’ attribute.

References

- https://www.owasp.org/index.php/SecureFlag
- http://capec.mitre.org/data/definitions/102.html

Samba Server Allows NULL Session Enumeration

DREAD Score Summary

Risk Rating: MODERATE						
Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total	Remediation Effort
3	5	1	2	3	14	LOW

Status

CONFIRMED

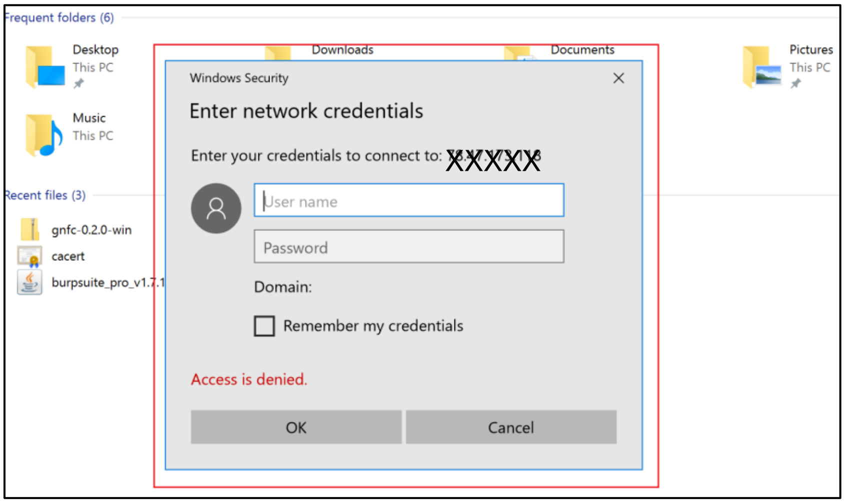
Summary of Finding

Rapid7 observed that the click2park.co.uk/[redacted] webserver allows for the enumeration of valid usernames by an anonymous user through the use of NULL session vulnerability. The NULL session vulnerability allows a NULL user (NULL username and NULL password) to enumerate local information from the affected servers.

A malicious actor could leverage this vulnerability to enumerate valid usernames and other domain resources. With this information, a malicious actor could leverage this information to perform password guessing against domain accounts in an attempt to authenticate to the [redacted] domain.

Proof of Concept

Rapid7 discovered that SMB TCP port 445 was open, and attempted to access the server over SMB from Windows File Explorer, as shown in Figure 6:



Windows Security

Enter network credentials

Enter your credentials to connect to: [redacted]

User name

Password

Domain:

□ Remember my credentials

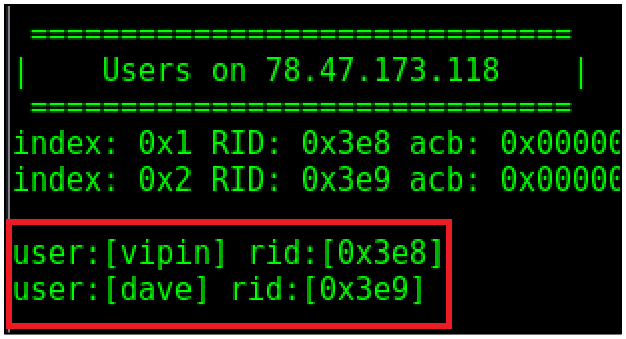
Access is denied.

Figure 6: Connection Established with Samba Server from File Explorer

After finding that the SMB service could be interacted with, Rapid7 attempted to establish a NULL session with the SMB server and enumerate the associated information available to a NULL session with the command:

```
# enum4linux -a 78.47.173.118
```

This resulted in the Rapid7 enumerating information about the Samba system. Figure 7 shows the system users for the SMB server:

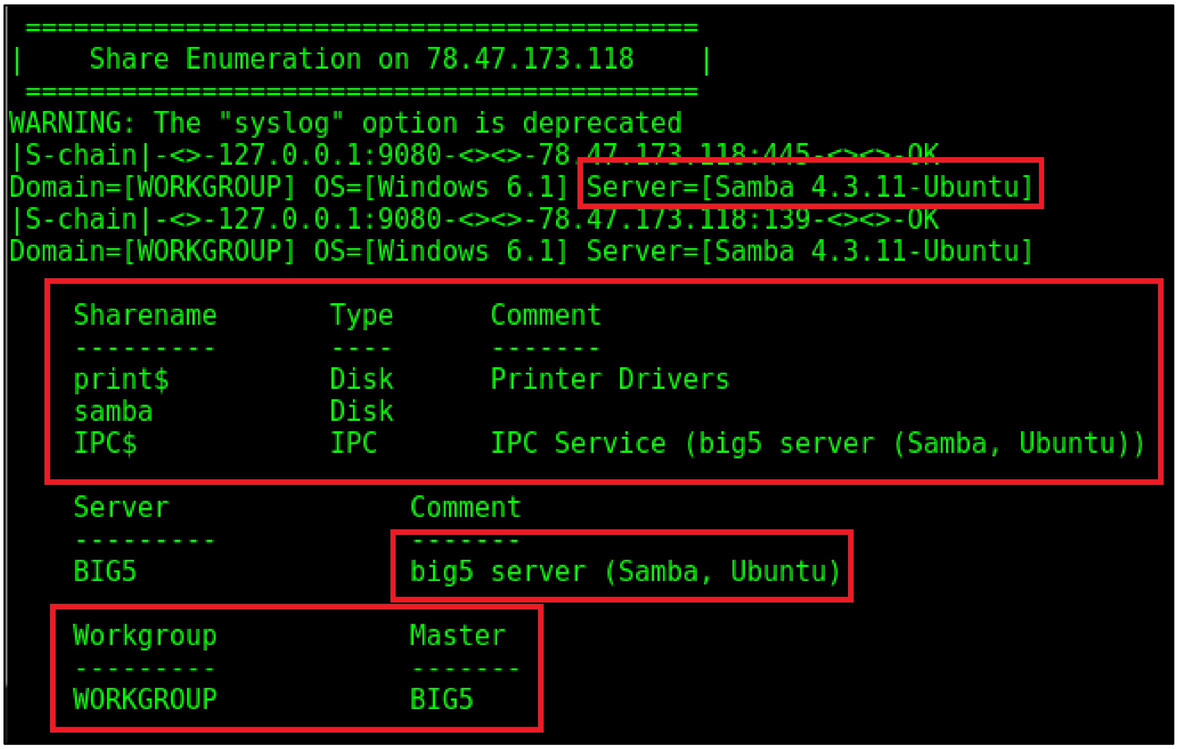


```
 =================================
|    Users on 78.47.173.118    |
 =================================
index: 0x1 RID: 0x3e8 acb: 0x00000
index: 0x2 RID: 0x3e9 acb: 0x00000

user:[vipin] rid:[0x3e8]
user:[dave] rid:[0x3e9]
```

Figure 7: Valid Samba Users Enumerated

Figure 8 shows the server's name, SMB server version, available SMB shares, and the WORKGROUP the server is belongs to:



```
 ==========================================
|    Share Enumeration on 78.47.173.118    |
 ==========================================
WARNING: The "syslog" option is deprecated
|S-chain|-<>-127.0.0.1:9080-<><>-78.47.173.118:445-<><>-OK
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]
|S-chain|-<>-127.0.0.1:9080-<><>-78.47.173.118:139-<><>-OK
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]

    Sharename       Type      Comment
    ---------       ----      -------
    print$          Disk      Printer Drivers
    samba           Disk
    IPC$            IPC       IPC Service (big5 server (Samba, Ubuntu))

    Server               Comment
    ---------            -------
    BIG5                 big5 server (Samba, Ubuntu)

    Workgroup            Master
    ---------            -------
    WORKGROUP            BIG5
```

Figure 8: Sensitive Samba Server Information Disclosed

Affected Hosts

- [redacted]:445

Recommendations

Rapid7 recommends that the organization managing the web server re-evaluate the necessity for exposing SMB TCP port 445 to the internet. If the port is not required for functionality, the organization should block all access to the service. If the SMB service must be externally facing, the organization should whitelist the service to only those who require access and otherwise block all access to the protocol.

Additionally, Rapid7 recommends disabling anonymous and NULL session connections to the Samba server.

References

- https://msdn.microsoft.com/en-us/library/ms913275(v=winembedded.5).aspx
- https://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html

TCP Timestamps Enabled

DREAD Score Summary

Risk Rating: **MODERATE**						
Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total	Remediation Effort
2	5	1	1	4	13	LOW

Status

CONFIRMED

Summary of Finding

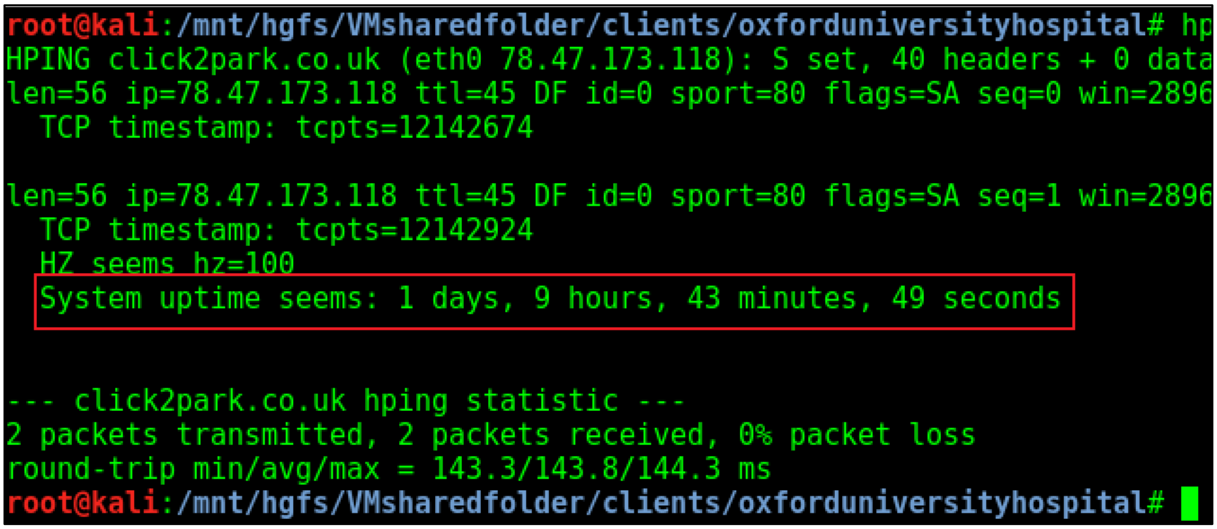
Rapid7 discovered that the remote host responded with a TCP timestamp. The TCP timestamp response can be used to approximate the remote host's uptime, potentially aiding in further attacks. Additionally, some operating systems can be fingerprinted based on the behavior of their TCP timestamps.

Proof of Concept

Using HPing3, Rapid7 found that TCP timestamps were enabled on network hosts by leveraging the external web server with the command:

```
# hping3 -c 2 -5 78.47.173.118 -p 80 --tcp-timestamp
```

The affected hosts responded to Rapid7 with their TCP Timestamps within packet headers, allowing HPing3 to determine systems' uptime, as shown in Figure 9:



```
root@kali:/mnt/hgfs/VMsharedfolder/clients/oxforduniversityhospital# hp
HPING click2park.co.uk (eth0 78.47.173.118): S set, 40 headers + 0 data
len=56 ip=78.47.173.118 ttl=45 DF id=0 sport=80 flags=SA seq=0 win=2896
  TCP timestamp: tcpts=12142674

len=56 ip=78.47.173.118 ttl=45 DF id=0 sport=80 flags=SA seq=1 win=2896
  TCP timestamp: tcpts=12142924
  HZ seems hz=100
  System uptime seems: 1 days, 9 hours, 43 minutes, 49 seconds


--- click2park.co.uk hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 143.3/143.8/144.3 ms
root@kali:/mnt/hgfs/VMsharedfolder/clients/oxforduniversityhospital# 
```

Figure 9: TCP Timestamps Enabled Showing Server System Uptime

Affected Hosts

- click2park.co.uk

Recommendations

Cisco

Disable TCP timestamp responses on Cisco:

Run the following command to disable TCP timestamps.

```
# no ip tcp timestamp
```

FreeBSD

Disable TCP timestamp responses on FreeBSD:

Set the value of `net.inet.tcp.rfc1323` to `0` by running the following command:

```
sysctl -w net.inet.tcp.rfc1323=0
```

Additionally, put the following value in the default sysctl configuration file, generally sysctl.conf:

```
net.inet.tcp.rfc1323=0
```

Linux

```
Disable TCP timestamp responses on Linux
```

Set the value of `net.ipv4.tcp_timestamps` to `0` by running the following command:

```
sysctl -w net.ipv4.tcp_timestamps=0
```

Additionally, put the following value in the default sysctl configuration file, generally sysctl.conf:

```
net.ipv4.tcp_timestamps=0
```

OpenBSD

Disable TCP timestamp responses on OpenBSD

Set the value of `net.inet.tcp.rfc1323` to `0` by running the following command:

```
sysctl -w net.inet.tcp.rfc1323=0
```

Additionally, put the following value in the default sysctl configuration file, generally sysctl.conf:

```
net.inet.tcp.rfc1323=0
```

Disable TCP timestamp responses on Windows versions before Vista:

Set the `Tcp1323Opts` value in the following key to `1`:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Paramet
ers
```

TCP timestamps cannot be reliably disabled on Windows operating systems after Windows Vista. If TCP timestamps present enough of a risk, put a firewall capable of blocking TCP timestamp packets in front of the affected assets.

Low Risk Findings and Remediation Guidance

Outdated Version of Nginx

DREAD Score Summary

Risk Rating: LOW						
Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total	Remediation Effort
1	1	1	2	3	8	LOW

Status

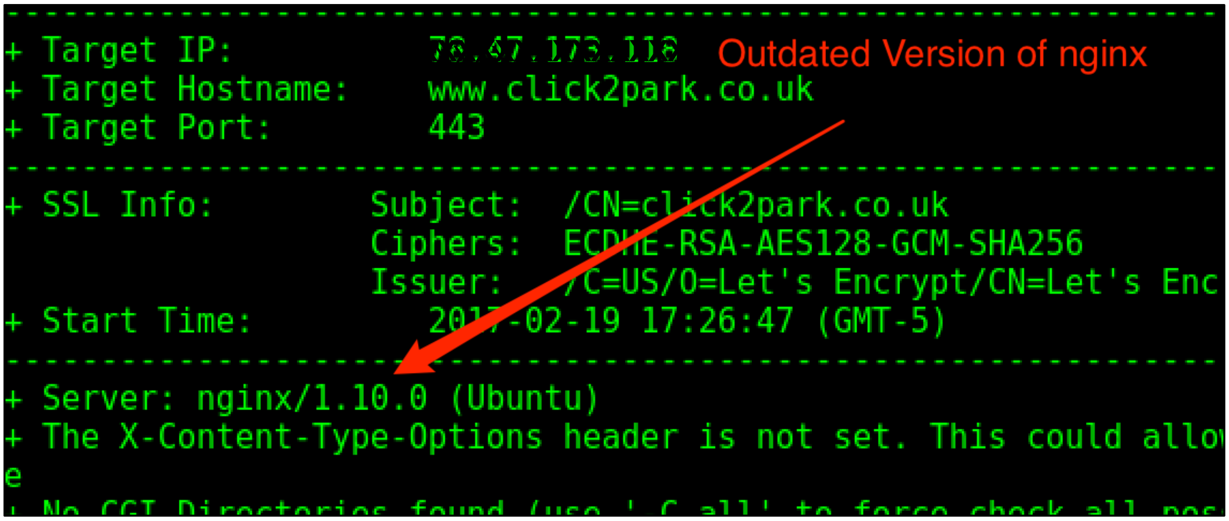
CONFIRMED

Summary of Finding

Rapid7 discovered that the click2park application used an outdated version of Nginx, which could be exploited by an authenticated malicious actor. The Nginx server's version, 1.10.0, is susceptible to Denial of Service (DoS) through crafted requests and writing client request bodies to a temporary file, which can cause a worker process crash and NULL pointer dereference. A malicious actor can also use Nginx 1.10.0 to gain root privileges with access to the local Nginx server by executing a Symlink attack on the error log.

Proof of Concept

Rapid7 discovered an outdated Nginx server version, 1.10.0, by fingerprinting its banner, as shown in Figure 10.



```
---------------------------------------------------------------
+ Target IP:          78.47.173.118
+ Target Hostname:    www.click2park.co.uk
+ Target Port:        443
---------------------------------------------------------------
+ SSL Info:        Subject:  /CN=click2park.co.uk
                   Ciphers:  ECDHE-RSA-AES128-GCM-SHA256
                   Issuer:   /C=US/O=Let's Encrypt/CN=Let's Enc
+ Start Time:         2017-02-19 17:26:47 (GMT-5)
---------------------------------------------------------------
+ Server: nginx/1.10.0 (Ubuntu)
+ The X-Content-Type-Options header is not set. This could allo
e
+ No CGI Directories found (use '-C all' to force check all pos
```

Figure 10: Outdated Nginx Server Version

Affected Hosts

- [redacted]

Recommendations

Rapid7 recommends that Nginx be updated to its latest available version. At this time, this version is 1.10.3.

A number of recommendations can be made in order to detect and prevent vulnerabilities from outdated systems. However, given the findings in this report, detection may be the most critical. A vulnerability management program is a cornerstone piece of any information security department and it consists of the following:

- Monitoring of information sources (such as vendor advisories, mailing lists, etc.) for vulnerability and patch information
- Inventory of all information technology assets within an organization
- Detection of vulnerable assets
- Notification of affected parties
- Patch testing procedures
- Patch installation/rollout procedures
- Tracking of remediation efforts

Rapid7 makes a vulnerability management system, Nexpose, which can help alleviate the workload associated with many of these issues. Similar products are available from other vendors.

References

- https://www.cvedetails.com/cve/CVE-2016-4450/
- https://www.cvedetails.com/cve/CVE-2016-1247/
- http://nginx.org/en/download.html

Appendix A: Rapid7 Contacts

Jacob Robles			
Title:	Security Analyst		
Telephone:	[redacted]	E-mail:	[redacted]
Jonathan Stines			
Title:	Security Consultant		
Telephone:	[redacted]	E-mail:	[redacted]
Andrew Whitaker			
Title:	Director, Global Services		
Telephone:	[redacted]	E-mail:	[redacted]
Todd Lefkowitz			
Title:	Vice President, Global Services		
Telephone:	[redacted]	E-mail:	[redacted]

Table 6: Points of Contact

Rapid7 Headquarters
100 Summer Street, 13th Floor
Boston, MA 02110

Appendix B: Engagement Scope Overview

Rules of Engagement and Assumptions

- Testing to occur during normal business hours.
- No Denial of Service (DoS) attacks.

Accounts

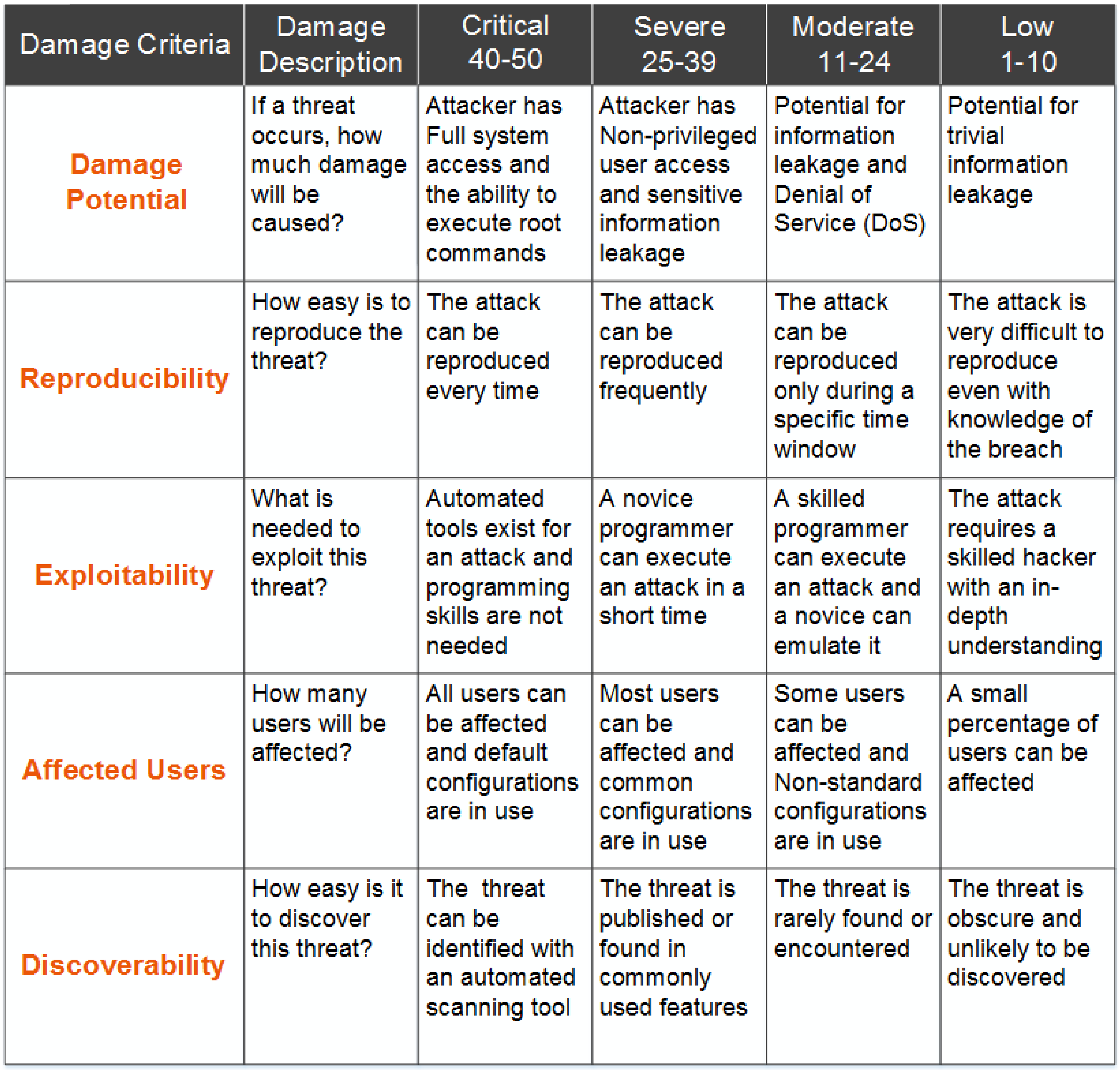
Rapid7 was provided with the following accounts and email addresses to login to the application:

Role	Account Email Address
Superuser	[redacted]
Parking Officer	[redacted]

Scope Targets

Rapid7 tested the unauthenticated portion of the click2park application, two roles for the application, and supporting infrastructure for click2park.co.uk/[redacted].

Appendix C: DREAD Overview



Damage Criteria	Damage Description	Critical 40-50	Severe 25-39	Moderate 11-24	Low 1-10
Damage Potential	If a threat occurs, how much damage will be caused?	Attacker has Full system access and the ability to execute root commands	Attacker has Non-privileged user access and sensitive information leakage	Potential for information leakage and Denial of Service (DoS)	Potential for trivial information leakage
Reproducibility	How easy is to reproduce the threat?	The attack can be reproduced every time	The attack can be reproduced frequently	The attack can be reproduced only during a specific time window	The attack is very difficult to reproduce even with knowledge of the breach
Exploitability	What is needed to exploit this threat?	Automated tools exist for an attack and programming skills are not needed	A novice programmer can execute an attack in a short time	A skilled programmer can execute an attack and a novice can emulate it	The attack requires a skilled hacker with an in-depth understanding
Affected Users	How many users will be affected?	All users can be affected and default configurations are in use	Most users can be affected and common configurations are in use	Some users can be affected and Non-standard configurations are in use	A small percentage of users can be affected
Discoverability	How easy is it to discover this threat?	The threat can be identified with an automated scanning tool	The threat is published or found in commonly used features	The threat is rarely found or encountered	The threat is obscure and unlikely to be discovered

Table 7: DREAD Score Overview

Appendix D: Manual Application Testing and OWASP Testing Methodology

Rapid7 will format the results of the Web Application Penetration Test according to the Full OWASP Testing Methodology. This will aid in the remediation process.

Access Control, Authorization and Authentication focuses on testing the business logic of the web application.

- *Parameter Analysis* to understand whether the application enforces its access control model by ensuring that any parameters available to a malicious actor would not afford additional service.
- *Authorization Testing* to understand whether resources that require authorization perform adequate authorization checks before being sent to a user.
- *Authorization Parameter Manipulation* to check that once a valid user has logged in, it is not possible to change the session ID's parameter to reflect another user account.
- *Authorized pages and functions testing* to check if it is possible to access pages or functions that require logon but can be bypassed.
- *Application Workflow testing* to determine whether, in situations that the application requires the user to perform actions in a specific sequence, that the sequence is enforced.
- *Authentication endpoint testing* to determine if requests are using HTTPS when users are asked to submit authentication credentials on pages that are secured with SSL.
- *Authentication Bypass testing* to determine if the authentication process can be bypassed.
- *Credentials transport over an encrypted channel* to understand whether usernames and passwords are sent over an encrypted channel.
- *Default Accounts testing* to check for default account names and passwords in use.
- *Username testing* to ensure that the username is not public information such as e-mail or SSN.
- *Password Quality testing* to validate that the password complexity policy makes guessing passwords difficult.
- *Password Reset testing* to test to see that someone other than the user cannot intercept the password during the reset process.
- *Password Lockout testing* to test to see that the users account us locked out for a period of time when an incorrect password is entered more than a specific number of times.
- *Password Structure testing* to test to see that special meta-characters cannot be used within the password.
- *Blank Passwords testing* to check that passwords cannot be blank.

Session Management focuses on testing the session management implemented within the web application.

- *Session Token Length testing* to determine whether the session token is of adequate length to provide protection from session hijacking attacks.
- *Session Timeout testing* to validate that the session tokens are only valid for a predetermined period after the last request by the users.
- *Session Reuse testing* to validate that session tokens are changed when a user moves from an SSL protected resource to a non-SSL protected resource.
- *Session Deletion testing* to validate that the session token is invalidated when the user logs out.
- *Session Token Format testing* to validate that the session token is non-persistent and is never written to the browsers history or cache.

Configuration Management and Application Architecture Testing focuses on testing the configuration of the web application.

- *HTTP Methods testing* to validate that the web server does not support the ability to manipulate resources from the Internet.
- *Website Debugging testing* to validate that the web server debugging methods are disabled for production websites.
- *Virtual Hosts testing* to attempt to determine if the site is a virtual host.
- *Web Server testing* to attempt to confirm that the Web Server is free from known vulnerabilities and have all security patches applied.
- *Back-up Files testing* to test to see that no backup files of source code are accessible on the publicly accessible part of the web server.
- *Web Server Configuration testing* to test to see that common configuration issues such as directory listing and sample files have been addressed.
- *Web Server Components testing* to attempt to validate that web server components (ASP, PHP, WebDAV, Apache Modules, etc.) do not introduce any security vulnerabilities.
- *Common Paths testing* to attempt to identify common directories within the application root.
- *Language/Application Defaults testing* to identify whether quirks of any application or language used do not introduce any security vulnerabilities.
- *Infrastructure Admin Interfaces testing* to validate that administrative interfaces to infrastructure, such as web servers and application servers, are not accessible from the Internet.
- *Application Admin Interfaces testing* to validate that administrative interfaces to application are not accessible from the Internet.

Error Handling focuses on testing the error handling of the web application.

- *Application Error Messages testing* to validate that the application does not present error messages from the application (such as stack traces or database errors) that could be used in an attack.
- *User Error Messages testing* to validate that the application does not present user error messages from the application (such as 'User does not exist' or 'User Correct, Password Incorrect') that could be used in an attack.

Data Protection focuses on protecting the data within the web application.

- *Sensitive Data in HTML testing* to attempt to validate that there is no sensitive data in the HTML (cached browser history) that could assist and a malicious actor.
- *Data Storage testing* to attempt to verify that data is protected in such a way so as to protect its confidentiality and integrity, where required.
- *SSL Version testing* to check that supported SSL versions do not have cryptographic weaknesses.
- *SSL Key Exchange Methods testing* to check that the web server does not allow anonymous key exchange methods.
- *SSL Algorithms testing* to check that weak algorithms are not available.
- *SSL Key Lengths testing* to determine whether the website uses an appropriate key length.
- *Digital Certificate Validity testing* to determine whether the application uses a valid digital certificate.

Input Validation focuses on testing the input validation of the web application.

- *Script Injection testing* to validate that any part of the application that allows user input does not process scripts as part of the input.
- *SQL Injection testing* to validate the application will not process SQL commands from the user.
- *Command Injection testing* to validate the user-supplied commands are not processed by the OS.
- *LDAP Injection testing* validate the application will not process LDAP commands from the user.
- *Cross-Site Scripting testing* to validate that the application will not store or reflect malicious script code.