CVE-2020-8794

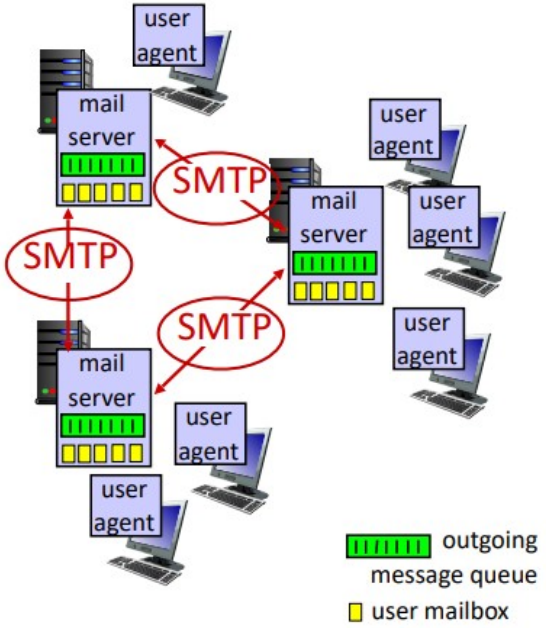
MAC0352 - Redes de Computadores e Sistemas Distribuídos
Instituto de Matemática e Estatística · USP

Arthur Font - 12036152
Lucas Pires - 10723624

Protocolo SMTP (Simple Mail Transfer Protocol)

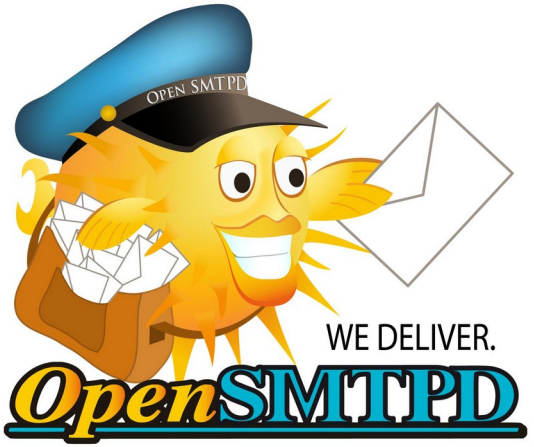
SMTP: envio e recebimento de emails

- Mail Server:
  - Caixa de entrada
  - Fila de envio
- User Agent

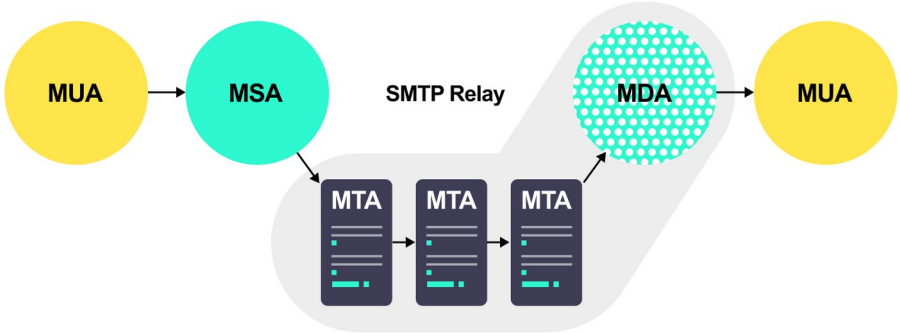


Software alvo: OpenSMTPD

- Daemon que implementa o protocolo SMTP
- Open Source
- Parte do projeto OpenBSD



MTA e MDA



- MTA (Message Transfer Age
  - Recebimento de emails remotos
  - Envio a outros servidores de email
- MDA (Message Delivery Agent):
  - Recebimento emails locais
  - Entrega para a caixa de entrada

Vulnerabilidade - Contextualização

- Envelope OpenSMTPD:
  - Descreve internamente o email recebido
  - Campos de “chave: valor\n”
  - Possível injetar comandos arbitrários que serão executados pelo MDA como super usuário

- Causa: Buffer overflow no MTA

Vulnerabilidade: mta_session.c

```
/* continuation reply, we parse out the repeating statuses and ESC */
if (cont) {
        if (s->replybuf[0] == '\0')
                (void)strlcat(s->replybuf, line, sizeof s->replybuf);
        else {
                line = line + 4;
                if (isdigit((int)*line) && *(line + 1) == '.' &&
                    isdigit((int)*line+2) && *(line + 3) == '.' &&
                    isdigit((int)*line+4) && isspace((int)*(line + 5)))
                        (void)strlcat(s->replybuf, line+5, sizeof s->replybuf);
                else
                        (void)strlcat(s->replybuf, line, sizeof s->replybuf);
        }
        goto nextline;
}

/* last line of a reply, check if we're on a continuation to parse out status and ESC.
 * if we overflow reply buffer or are not on continuation, log entire last line.
 */
if (s->replybuf[0] != '\0') {
        p = line + 4;
        if (isdigit((int)*p) && *(p + 1) == '.' &&
            isdigit((int)*p+2) && *(p + 3) == '.' &&
            isdigit((int)*p+4) && isspace((int)*(p + 5)))
                p += 5;
        if (strlcat(s->replybuf, p, sizeof s->replybuf) >= sizeof s->replybuf)
                (void)strlcpy(s->replybuf, line, sizeof s->replybuf);
}
else
        (void)strlcpy(s->replybuf, line, sizeof s->replybuf);
```

Exploit

- Client-side:
  - OpenSMTPD conecta ao servidor de email (*attacker*)
  - Servidor responde com uma mensagem de erro maliciosa
  - Buffer overflow no MTA do cliente permite escrita no envelope
  - MDA executa comandos como super usuário

Exploit

- Server-side:
  - O cliente (*attacker*) se conecta e envia uma mensagem que exija um *bounce* e aguarda a resposta do servidor OpenSMTPD
  - O cliente responde com um erro temporário (4xx) contendo a mensagem maliciosa
  - O cliente força um *crash* do OpenSMTPD e assim que ele reiniciar será executado o comando malicioso

Patch

```
/* continuation reply, we parse out the repeating statuses and ESC */
if (cont) {
    if (s->replybuf[0] == '\0')
        (void)strlcat(s->replybuf, line, sizeof s->replybuf);
    else {
        line = line + 4;
        if (isdigit((int)*line) && *(line + 1) == '.' &&
            isdigit((int)*line+2) && *(line + 3) == '.' &&
            isdigit((int)*line+4) && isspace((int)*(line + 5)))
            (void)strlcat(s->replybuf, line+5, sizeof s->replybuf);
        else
            (void)strlcat(s->replybuf, line, sizeof s->replybuf);
    }
    goto nextline;
}

/* last line of a reply, check if we're on a continuation to parse out status and ESC.
 * if we overflow reply buffer or are not on continuation, log entire last line.
 */
if (s->replybuf[0] != '\0') {
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
    p = line + 4;
    if (isdigit((int)*p) && *(p + 1) == '.' &&
        isdigit((int)*p+2) && *(p + 3) == '.' &&
        isdigit((int)*p+4) && isspace((int)*(p + 5)))
        p += 5;
    if (strlcat(s->replybuf, p, sizeof s->replybuf) >= sizeof s->replybuf)
        (void)strlcpy(s->replybuf, line, sizeof s->replybuf);
}
else
    (void)strlcpy(s->replybuf, line, sizeof s->replybuf);
```

```
1293        /* continuation reply, we parse out the repeating statuses and ESC */
1294        if (cont) {
1295            if (s->replybuf[0] == '\0')
1296                (void)strlcat(s->replybuf, line, sizeof s->replybuf);
1297            else if (len > 4) {
1298                line = line + 4;
1299                if (isdigit((int)*line) && *(line + 1) == '.' &&
1300                    isdigit((int)*line+2) && *(line + 3) == '.' &&
1301                    isdigit((int)*line+4) && isspace((int)*(line + 5)))
1302                    (void)strlcat(s->replybuf, line+5, sizeof s->replybuf);
1303                else
1304                    (void)strlcat(s->replybuf, line, sizeof s->replybuf);
1305            }
1306            goto nextline;
1307        }
1308
1309        /* last line of a reply, check if we're on a continuation to parse out status and ESC.
1310         * if we overflow reply buffer or are not on continuation, log entire last line.
1311         */
1312        if (s->replybuf[0] == '\0')
1313            (void)strlcat(s->replybuf, line, sizeof s->replybuf);
1314        else if (len > 4) {
1315            p = line + 4;
1316            if (isdigit((int)*p) && *(p + 1) == '.' &&
1317                isdigit((int)*p+2) && *(p + 3) == '.' &&
1318                isdigit((int)*p+4) && isspace((int)*(p + 5)))
1319                p += 5;
1320            if (strlcat(s->replybuf, p, sizeof s->replybuf) >= sizeof s->replybuf)
1321                (void)strlcpy(s->replybuf, line, sizeof s->replybuf);
1322        }
    ----------------------------------------------------------------------------
    ----------------------------------------------------------------------------
```

Ambiente utilizado

- OS:
  - VM - OpenBSD
  - HOST - Arch Linux
- Ferramentas:
  - Wireshark
  - VirtualBox
  - Git

Referências

1. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8794
2. https://www.openwall.com/lists/oss-security/2020/02/26/1
3. https://www.exploit-db.com/exploits/48140
4. https://github.com/OpenSMTPD/OpenSMTPD/commit/3a7096cdc252185eaf30552987aedc988b9cde6d
5. https://en.wikipedia.org/wiki/OpenSMTPD

Obrigado pela atenção!