iSECpartners
part of nccgroup

Psiphon Inc.
Psiphon 3

Security Assessment

Prepared for:

psiphon

Prepared by:

Mark Manning — Security Engineer

Anson Gomes — Security Engineer

Andrew Rahimi — Security Engineer

While precautions have been taken in the preparation of this document, iSEC Partners, Inc, the publisher, and the author(s) assume no responsibility for errors, omissions, or for damages resulting from the use of the information contained herein. Use of iSEC Partners services does not guarantee the security of a system, or that computer intrusions will not occur.

Table of Contents

1 Executive Summary

Application Summary	
Application Name	Psiphon 3
Application Version	3.0 - Client Version: 78
Application Type	Windows and Android Clients
Platform	Python, C++, Debian 6 & 7, Ubuntu 14

Engagement Summary	
Dates	July 21, 2014 – August 22, 2014
Consultants Engaged	Three
Total Engagement Effort	Six person-weeks
Engagement Type	Security Assessment
Testing Methodology	White Box

Vulnerability Summary	
Total High severity issues	1
Total Medium severity issues	1
Total Low severity issues	3
Total Informational severity issues	6

Total vulnerabilities identified: 11

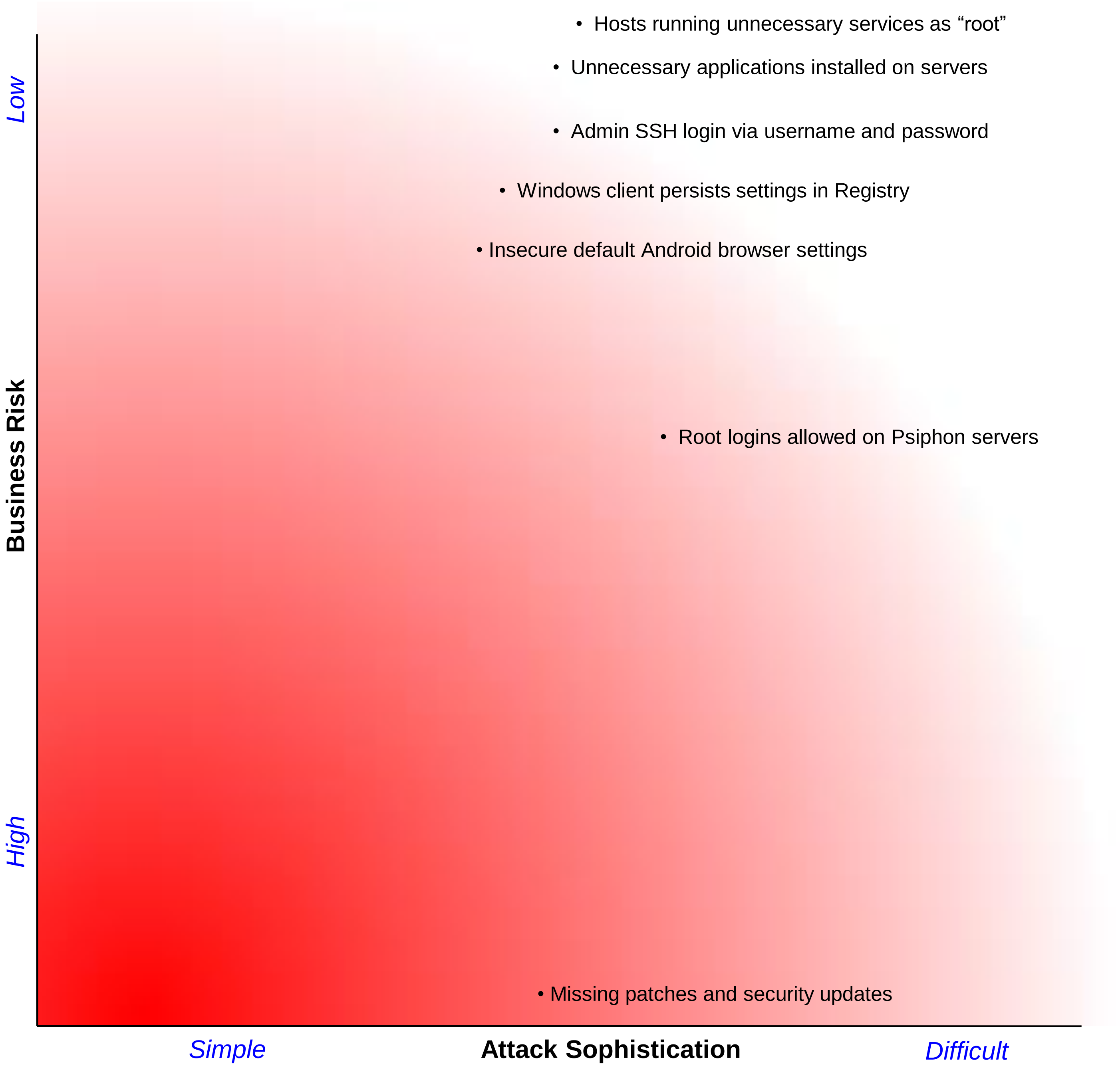
See section 3.1 on page 12 for descriptions of these classifications.

Category Breakdown:

Access Controls	0	
Auditing and Logging	0	
Authentication	0	
Configuration	8	■■■■■■■■
Cryptography	0	
Data Exposure	1	■
Data Validation	0	
Denial of Service	0	
Error Reporting	0	
Patching	2	■■
Session Management	0	
Timing	0	

1.1 iSEC Risk Summary

The iSEC Partners Risk Summary chart evaluates discovered vulnerabilities according to user risk. The impact of the vulnerability increases towards the bottom of the chart. The sophistication required for an attacker to find and exploit the flaw decreases towards the left of the chart. The closer a vulnerability is to the chart origin, the greater the risk to Psiphon users.



1.2 Project Summary

Open Technology Fund[1] engaged iSEC Partners (iSEC), an NCC Group company, to perform a security assessment for Psiphon Inc. (Psiphon) to evaluate their Psiphon 3 software. The Psiphon software is designed for censorship circumvention through the use of VPN, SSH, Obfuscated SSH (OSSH), and HTTP/SOCKS Proxy technologies to bypass enforced restrictions and provide users with uncensored access to Internet content. The end goal of which is to provide residents of censored states a way to gain unfettered access to the Internet.

The Psiphon 3 software has numerous components that were deemed in scope for the engagement. These included the Psiphon Android and Windows clients, Psiphon server configuration, fingerprinting defense mechanism, OSSH protocol, automation server, feedback system, and stats tracking.

Three iSEC consultants worked for a total of six person-weeks and completed the engagement across three calendar-weeks that started on July 21, 2014 and ended on August 22, 2014, with breaks in between. The engagement was white box in nature and Psiphon provided iSEC with source code, test clients for Android and Windows, and credentials to access a dedicated Psiphon server.

The engagement began with a short architecture review using documentation provided by the Psiphon team and information garnered through technical interviews conducted with the various team members. This was followed by a review of the Psiphon server configurations and subsequently an assessment of the security of the Psiphon Android application.

Later phases of the project expanded the testing team and focused on assessing the Psiphon Windows client, analyzing network traffic, and assessing the server discovery protocol. A review of the OSSH handshake implementation was also performed and iSEC performed a cursory analysis of the Psiphon automation scripts that are used for server management and other infrastructure logistics.

The Psiphon team was very helpful during the testing process. The team made themselves readily available to respond to queries and resolved any issues or difficulties that were encountered during the testing phase.

1.3 Findings Summary

The following report reflects that during the assessment, iSEC identified many areas where the Psiphon team was following industry best practices, or were properly mitigating any threats. Many of the findings identified are considered as suggested improvements to a defense-in-depth approach to their system. No inherent architecture flaws were identified during the testing period and the developers have shown to be actively invested in ensuring the security of their users.

As Psiphon continues to grow both from an organizational perspective and in infrastructure, their resources may be difficult to scale. For this reason, iSEC has made recommendations that can assist in their growth and believes that the improvements to logging, host security, and management will help allow the group to expand while minimizing the risk of affecting their security posture.

[1] https://www.opentechfund.org/

1.4 Recommendations Summary

In general, Psiphon should continue to invest time hardening the hosts that are running the Psiphon tools. This includes the supporting infrastructure like the automatic email responder as well as their core operating environment that runs the Psiphon 3 suite of services. The organization should also start to make enterprise-grade improvements to the way servers are managed as well as making it a goal to have a better perspective of what types of attacks occur, real-time, on the network.

Short Term

Short term recommendations are meant to be relatively easily executed actions, such as configuration changes or file deletions that resolve security vulnerabilities. These may also include more difficult actions that should be taken immediately to resolve high-risk vulnerabilities. This area is a summary of short term recommendations; additional recommendations can be found in the vulnerabilities section.

Keep Psiphon servers patched and updated. Ensure that all systems stay up to date without affecting Psiphon services. If automatic updates pose too much of a risk to the stability of the server, using a tool like Ansible[2] may make automating the update process of approved patches easier.

Enable alerts for active attacks. When systems are being attacked, administrators should receive alerts to effectively respond. The `fail2ban` tool should be configured to alert on repeated attacks.

Disable risky features in the Android browser. The Psiphon browser application that is integrated into the Android client can help mitigate the risk of exploitation or attack by disabling JavaScript, auto-complete of form fields, and caching of sensitive information, by default.

Long Term

Long term recommendations are more complex and systematic changes that should be taken to secure the system. These may include significant changes to the architecture or code and may therefore require in-depth planning, complex testing, significant development time, or changes to the user experience that require retraining.

Integrate a patch management process. Patch management systems and configuration management tools allow administrators to manage mass quantities of systems with little effort. The management process should include a review of each of the patches to ensure they do not affect the Psiphon servers and their users.

Consolidate authentication to the Psiphon servers. Consider implementing a consolidated authentication approach like LDAP. This would improve the control the organization has over auditing access to the servers as well as make it easier to scale as Psiphon grows (Appendix B on page 27).

Implement an organization-wide SIEM/IDS. A SIEM/IDS/Log Aggregator could provide better insight over what is currently happening on the Psiphon network and reduce the amount of time to react to events. For more information see Appendix A.1 on page 26.

[2] http://www.ansible.com

2 Engagement Structure

2.1 Internal and External Teams

The iSEC team has the following primary members:

- Mark Manning — Security Engineer
  mmanning@isecpartners.com
- Anson Gomes — Security Engineer
  anson@isecpartners.com
- Andrew Rahimi — Security Engineer
  arahimi@isecpartners.com
- Tom Ritter — Account Manager
  tritter@isecpartners.com
- Dana Bost — Project Manager
  dbost@isecpartners.com

The Psiphon team has the following primary members:

- Rod Hynes — Psiphon Primary Contact
  r.hynes@psiphon.ca
- Karl Kathuria — Psiphon
  k.kathuria@psiphon.ca
- Michael Hull — Psiphon
  m.hull@psiphon.ca
- Adam Pritchard — Psiphon
  a.pitchard@psiphon.ca

2.2 Project Goals and Scope

The goal of this engagement was to identify concerns in the Psiphon 3 that endangers individuals using the system to evade Internet censorship in their respective country and mechanisms that would cause Psiphon to become unavailable to those users. Discussions with the Psiphon team led iSEC to focus on the following adversarial threats:

- Failure to circumvent and bypass Internet censorship (e.g.: The Psiphon software fails to tunnel traffic)
- Adversary enumerates/blocks Psiphon servers (e.g.: The failure to partition servers or scanning signature for server hosts)
- Adversary identifies and blocks Psiphon network traffic between clients and servers (e.g.: The traffic signature or deep packet inspection(DPI) of the protocol)
- Exploitation of the Psiphon client software (e.g.: Vulnerabilities like buffer overflows in Polipo via attack-crafted web pages as well as the Psiphon client upgrade mechanism)

Threat Model

The Psiphon client's threat model, as defined by the Psiphon team, is focused on adversaries interested in blocking the censorship-evading capabilities of the application, and/or attacking the Psiphon users via vulnerabilities in the application. The application does not aim to provide any type of anonymity, pseudonimity, or privacy-enhancing features. Because of this, the threat model does not factor in attacks on user privacy unless it affects the software's primary goal of Internet censorship evasion. iSEC consultants did not review the software for information tracking purposes, or privacy issues as this was deemed outside the threat model, and therefore out of scope. As Psiphon continues to evolve to respond to new types of attacks (be it technical or political), its threat model will follow suit and may need to adapt beyond what is currently in scope today.

Areas of Focus and Completed Coverage

The Psiphon project consists of a variety of parts including the Android client, Windows client, third-party libraries, hosted servers, and scripts and tools used for automation. iSEC worked with the Psiphon team to come up with specific areas believed to be of concern to the Psiphon and their users within the time constraints of the project. Below is an overview of areas of focus, items that have been partially reviewed, and areas that were given a cursory review:

Completed Review	
Component	**Status**
Psiphon Android Library and Client	Reviewed for common issues related to storage, IPC mechanisms, and implementing third-party code.
Psiphon Windows Client	Reviewed the Psiphon Windows client software and the integration and usage of third-party libraries with the client.
Psiphon Network Traffic Fingerprint	Reviewed network traffic to examine traffic between the Psiphon client and server application for patterns and fingerprints.
Psiphon Server Discovery Protocol	Assessed the Psiphon automated server discovery protocol to verify if an attacker could enumerate all the Psiphon servers at a given time.
Psiphon Server Host Configuration and Hardening	Reviewed for common misconfigurations, bad permissioning, insecure practices, and other areas where the server could potentially be compromised.
Psiphon OSSH Handshake	Reviewed the Obfuscated SSH protocol handshake between the Psiphon Client and OSSH service hosted on Psiphon servers. The results of which showed that it defended against fingerprinting by appearing as random data. The OSSH protocol itself would potentially be vulnerable to identification by sniffing the initial key exchange, but Psiphon mitigates this risk by exchanging the OSSH ”keyword” value out-of-band through the server list exchange. Without this ”keyword” value, it would require heavy resources for an adversary to properly fingerprint and block the OSSH handshake using DPI, based on the Layer 4 contents alone.
Psiphon Automailer System	Reviewed the automailer tools, host configuration, and security practices.
Psiphon Web Server	Reviewed the Cherry Py web server implementation including input validation mechanisms, sensitive logging functions, and general web application vulnerabilities.

Partially Reviewed	
Component	**Status**
Psiphon Automation Scripts	Reviewed many of the automation scripts within the testing period. Due to the number of scripts, a complete assessment could not be done during the testing period.
Psiphon Stats Tracking and Feedback Systems	Reviewed the stats tracking and feedback mechanisms looking for potential flaws in design or vulnerabilities in the communication occurs. iSEC did not review the privacy implications of stats tracking or what information was being logged as this was outside of the Psiphon threat model.

Cursory Review	
Component	**Status**
Third-Party Modules	iSEC focused on third-party implementations but did not perform in-depth review of all third-party modules source code. This include the Android browser, DNS server, socks proxy, and meek code.
Psiphon `badVPN` Implementation	Reviewed the VPN configuration practices including cryptographic functions. A complete assessment of `badVPN` code base that implements the VPN was not performed.
Third-Party Android Libraries	The Android libraries were reviewed for their implementation but due to time constraints a full source review of the SSH, meek, and Zirco browser could not be performed.

3 Detailed Findings

3.1 Classifications

The following section describes the classes, severities, and exploitation difficulty rating assigned to each identified issue by iSEC.

Vulnerability Classes

Class	Description
Access Controls	Related to authorization of users, and assessment of rights
Auditing and Logging	Related to auditing of actions, or logging of problems
Authentication	Related to the identification of users
Configuration	Related to security configurations of servers, devices, or software
Cryptography	Related to mathematical protections for data
Data Exposure	Related to unintended exposure of sensitive information
Data Validation	Related to improper reliance on the structure or values of data
Denial of Service	Related to causing system failure
Error Reporting	Related to the reporting of error conditions in a secure fashion
Patching	Related to keeping software up to date
Session Management	Related to the identification of authenticated users
Timing	Related to the race conditions, locking, or order of operations

Severity Categories

Severity	Description
Informational	The issue does not pose an immediate risk, but is relevant to security best practices or Defense in Depth
Undetermined	The extent of the risk was not determined during this engagement
Low	The risk is relatively small, or is not a risk the client has indicated is important
Medium	Individual user's information is at risk, exploitation would be bad for client's reputation, of moderate financial impact, possible legal implications for client
High	Large numbers of users, very bad for client's reputation or serious legal implications.

Difficulty Levels

Difficulty	Description
Undetermined	The difficulty of exploit was not determined during this engagement
Low	Commonly exploited, public tools exist or can be scripted that exploit this flaw
Medium	Attackers must write an exploit, or need an in depth knowledge of a complex system
High	The attacker must have privileged insider access to the system, may need to know extremely complex technical details or must discover other weaknesses in order to exploit this issue

3.2 Vulnerabilities

The following tables are a summary of iSEC's identified vulnerabilities. Subsequent pages of this report detail each of the vulnerabilities, along with short and long term remediation advice.

Psiphon Server Vulnerabilities

Vulnerability	Class	Severity
1. Missing patches and security updates	Patching	High
2. Root logins allowed on Psiphon servers	Configuration	Medium
3. Admin SSH login via username and password	Configuration	Low
4. Unnecessary applications installed on servers	Configuration	Low
5. Hosts running unnecessary services as "root"	Configuration	Low
6. `fail2ban` does not alert on attacks	Configuration	Informational
7. SSH service displays sensitive information in banner on login	Configuration	Informational
8. Weak encryption standards for SSH	Configuration	Informational

Psiphon Android Client Vulnerabilities

Vulnerability	Class	Severity
9. Insecure default Android browser settings	Configuration	Informational

Psiphon Windows Client Vulnerabilities

Vulnerability	Class	Severity
10. Windows client persists settings in Registry	Data Exposure	Informational
11. Windows client update script unreliable	Patching	Informational

3.3 Detailed Vulnerability List

Psiphon Server Vulnerabilities

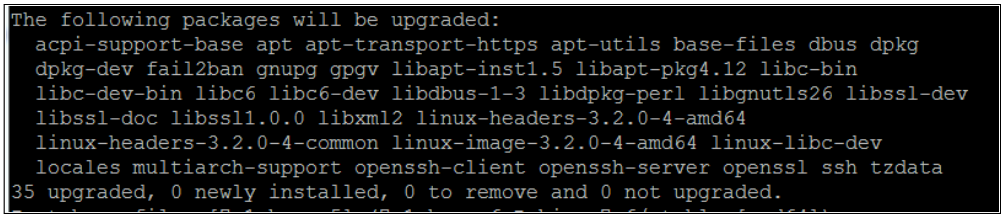
1. Missing patches and security updates

Class: Patching **Severity:** High **Difficulty:** Medium

Finding ID: iSEC-OTFPSI14-01

Targets: The Psiphon 3 servers.

Description: Psiphon relay servers do not support a patch management system. Updates are applied to systems manually when deemed necessary, but machines do not receive regular updates. While remotely exploitable issues are believed to be regularly patched (e.g. Heartbleed), the lack of consistency across systems increases the risk of a vulnerable application resulting in a server compromise.



```
The following packages will be upgraded:
  acpi-support-base apt apt-transport-https apt-utils base-files dbus dpkg
  dpkg-dev fail2ban gnupg gpgv libapt-inst1.5 libapt-pkg4.12 libc-bin
  libc-dev-bin libc6 libc6-dev libdbus-1-3 libdpkg-perl libgnutls26 libssl-dev
  libssl-doc libssl1.0.0 libxml2 linux-headers-3.2.0-4-amd64
  linux-headers-3.2.0-4-common linux-image-3.2.0-4-amd64 linux-libc-dev
  locales multiarch-support openssh-client openssh-server openssl ssh tzdata
35 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Figure 1: Available updates on Psiphon server

NOTE: The affected systems above are from the dedicated testing servers, and not a publicly used Psiphon system.

Exploit Scenario: A vulnerability is discovered in the Python CherryPy web server used by Psiphon. This allows an attacker to SSH into the system and remotely exploit the web server to compromise the host. The attacker gains root access and installs keyloggers as well as injects malicious content into Psiphon clients' web sessions. This in-turn results in the compromise of users' email credentials.

Short Term Solution: On systems that support an automatic update process, consider enabling the automatic download and installation of security related packages. On the Ubuntu servers, for example, Ubuntu supports automatic updates on a regular basis.[3]

Long Term Solution: Integrate an upgrade process into the Psiphon host management system. Administrators should be alerted of which hosts are missing patches, and be able to remotely issue update commands. IT management tools like Nagios[4] may help provide this function. Patch management tools that work over SSH, such as Ansible,[5] will also make it easier for administrators to integrate patches across large numbers of servers.

[3] https://help.ubuntu.com/community/AutomaticSecurityUpdates

[4] http://www.nagios.org/

[5] http://www.ansible.com/

2. Root logins allowed on Psiphon servers

Class: Configuration **Severity:** Medium **Difficulty:** High

Finding ID: iSEC-OTFPSI14-02

Targets: The Psiphon 3 servers.

Description: The OpenSSH service used for administration on port 2222/TCP allows the root user to log in. Administrators use this access to remotely manage the system. All commands issued during that SSH session will run as the root user, even scripts and commands that do not require root access to the system. Root access should only be provided when necessary, and restricted in all other cases. If these credentials were compromised, an attacker would be able to gain full access to the system. Furthermore, it is impossible to audit who logged in and performed what activities in the case of a compromise.

Exploit Scenario: The SSH credentials are found on a developer's system, and used to log in and compromise the remote host. With access, the adversary is able to inject malicious information into the Psiphon users' web traffic to hijack their sessions, and compromise their accounts.

Short Term Solution: Disable remote root logins. Instead, create separate user accounts on the systems and use the `sudo` command. Log all attempts to use the `sudo` command. Administrators and scripts should run as separate users and have limited privileges.

Long Term Solution: Make sure that future Psiphon images do not support `root` login neither interactively nor via SSH. Consider consolidating authentication with a solution like LDAP. See Appendix B on page 27.

3. Admin SSH login via username and password

Class: Configuration	**Severity:** Low	**Difficulty:** High

Finding ID: iSEC-OTFPSI14-03

Targets: The Psiphon 3 servers.

Description: Remote administration done on port 2222/TCP via SSH supports authentication via a username and password instead of key authentication. When SSH supports logins via username and password combination, this puts them at risk of online guessing attacks. Although Psiphon employed strong passwords (178 bits)[6] for each server, a more defense-in-depth approach would be to configure the SSH service to only authenticate via SSH key, and block all attempts to login via password.

Exploit Scenario: A service is installed on the host that automatically creates a user account for itself with a weak password. This account is guessable by an attacker and leads to the system being compromised.

Short Term Solution: Configure the administrative SSH service to only allow key authentication to a designated account. Drop all other requests to log in via standard username and password. Configure `fail2ban` to automatically ban any attempts to authentication via username and password. Update the automation scripts to handle managing SSH keys instead of passwords.

Long Term Solution: Integrate this policy into future Psiphon image building processes.

[6] https://bitbucket.org/psiphon/psiphon-circumvention-system/src/40cc1a2c7672e676d1ff7780f18a3d76bd96e431/Automation/psi_utils.py?at=default#cl-207

4. Unnecessary applications installed on servers

Class: Configuration **Severity:** Low **Difficulty:** High

Finding ID: iSEC-OTFPSI14-04

Targets: The Psiphon 3 servers.

Description: The Psiphon server images contain a variety of unnecessary applications that would make it easier for an attacker to take advantage of. Unused applications create unnecessary risk of being compromised by an exploitable vulnerability, or to be used in chain of other commands to complete an attack. One example is compilation tools such as `gcc` that allow source code to be compiled into a system-specific binary. A system's attack surface should be minimized as much as possible.

- gcc
- whois
- www-browser
- nc
- wget
- ssh
- sftp
- ftp

Exploit Scenario: An unused application that contains a vulnerability is installed on a system. An attacker finds a way to gain a shell on the server, and escalates to full root access by exploiting this vulnerability in the unused software. With access, they can inject malicious payloads into a Psiphon user's requests resulting in their user compromise.

Short Term Solution: Review common hosting images and remove unnecessary applications including build tools such as `gcc`. Apply these changes to future versions of Psiphon servers that are built.

Long Term Solution: Consider building a custom image from the ground up that starts with the bare minimum. Debian "Minimal" builds, for example, provide a small instance of the OS with only the bare-essentials. Build this image to include other applications as necessary. Integrate the design changes into future host hardening procedures.

5. Hosts running unnecessary services as "root"

Class: Configuration	**Severity:** Low	**Difficulty:** High

Finding ID: iSEC-OTFPSI14-05

Targets: The Psiphon 3 servers.

Description: The following processes were running as the root user on the supplied server: `fail2ban`, `meek-server`, and `_plutorun_`. Some of these processes, such as `meek-server`, are remotely accessible services. Running processes as root puts the system at risk of being completely compromised in the case one of these processes is exploited. This goes against the principle of least privilege when running processes on a system.

Exploit Scenario: The meek service is remotely compromised. This results in the attacker gaining complete control over the system and manipulating the traffic of Psiphon users.

Short Term Solution: Ensure that services such as SSH are configured as a dedicated user without root access. Permissions should be granted on an as-needed basis following the principal of least privilege.

Long Term Solution: Consider implementing stricter controls of what a single process can gain access to. Besides the already enabled address space randomization build flags that were found to be enabled on the Psiphon servers, other tools like `systemd` and `SELinux` are security mechanisms that emulate root environments and define exactly how a process can interact with the underlying system. Even in the case of a compromise, these processes should be contained to only those objects they require access to rather than the entire machine.[7]

[7] https://wiki.archlinux.org/index.php/fail2ban

6. fail2ban does not alert on attacks

Class: Configuration **Severity:** Informational **Difficulty:** NA

Finding ID: iSEC-OTFPSI14-06

Targets: The fail2ban service on the Psiphon servers.

Description: The fail2ban service used to protect Psiphon services does not send out alerts when an issue is raised. While it is configured to actively defend against attacks by blocking repeated attacks, the Mail Transfer Agent (MTA) is not configured. It may not be possible for Psiphon administrators to know when an issue has occurred leaving a potential for servers to be exploited and left compromised over extended periods of time.

Exploit Scenario: An attacker exploits a Psiphon service and gains remote access, while the fail2ban service detects the irregularity, administrators are not notified of the system compromise. The attacker takes over the system including injecting malicious data into Psiphon clients.

Short Term Solution: Configure the MTA to send alerts to a designated administrator, mailing list, or bug-tracking tool that is able to handle the requests and take action. Repeated false positives can be used to create a more customized ruleset that over time can be used to ensure only critical attacks result in a notification.

Long Term Solution: Consider implementing an enterprise-grade log management system/IDS that would provide a perspective on the Psiphon network operations. Splunk [8] is an example of a tool that could consolidate logs and help administrators respond to issues.

[8] http://www.splunk.com/

7. SSH service displays sensitive information in banner on login

Class: Configuration **Severity:** Informational **Difficulty:** NA

Finding ID: iSEC-OTFPSI14-06

Targets: The SSH banner of the Psiphon servers.

Description: Psiphon 3 by design grants untrusted users the ability to connect to Psiphon servers via SSH. These users are unable to access a shell and therefore cannot execute commands. However, upon logging in, the SSH service displays the MOTD banner that includes information about the host being logged into. This includes the Linux version and distribution. While this information alone does not present an exploitable vulnerability, it may give attackers information about how to attack the host in the future. The iSEC team extracted the SSH username and password out of the Psiphon client server list, and logged in using an SSH client. While the host was configured to disallow shell access, the following is the output to the terminal during the login process:

```
Using username "psiphon_ssh_92a09d5b5e7b6b58".
Using keyboard-interactive authentication.
Password:
Linux do-665-pzqlrqvo 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Could not chdir to home directory /dev/null: Not a directory
```

Listing 1: SSH banner on Psiphon server

Exploit Scenario: An attacker extracts the login credentials to access the Psiphon servers from the publicly distributed binary and logs in to collect the MOTD banner. By connecting to multiple servers, the attacker identifies which systems are running an outdated version of the Linux Kernel and use this information to launch an exploit, taking full control of the system.

Short Term Solution: Modify the banner used in the SSH configuration to not return identifying information, or return false information. This is often stored in the `/etc/motd` file on most of the Psiphon servers.

Long Term Solution: Update automated build scripts to strip out sensitive information from being displayed during the login. This includes removing MOTD banners from the host for the SSH service hosted on port 22/TCP as well as the administrative SSH service hosted on port 2222/TCP.

8. Weak encryption standards for SSH

Class: Configuration **Severity:** Informational **Difficulty:** NA

Finding ID: iSEC-OTFPSI14-08

Targets: The SSH configuration of the Psiphon servers.

Description: The SSH configuration was found to support cryptographically weak MAC algorithms as well as CBC-mode ciphers. While there are no known exploits for these configurations, nor a current attack scenario where an existing SSH connection could be down-graded to use this weaker configuration, a more defense-in-depth approach would not support weak algorithms such as MD5 or less than 128-bit MAC algorithms.

```
hmac-md5
hmac-md5-96
hmac-md5-96-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-sha1-96
hmac-sha1-96-etm@openssh.com
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
blowfish-cbc
cast128-cbc
rijndael-cbc@lysator.liu.se
```

Listing 2: SSH configuration supporting MD5 algorithm, 96-bit ciphers, and CBC mode ciphers

Exploit Scenario: A nation-state adversary obtains enough computational power to exploit one of these weak encryption options resulting in the plaintext recovery of communications between the Psiphon client and the server.

Short Term Solution: Remove support in SSH and OSSH configurations for the above weak algorithms. This is a modification to each of the sshd configuration files on the Psiphon servers.[9]

Long Term Solution: Consider removing support for CBC mode ciphers, the MD5 algorithm, and any algorithm less than 128 bits in all future development practices where security is required.

[9] https://security.stackexchange.com/questions/39756/secure-configuration-of-ciphers-macs-kex-available-in-ssh

Psiphon Android Client Vulnerabilities

9. Insecure default Android browser settings

Class: Configuration **Severity:** Informational **Difficulty:** Medium

Finding ID: iSEC-OTFPSI14-09

Targets: The Psiphon Android client software.

Description: The Psiphon 3 Android application provides an optional browser for users using the third-party Zirco Browser.[10] This browser expands on the built-in Android WebView API to provide a full featured Android browser environment, complete with bookmarks, web history, and tabs. (NOTE: iSEC did not review the Zirco Browser itself.) The default configurations of this browser enable JavaScript, cache sessions such as cookies, log the browsing history, and auto-complete form fields if cached information is present. Sensitive information, such as user names and passwords, can be saved in the local cache because these features are present. In the case of a compromise of the device or the browser session, these values can be retrieved. While Psiphon deemed this outside of the threat model of the Psiphon client, disabling these settings would help mitigate the risk of sensitive information disclosure in the case of a compromise.

Exploit Scenario: The Zirco Browser is exploited via a malicious web page. This results in the session information that is cached in the data directory of the Psiphon application is recovered by the attacker, and used to log into other sites the user has visited such as Gmail or a bank.

Short Term Solution: Consider disabling sensitive features such as JavaScript, auto-fill functions, history, and bookmarks by default. While these options can be granted to some users, they should only be enabled if necessary to the user.

Long Term Solution: As the feature gap between legacy versions (3.x and below) and the current version of Android continues to expand, Psiphon should consider splitting versions of the app; one that supports legacy devices (requiring the Zirco Browser) and another that is for newer devices that support the VPN API.[11] On the legacy application, the existing configuration of the browser would suffice as it is a limitation of the OS, but the newer version should provide the Zirco Browser only if necessary, and in this case, a limited, feature stripped version.

[10] https://code.google.com/p/zirco-browser/

[11] http://developer.android.com/about/versions/android-4.0-highlights.html

Psiphon Windows Client Vulnerabilities

10. Windows client persists settings in Registry

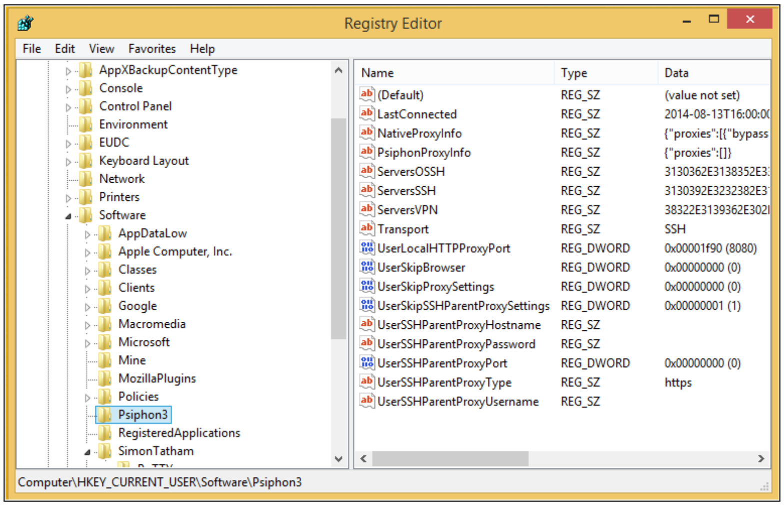
Class: Data Exposure **Severity:** Informational **Difficulty:** NA

Finding ID: iSEC-OTFPSI14-10

Targets: The following registry keys:

- \HKEY_CURRENT_USER\Software\Psiphon3
- \HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\SshHostKeys (Created by PoTTY[12])

Description: When initially running Psiphon.exe on a new host, a series of registry keys are created for the current user that hold required server list information to connect to Psiphon, as well as user settings such as system proxy server settings. In environments where a computer is shared between users, or where a user may want to hide that they have used Psiphon on that machine, the Registry will provide evidence that a given Windows user account has used Psiphon. NOTE: This issue is given a severity of informational as it was deemed outside of the Psiphon Threat Model as the software makes no attempt to hide itself when installed.



Name	Type	Data
(Default)	REG_SZ	(value not set)
LastConnected	REG_SZ	2014-08-13T16:00:0
NativeProxyInfo	REG_SZ	{"proxies":[{"bypass
PsiphonProxyInfo	REG_SZ	{"proxies":[]}
ServersOSSH	REG_SZ	3130362E3138352E3
ServersSSH	REG_SZ	3130392E3232382E3
ServersVPN	REG_SZ	38322E3139362E302
Transport	REG_SZ	SSH
UserLocalHTTPProxyPort	REG_DWORD	0x00001f90 (8080)
UserSkipBrowser	REG_DWORD	0x00000000 (0)
UserSkipProxySettings	REG_DWORD	0x00000000 (0)
UserSkipSSHParentProxySettings	REG_DWORD	0x00000001 (1)
UserSSHParentProxyHostname	REG_SZ	
UserSSHParentProxyPassword	REG_SZ	
UserSSHParentProxyPort	REG_DWORD	0x00000000 (0)
UserSSHParentProxyType	REG_SZ	https
UserSSHParentProxyUsername	REG_SZ	

Computer\HKEY_CURRENT_USER\Software\Psiphon3

Figure 2: Psiphon registry entries

Exploit Scenario: A user makes use of Psiphon where it is against company policy to do so. Although the user has deleted the program, the company's IT staff checks the registry and determines that the user has indeed executed the Psiphon application.

Short Term Solution: Make use of the Win32 `GetDriveType`[13] API call to determine if Psiphon is running from local or removable media. If running from removable media, do not make registry changes but instead use a configuration file that sits beside the Psiphon executable.

Long Term Solution: Make the configuration file stored in the text region of a second executable. This second executable could be the same as the "updater" binary as described in the long term solution of finding 11 on the following page. The advantage of this configuration would be that while Psiphon is running there would only be two files – the main Psiphon executable and the updater/config file. When the main application is closed, the updater/config file could rewrite itself into the main Psiphon executable and then delete itself. This would mean that a single executable file would always contain the most up-to-date Psiphon client, server list, and settings.

[12] http://www.mrhinkydink.com/potty.htm
[13] http://msdn.microsoft.com/en-us/library/windows/desktop/aa364939(v=vs.85).aspx

11. Windows client update script unreliable

Class: Patching **Severity:** Informational **Difficulty:** NA

Finding ID: iSEC-OTFPSI14-11

Targets: The `Connectionmanager.cpp:PaveUpgrade()` class of the Windows client binary.

Description: During testing, some crashes were observed in the Psiphon3.exe application when it attempted to automatically update itself. Code review of the `PaveUpgrade` and `ConnectionManagerUpgradeThread` methods of `connectionmanager.cpp` show that the update mechanism does not launch a new process to overwrite the Psiphon 3 executable, but rather uses a new thread. Sometimes Windows will not allow the executable to be overwritten, or some threads will still be open from the original process. Comments note that Windows does not allow an executable to be overwritten while it is running.

One side effect of Psiphon crashes is that system proxy settings that have been manipulated by Psiphon are not automatically restored, leaving the user without Internet connectivity in most circumstances. It is not a given that a user will restart Psiphon to reset their settings if their Internet connection is not working.

Short Term Solution: Spawn yet another thread to act as a watchdog for Psiphon. If the application crashes, be sure to restore the user's original proxy settings.

Long Term Solution: Package the Psiphon update in an updater executable that gets distributed via the existing update distribution network. The main executable can continue to download and signature verify the update package as it does now. However, instead of trying to replace itself, the Psiphon binary can simply execute the updater. The updater should then replace the Psiphon3.exe binary after monitoring that the process has been gracefully terminated. Once the updated version of Psiphon has been installed, it can signal the updater that the update is complete and then remove the updater binary.

Appendices

A Improvements to Host Management

Throughout the testing period, iSEC consultants identified areas where enterprise-class management tools would potentially improve the overall security posture of Psiphon hosts, servers, and infrastructure. The areas below are recommendations to help give Psiphon engineers fine-grain control of their systems as it continues to expand in size.

A.1 Log Aggregation and SIEM

A Security Information and Event Monitoring (SIEM) solution helps provide control over the events that happen on a network. As Psiphon grows, and the number of hosts to manage increases, a SIEM would help reduce the amount of effort to maintain a secure infrastructure. A variety of features are normally provided in a SIEM, but specifically the Log Aggregation features would provide a high value for the Psiphon project.

Psiphon hosts normally will use a custom `syslog` configuration to redirect pertinent logs to a flat log file under `/var/log/PsiphonV.log`. An automation server regularly goes out and collects these logs and then imports them into a PostgreSQL database. This provides diagnostic information to help engineers maintain perspective on the network.

On the the Psiphon Hosts that were provided, each were actively defended by a `fail2ban` tool that protected services from attacks like brute forcing. This is a standard way of defending SSH and OSSH services from attacks, but it lacked any type of alerting mechanism to let engineers know when active attacks were occurring on these systems.

A log aggregator/SIEM such as Splunk could provide engineers with more insight about active attacks happening on the network, integrate into the existing logging mechanism, and allow engineers to respond to attacks with lower latency. It is recommended that Psiphon host configurations be modified so that `syslog` output is copied or redirected to a centralized `syslog` server hosted by Psiphon. From there, the logs can be processed and compiled into a report to give insight about current issues on the network.

Alternatively, if these solutions are cost prohibitive, it may be possible to expand the current logging initiatives to include more information about the hosts. For instance, expanding the automation server sync tools so that it not only downloads the `PsiphonV.log` file, but also the `fail2ban` and other security related logs, could perform a similar function to a log aggregator solution.

B Authorization and Authentication Controls

The Psiphon team granted iSEC access to a few servers via SSH. This was done by either providing the root password to the server, or a private key that could be used for authentication. The following sections outline improvements that would occur if a centralized authentication system (such as LDAP) were in place. The goal of these improvements may not have a direct affect on the security of the organization at its current size, but as Psiphon grows to include more developers, more administrators, and more servers, this centralized authentication would make it easier to scale.

B.1 Psiops Database Management

One point of improvement identified was related to password and key material management. Currently, a lot of sensitive material is maintained in the `psi_ops` database, which, in its encrypted form, is managed by Ciphershare.[14] The Psiops Database stores sensitive information such as the key material used to establish secure communications between a Psiphon client and server, root credentials for Psiphon servers, and API keys for hosting providers. The file is a serialized object containing the most sensitive information about the project/organization. This file is normally stored encrypted using Ciphershare document management system, but on some systems a portion is stored in the clear relying on the security of the host to protect it.

As this is a centralized point of failure and a primary attack vector, the utmost care should be given when using this including restricting file access, encrypting it at rest, and ensuring that it is only transferred between systems securely. As Psiphon continues to grow its code base, servers, and potentially its project members, it is important that the solution to manage this sensitive information scales to the size of the organization. Services like LDAP may provide a way of authenticating users and deciding whether they should receive access to the file. This would allow managed control of the file as people enter and leave the organization over time.

B.2 Remote Administration

Remote administration is done using a separate SSH process on the Psiphon servers. This is used by administrators to configure the host and by scripts to automatically connect and download server logs. To scale to an enterprise-ready environment, Psiphon should consider replacing local root authentication, with a network authentication service like LDAP. This would allow various Psiphon members to log in with their own credentials. This provides a variety of improvements including:

- A way of tracking who logged in and when for auditing purposes
- Ability to control what servers a user should have access
- Ability to provide temporary access to users and revoke those privileges at a later time
- Cut down on the amount of effort required to revoke access

[14] http://www.provensecuritysolutions.com/

C Host Hardening

Throughout the hosts provided for testing (Amazon AMI, Ubuntu, Debian), inconsistencies were identified relating to security practices of the hosts. iSEC reviewed the host security of one of the Psiphon server images, and found missing patches, lack of process isolation, and processes running with root privileges. There were also good security practices noted: the systems all seemed to implement ASLR and had very strict `iptables` rules. Psiphon should continue to invest time to harden their systems.[15]

Areas to focus on in the future are:

- Ensuring processes run with least-privilege
- Securing SSH to only allow key authentication
- Removing unnecessary applications or restricting access to sensitive tools
- Restricting read/write access to Psiphon files and logs
- Remove banners or other identifying information from the system. (e.g. MOTD)

[15] https://wiki.debian.org/Hardening