



PROFILE

CVE-2020-7247

ZEROFOX INTELLIGENCE

P-2022-10-25a

Classification: TLP:GREEN

Criticality: High

Intelligence Requirements: Exploits & Vulnerabilities

October 25, 2022



PROFILE: CVE-2020-7247

P-2022-10-25a

TLP:GREEN

Created on: October 10, 2022

Intelligence Cut-off: 4:30 PM (EDT) on Oct. 7, 2022

First Observed: 1/28/2020

CVE Dictionary Entry: CVE-2020-7247

NVD Published Date: 1/29/2020

NVD Last Modified: 4/29/2020

Severity: HIGH

Base Score: 9.8

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Metasploit Available: YES

Current Description

When used in OpenBSD 6.6 and other products, *smtp_mailaddr* in *smtp_session.c* in OpenSMTPD 6.6.1 allows remote attackers to execute arbitrary commands as root via a crafted SMTP session, as demonstrated by shell metacharacters in a MAIL FROM field. This affects the "uncommented" default configuration. The issue exists because of an incorrect return value upon failure of input validation.

Known Affected Software Configurations

Configuration	Common Platform Enumeration
1	cpe:2.3:a:openbsd:opensmtpd:6.6:*****
2	cpe:2.3:o:debian:debian_linux:9.0:***** cpe:2.3:o:debian:debian_linux:10.0:*****
3	cpe:2.3:o:fedoraproject:fedora:32:*****
4	cpe:2.3:o:canonical:ubuntu_linux:18.04:****lts:*** cpe:2.3:o:canonical:ubuntu_linux:19.10:*****



Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-78	Improper Neutralization of Special Elements used in an OS Command (OS Command Injection)	NIST
CWE-755	Improper Handling of Exceptional Conditions	NIST

Vulnerability Analysis

CVE-2020-7247 Remote Code Execution (RCE) and Local Privilege Exploitation (LPE) was discovered by Qualys within OpenSMTPD 6.6.1p1. This vulnerability was patched on Jan. 28, 2020, within version 6.6.2p1. This security patch applies updates to five different files (Figure 1).

```
diff --git a/smtpd/mail.lmtp.c b/smtpd/mail.lmtp.c
diff --git a/smtpd/mail.mboxfile.c b/smtpd/mail.mboxfile.c
diff --git a/smtpd/mta_session.c b/smtpd/mta_session.c
diff --git a/smtpd/smtp_session.c b/smtpd/smtp_session.c
diff --git a/smtpd/smtpd.h b/smtpd/smtpd.h
```

Figure 1. Files Changed During Software Patch

source: <https://github.com/OpenSMTPD/OpenSMTPD>

Of the file changes, the most important changes were located in *smtp_sessions.c* within the *smtp_mailaddr* function. This function attempts to validate all addresses within a particular email communication sent. If *smtp_mailaddr* determines that any of these email addresses are not valid, the email message will not be processed.

Analyzing the applied patch to *smtp_mailaddr* as shown in Figure 2 provides insight into the inherent issue. In the original function, the first condition checks to see if both the “user” part and the “domain” part are valid within a given email. Failing these checks will enter the body of this condition.

Within the body, as long as the user part of an email address is not empty, then code will continue to execute without returning a False condition (Figure 3). The patch provided, represented by the lines with a plus (+) sign, correctly checks for invalid user format and will correctly return zero (Figure 2).



```
--- a/smtpd/smtp_session.c
+++ b/smtpd/smtp_session.c
@@ -2224,25 +2224,23 @@ smtp_mailaddr(struct mailaddr *maddr, char *line, int mailfrom, char **args,
    memmove(maddr->user, p, strlen(p) + 1);
}

- if (!valid_localpart(maddr->user) ||
-     !valid_domainpart(maddr->domain)) {
-     /* accept empty return-path in MAIL FROM, required for bounces */
-     if (mailfrom && maddr->user[0] == '\0' && maddr->domain[0] == '\0')
-         return (1);
-
-     /* no user-part, reject */
-     if (maddr->user[0] == '\0')
-         return (0);
-
-     /* no domain, local user */
-     if (maddr->domain[0] == '\0') {
-         (void)strncpy(maddr->domain, domain,
-             sizeof(maddr->domain));
-         return (1);
-     }
-     /* accept empty return-path in MAIL FROM, required for bounces */
-     if (mailfrom && maddr->user[0] == '\0' && maddr->domain[0] == '\0')
-         return (1);
-
-     /* no or invalid user-part, reject */
-     if (maddr->user[0] == '\0' || !valid_localpart(maddr->user))
-         return (0);
-
-     /* no domain part, local user */
-     if (maddr->domain[0] == '\0') {
-         (void)strncpy(maddr->domain, domain,
-             sizeof(maddr->domain));
-     }
-     if (!valid_domainpart(maddr->domain))
-         return (0);
-
-     return (1);
- }
```

Figure 2. OpenSMTPD 6.6.2p1 Patch

Source: [hXXps://github.com/OpenSMTPD/OpenSMTPD](https://github.com/OpenSMTPD/OpenSMTPD)

```
Breakpoint 1, 0x000007fce2fcbfc4 in ?? () Date:
calling valid_localpart
0x7ff83ca2d58: ";for P in I s M i k v z z I w V a p j;do read P;done;sh;exit 0;"

Breakpoint 2, 0x000007fce2fcbfc9 in ?? ()
ret of valid_localpart
rax                0x0    0

Breakpoint 3, 0x000007fce2fcbf57 in ?? ()
ret of smtp_mailaddr
rax                0x1    1
```

Figure 3. Smtplib Vulnerability

Source: ZeroFox Intelligence

The consequence of this vulnerability is the continuation of processing an invalid user string that contains special characters, which will be interpreted eventually. Comparing Figure 3 with Figure 4, note that the string being validated by *valid_localpart* returns a False condition of zero, which should have caused *smtp_mailaddr* to return a False condition of zero as well. Since *smtp_mailaddr* returned a True statement, continuing processing, the user string that contains a special character (;) becomes present within a string that will be interpreted by */bin/sh*. This “;”



informs the shell that two commands are within this string, treating the user part of an email address as a legitimate bash command to execute.

```
Breakpoint 4, 0x0000cf1c2499e90 in ?? ()
execle triggered
0xcefbc075c9e: "/bin/sh"
0xcefbc075c9e: "/bin/sh"
0xcefbc07aac0: "-c"
0x7f7ffffde770: "/usr/local/libexec/smtpd/mail.local -f ;for P in I s M i k v z z I w V a p j;do read P;done;sh;exit 0;@victim.warface.org victim"
```

Figure 4. OS Injection Vulnerability

Source: ZeroFox Intelligence

Risk

CVE-2020-7247 is a RCE that can also be leveraged as a LPE. The impact of this vulnerability is the threat actor's increased ability to gain a secure foothold within a corporate network. By compromising a mailing server, a threat actor will have privileged access to all emails—along with any credentials stored locally on the affected mailing server. Threat actors will often reuse these credentials to gain unauthorized access to other devices within a corporate network.

The likelihood of a threat actor abusing this vulnerability is extremely high. This vulnerability does have a multitude of Proof Of Concept code available on Github. Additionally, a Metasploit Module of this vulnerability has been published.

Recommendation

Upgrade to the latest version of OpenSMTPD.



Appendix 1: Sigma Rules

```
title: OpenSMTPd CVE-2020-7247 Exploit
id: a60dfa67-4ebf-43a2-899b-c47f9d10cb0d
status: experimental
description: Detection of OpenSMTPd CVE-2020-7247 Exploit attempt within FileBeat logs
author: ZeroFox
date: 2022/10/3
modified: 2022/10/3
references:
  - https://www.rapid7.com/db/vulnerabilities/opensmtpd-cve-2020-7247-rce/
  - https://nvd.nist.gov/vuln/detail/CVE-2020-7247
  - https://www.exploit-db.com/exploits/47984
logsource:
  product: linux
tags:
  - cve.2020.7247
  - mitre.T1190
  - mitre.T1068
detection:
  selection:
    - event.original.re: '.*smtp.*from=.*;.*to=.*'
  condition: selection
level: critical
```

Sigma Detection Rule for CVE-2020-7247: Inx_opensmtpd_cve_2020_7247.yml

Source: ZeroFox Intelligence



Appendix 2: Traffic Light Protocol for Information Dissemination

	Red	Amber
WHEN SHOULD IT BE USED?	Sources may use TLP:RED when information cannot be effectively acted upon by additional parties and could lead to impacts on a party's privacy, reputation, or operations if misused.	Sources may use TLP:AMBER when information requires support to be effectively acted upon but carries risks to privacy, reputation, or operations if shared outside of the organizations involved.
HOW MAY IT BE SHARED?	Recipients may NOT share TLP:RED with any parties outside of the specific exchange, meeting, or conversation in which it is originally disclosed.	Recipients may ONLY share TLP:AMBER information with members of their own organization and its clients, but only on a need-to-know basis to protect their organization and its clients and prevent further harm. Note that TLP:AMBER+STRICT restricts sharing to the organization only.
	Green	Clear
WHEN SHOULD IT BE USED?	Sources may use TLP:GREEN when information is useful for the awareness of all participating organizations, as well as with peers within the broader community or sector.	Sources may use TLP:CLEAR when information carries minimal or no risk of misuse in accordance with applicable rules and procedures for public release.
HOW MAY IT BE SHARED?	Recipients may share TLP:GREEN information with peers and partner organizations within their sector or community but not via publicly accessible channels.	Recipients may share TLP:CLEAR information without restriction, subject to copyright controls.