

Overlord RAT · HTML Smuggling · Process Hollowing · HVNC

1. Resumen Ejecutivo



ColCERT identificó y analizó una campaña activa de spear-phishing dirigida a funcionarios de entidades públicas colombianas, atribuida con un alto nivel de confianza al grupo BlindEagle (APT-C-36). El vector de entrada es un correo electrónico con señuelo de temática judicial enviado desde una cuenta institucional .gov.co comprometida. El adjunto es un archivo .xhtml (Extensible HyperText Markup Language, un formato de página web ejecutable por navegadores), que implementa la técnica HTML Smuggling para ensamblar y ejecutar una cadena de ataque de ocho etapas directamente en memoria del navegador.

El implante final, Overlord RAT, es un troyano de acceso remoto de alta complejidad que proporciona al atacante control total y sigiloso del equipo comprometido, sus capacidades incluyen escritorio remoto oculto (HVNC), captura de pantalla, registro de pulsaciones, robo de cookies y credenciales de navegadores, así como la ejecución arbitraria de comandos. Además, se instala en la memoria del proceso legítimo RegSvc.exe mediante la técnica Process Hollowing, utilizando llamadas directas al kernel (syscalls) para evadir la mayoría de EDR convencionales.

La validación dinámica en la plataforma de análisis de malware del ColCERT (<https://detectic.colcert.gov.co/> - (23/09/2026)) confirmó los tres artefactos como Malicioso (peso máximo 200/200, 136 indicadores activos en el implante), asimismo, este análisis permitió revelar una dirección IP de Comando y Control (C2) adicional que no estaba registrada previamente: 199.16.199.2.

Hallazgos clave

- ❑ Tres artefactos confirmados maliciosos en sandbox dinámico DetecTIC.
- ❑ Nueva IP de C2 identificada: 199[.]16[.]199[.]2 (puerto 4782/TCP, beacon xGKGL-4447DF1C).
- ❑ Process hollowing sobre RegSvc.exe mediante NT direct syscalls — evasión avanzada de EDR.
- ❑ HVNC activo: el atacante opera el equipo víctima con escritorio completamente oculto.
- ❑ Persiste vía registro (Run key) y carpeta de Inicio (.vbs) — sobrevive reinicios.
- ❑ Infraestructura compartida con Quasar RAT (dominio qsar20.duckdns.org, puerto 6001/TCP).

¿Cómo actúa el malware?

Campaña de phishing con falsa notificación judicial - troyano Overlord RAT con escritorio oculto (HVNC)



Figura 1. Mecanismo de acción del malware — Campaña BlindEagle/APT-C-36 COLCERT-INC-2026-0922

2. Actor de amenaza – BlindEagle / APT-C-36

BlindEagle, también conocido como APT-C-36 y documentado por múltiples proveedores de inteligencia de amenazas, es un actor de amenaza persistente avanzada con foco geográfico en Colombia, Ecuador, Chile y América Latina en general. Sus objetivos primarios son entidades gubernamentales, instituciones financieras y organismos de seguridad nacional.

Atributo	Valor
Nombre del actor	BlindEagle / APT-C-36 / AguilaCiega
Origen probable	América Latina (atribución con confianza media)
Motivación	Espionaje, robo de credenciales, acceso persistente
Sectores objetivo	Gobierno, finanzas, telecomunicaciones, seguridad
Geografía	Colombia (principal), Ecuador, Chile, Perú
TTPs característicos	Spear-phishing, HTML Smuggling, RATs abiertos/semipúblicos
Infraestructura	DNS dinámico (duckdns.org, no-ip.com), VPS en América Latina
RATs usados históricamente	Quasar RAT, AsyncRAT, NjRAT, RemcosRAT, Overlord RAT
Nivel de sofisticación	Medio-alto (combina herramientas públicas con técnicas avanzadas)

2.1 Infraestructura compartida identificada

Se identificaron indicadores que vinculan esta campaña con infraestructura Quasar RAT activa simultáneamente, sugiriendo que el actor opera múltiples accesos concurrentes o reutiliza servidores C2:

Indicador	Protocolo / Puerto	RAT asociado	Acción
envio14-9[.]duckdns[.]org	4782/TCP	Overlord RAT	Bloquear
qsar20[.]duckdns[.]org	6001/TCP	Quasar RAT	Bloquear
181[.]235[.]13[.]254	DNS dinámico / múltiple	Compartida	Bloquear
199[.]16[.]199[.]2	4782/TCP	Overlord RAT (C2 adicional)	Bloquear — nuevo

3. Análisis técnico – cadena de ataque (8 etapas)



La cadena de infección está completamente embebida en un único archivo .xhtml (SVG con JavaScript). Ninguna etapa descarga contenido externo; todo viaja comprimido, cifrado y codificado dentro del adjunto inicial.

3.1 Etapa 1 – Correo señuelo (spear-phishing)

Vector inicial: correo con remitente de cuenta institucional .gov.co comprometida. Asunto de temática judicial — "Traslado de demanda" — con referencia THG2026. El adjunto se presenta con nombre del tipo:

armenia 10112026 MIN-pasto.xhtmll

El nombre simula una citación de juzgado, aumentando la credibilidad y el formato .xhtml pasa desapercibido ante usuarios que no distinguen extensiones.

3.2 Etapa 2 – Contenedor SVG con HTML Smuggling

El .xhtml es una imagen SVG que contiene un blob JavaScript de ~26 KB codificado en Base64. Al abrir el archivo en cualquier navegador, el script:

- ☐ Decodifica el blob Base64 y lo materializa como objeto Blob URL en memoria.
- ☐ Crea un elemento <a> oculto con atributo download y lo "dispara" automáticamente.
- ☐ El navegador descarga un ZIP de ~5 MB con nombre aleatorio a la carpeta de Descargas.
- ☐ El ZIP está protegido con contraseña visible en el cuerpo del correo: THG2026.

Característica polimórfica: el ZIP se regenera en cada entrega con nombre diferente para evadir detección por hash. El hash del ZIP no sirve como IOC.

3.3 Etapa 3-4 – Dropper JavaScript y búsqueda en disco

Dentro del ZIP, un script JavaScript (etapa 3) es el motor central de la cadena. Al ejecutarlo la víctima (doble clic o wscript.exe), realiza:

- ☐ Búsqueda del .xhtml original en disco (etapa 4) usando APIs de Windows Script Host para localizar el contenedor por nombre de patrón.
- ☐ Extracción en memoria del blob XOR-ofuscado con clave 0xF4 y re-cifrado AES-256-ECB que contiene las etapas restantes.
- ☐ Validación dinámica DetecTIC (sandbox): wscript.exe (PID variable) crea artefactos temporales C:\Windows\Temp\pid_NNNN_seq_1.js.pe — evidencia forense de extracción de PE en memoria.
- ☐ Firma DetecTIC detectada: CustomPolicy.MVX.65056.JSObfuscationFunction — veredicto Custom Riskware (el dropper no es el payload, de ahí el nivel inferior).

Archivo temporal: C:\Windows\Temp\pid_1996_seq_1.js.pe (Win10x64)

Archivo temporal: C:\WINDOWS\Temp\pid_3932_seq_1.js.pe (WinXP)

Firma DetecTIC detectada: CustomPolicy.MVX.65056.JSObfuscationFunction — veredicto Custom Riskware (el dropper no es el payload, de ahí el nivel inferior).

3.4 Etapa 5-6 – VBScript con persistencia y señuelo PDF

El JavaScript extrae y ejecuta un VBScript (~3.4 MB, obfuscado, MD5: 9a49c0adc1ea3c671b243f018a376ab1). El VBScript realiza cuatro acciones en paralelo:

- ☐ **Persistencia #1 — Run key:** escribe una entrada en HKCU\Software\Microsoft\Windows\CurrentVersion\Run apuntando a sí mismo con nombre disfrazado (p. ej. NetGuard*.vbs).
- ☐ **Persistence #2 — Startup folder:** copia el .vbs a %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\.
- ☐ **Señuelo:** muestra un PDF falso del Juzgado de Pitalito para que la víctima no sospeche.
- ☐ Ejecuta PowerShell cifrado (etapa 7) con parámetros -EncodedCommand para evadir inspección de línea de comandos.
- ☐ Se autoborra al completar la cadena.

3.5 Etapa 7 – PowerShell descifrado AES-256-ECB

El PowerShell descifrado (script.ps1, ~2.5 MB, MD5: 44217608a648689fdc5b126a9dfbd941) actúa como loader final:

- ☐ Descifra el payload PE (Overlord RAT) embebido en sí, mismo usando AES-256-ECB.
- ☐ Inyecta el PE en memoria mediante process hollowing — NO escribe el ejecutable en disco.
- ☐ **Target del hueco:** RegSvcs.exe (proceso legítimo de .NET Framework).

3.6 Etapa 8 – Process hollowing con NT direct syscalls

Para evadir EDR y anti-hooking, el loader NO usa las funciones estándar de la Win32 API (CreateRemoteThread, VirtualAllocEx, etc.). En su lugar invoca directamente las funciones NT del kernel:

NtCreateSection → crea sección de memoria compartida
 NtMapViewOfSection → mapea el PE del RAT en el espacio de RegSvcs.exe
 NtSetContextThread → redirige el punto de entrada
 NtResumeThread → reanuda la ejecución del proceso hueco

Este patrón evita que los hooks de userland de los EDR (que interceptan las funciones de kernel32/ntdll vía IAT patching o inline hooking) puedan detectar la inyección. Solo soluciones con kernel callbacks o ETW pueden capturarla.

Árbol de procesos confirmado en sandbox DetecTIC:

```
explorer.exe (PID 1244)
├─ etapa8_implante.bin.exe (PID 3024) ← implante, primera instancia
│   └─ etapa8_implante.bin.exe (PID 700) ← auto-copia (persistencia de proceso)
│       └─ WerFault.exe ← crash reporting (comportamiento normal post-hollowing)
```

3.7 Impacto final – Overlor RAT

El implante (2,011,136 bytes, PE x64, MD5: 243e2bef112b584b9a94adabb033ed5c) es Overlord RAT — fork de Quasar RAT con capacidades HVNC añadidas. Capacidades confirmadas:

Capacidad	Descripción técnica
HVNC (Hidden VNC)	Escritorio remoto completamente invisible para la víctima — el atacante opera una sesión de Windows separada
Keylogger	Captura todas las pulsaciones de teclado en tiempo real
Robo de credenciales	Extrae cookies, contraseñas y tokens de Chrome, Edge, Brave, Opera, Firefox
Captura de pantalla	Capturas periódicas del escritorio visible de la víctima
Shell remota	Ejecución arbitraria de comandos (cmd.exe / PowerShell)
Gestor de archivos	Navegación, descarga y subida de archivos del sistema víctima
Grabación de audio	Captura de micrófono
Proxy SOCKS	El equipo comprometido actúa como proxy para ataques posteriores
Beacon C2	Puerto 4782/TCP, cabecera mágica: xGKGL-4447DF1C; intervalo de reconexión configurable

Comunicación C2 — flujo de red confirmado en sandbox:

etapa8_implante.bin.exe → 199[.]16[.]199[.]2:4782/TCP
 beacon header: xGKGL-4447DF1C
 Dominio primario: envio14-9[.]duckdns[.]org → 181[.]235[.]13[.]254:4782/TCP

Mapecto MITRE ATT&CK

Táctica	Técnica / Sub-técnica	Descripción en esta campaña
Initial Access	T1566.001 — Spear-phishing con adjunto	Correo con .xhtml adjunto desde cuenta .gov.co comprometida
Execution	T1059.007 — JavaScript	Dropper JS ejecutado vía wscript.exe
Execution	T1059.005 — VBScript	VBScript de persistencia y carga del loader PS
Execution	T1059.001 — PowerShell	Loader PowerShell con EncodedCommand
Defense Evasion	T1027 — Archivos ofuscados	XOR 0xF4 + AES-256-ECB + Base64 multicapa
Defense Evasion	T1140 — Deobfuscate/Decode	Descifrado en memoria, sin tocar disco
Defense Evasion	T1036.005 — Nombre de recurso legítimo	Proceso hollowing en RegSvcs.exe
Defense Evasion	T1055.012 — Process Hollowing	Inyección en RegSvcs.exe via NT syscalls directas
Defense Evasion	T1562.001 — Impair Defenses	NT direct syscalls para evadir hooks de EDR
Defense Evasion	T1027.006 — HTML Smuggling	ZIP ensamblado en memoria del navegador desde SVG/JS
Persistence	T1547.001 — Registry Run Keys	HKCU\...\Run con nombre NetGuard*.vbs
Persistence	T1547.001 — Startup Folder	Copia .vbs en %APPDATA%\...\Startup\
Collection	T1056.001 — Keylogging	Captura de todas las pulsaciones
Collection	T1539 — Steal Web Session Cookie	Robo de cookies de Chrome, Edge, Firefox, Opera
Collection	T1113 — Screen Capture	Capturas periódicas del escritorio
Command & Control	T1571 — Non-Standard Port	Puerto 4782/TCP para C2
Command & Control	T1568.002 — DNS Dinámico	duckdns.org como infraestructura C2
Command & Control	T1095 — Non-Application Layer Protocol	Beacon binario con cabecera xGKGL-4447DF1C

5. Indicadores de compromiso (IOC)

5.1 Red

Tipo	Indicador	Descripción	Acción
Domini o	envio14-9[.]duckdns[.]org	C2 primario Overlord RAT	Bloquear
Domini o	qsar20[.]duckdns[.]org	C2 Quasar RAT (infraestr. compartida)	Bloquear
IP	181[.]235[.]13[.]254	Resolución DNS dinámica — compartida	Bloquear
IP	199[.]16[.]199[.]2	C2 adicional implante (nuevo — DetecTIC 23/09)	Bloquear
Puerto	4782/TCP	Puerto C2 Overlord RAT	Bloquear salida
Puerto	6001/TCP	Puerto C2 Quasar RAT (infraestr. compartida)	Bloquear salida
Beacon	xGKGL-4447DF1C	Cabecera mágica del protocolo Overlord RAT	Firma IDS/NSM

5.2 Hashes de artefactos

ID	Artefacto	MD5	SHA-256 (primeros 40 hex)	Veredicto
E-01	armenia 10112026 MIN-pasto.xhtml	7efcca2c4d215cd015d2e54006ce48a9937c4cb2d44b5d7e8ab14fb	7efcca2c...fbcece8cf...	Malicious
D-02	etapa3_dropper.js (JS)	3f6c75ff05e8b316e6f8783f79fadeeb	722ec4b59b58dedf799c6c342f032f45	Custom Riskware (DetecTICix)
D-03	VBScrip t~WRFZY7YYkkd.tmp	9a49c0adc1ea3c671b243f018a376ab1	406ebd1276fd0e8396bdde037cd2788a	Malicious
D-04	script.ps1 (PowerShell descifrado)	44217608a648689fdc5b126a9dfbd941	4f0234ea858fe2babb68eb0285fa0998	Malicious
D-05	etapa8_implante.bin (Overlord RAT)	243e2bef112b584b9a94adabb033ed5c	5b8a9114512f1fb26e080c91e17e6976	Malicious 200/200 (DetecTIC)
D-06	recurso_101.bin / HVNCInjection.x64.dll	1c9b0e74a479617570bc33f32d2d6b55	c439e8a6648302324b0eb8f20c52cb64	Malicious GTI (DetecTIC)

5.3 Host - artefactos y rutas

Tipo	Indicador	Significado
Proceso	RegSvcs.exe con conexión 4782/TCP	Hueco de proceso — equipo comprometido
Proceso	wscript.exe ejecutando .js desde %TEMP%	Dropper JS activo
Archivo	%TEMP%\pid_NNNN_seq_1.js.pe	Artefacto temporal de extracción PE (forense)
Archivo	%TEMP%\~WRFZY7YYkkd.tmp	VBScript ofuscado
Archivo	%TEMP%\hlog.txt	Log interno del RAT (keylogger buffer)
Archivo	%TEMP%\hvnc_debug.log	Log de sesión HVNC
Archivo	%APPDATA%\...\Startup\NetGuard*.vbs	Persistencia Startup folder
Registro	HKCU\Software\Microsoft\Windows\CurrentVersion\Run	Persistencia Run key
Proceso hijo	WerFault.exe (hijo de RegSvcs.exe)	Artefacto post-hollowing — crash handler

6 Reglas de detección

6.1 YARA — Implante Overlord RAT

```
rule COLCERT_OverlordRAT_Implant {
  meta:
    description = "Overlord RAT / BlindEagle — implante etapa8"
    reference = "COLCERT-INC-2026-0922"
    date = "2026-09-23"
    tlp = "AMBER"
  strings:
    $beacon = "xGKGL-4447DF1C" ascii wide
    $c2domain = "envio14-9.duckdns.org" ascii wide nocase
    $hvnc = "HVNCInjection" ascii wide nocase
    $rat1 = "Overlord" ascii wide nocase
    $port = { B8 B2 12 00 00 } // push 4782 (0x12B2)
  condition:
    uint16(0) == 0x5A4D and filesize < 3MB and
    ($beacon or ($c2domain and $hvnc) or ($rat1 and $port))
}
```

6.2 YARA – Dropeer JavaScript

```
rule COLCERT_BlindEagle_JSDropper {
  meta:
    description = "Dropper JS etapa3 — ofuscacion XOR 0xF4 + AES-256-ECB"
    reference = "COLCERT-INC-2026-0922"
  strings:
    $obf1 = "JSObfuscationFunction" ascii nocase
    $obf2 = { F4 F4 F4 F4 F4 F4 F4 F4 } // patron XOR 0xF4
    $wsh = "WScript.Shell" ascii wide nocase
    $tmp = "pid_" ascii
    $pe = ".js.pe" ascii
  condition:
    filesize < 50KB and $wsh and (($obf1 and $obf2) or ($tmp and $pe))
}
```

6.3 Sigma – Process hollowing en RegSvcs-exe

title: BlindEagle Process Hollowing via RegSvcs.exe
 id: a4f2b3c1-9e8d-4f2a-b7e6-1c3d5e7a9b2f
 status: experimental
 description: Detects Overlord RAT injection into RegSvcs.exe
 references:
 - COLCERT-INC-2026-0922
 logsource:
 category: process_creation
 product: windows
 detection:
 selection:
 Image|endswith: '\RegSvcs.exe'
 ParentImage|endswith:
 - '\wscript.exe'
 - '\powershell.exe'
 - '\cmd.exe'
 network_conn:
 Image|endswith: '\RegSvcs.exe'
 DestinationPort: 4782
 condition: selection or network_conn
 falsepositives:
 - Aplicaciones .NET legítimas que invoquen RegSvcs.exe con red (muy raro)
 level: high
 tags:
 - attack.defense_evasion
 - attack.t1055.012

6.4 KQL - Microsoft Defender XDR / Sentinel

```
// Overlord RAT C2 beacon o proceso hollowing
let c2_ips = dynamic(["199.16.199.2","181.235.13.254"]);
let c2_domains = dynamic(["envio14-9.duckdns.org","qsar20.duckdns.org"]);
```

```
DeviceNetworkEvents
| where RemotePort == 4782
   or RemoteIP in (c2_ips)
   or RemoteUrl has_any (c2_domains)
| project Timestamp, DeviceName, InitiatingProcessFileName,
   RemoteIP, RemotePort, RemoteUrl
```

```
union
DeviceProcessEvents
| where FileName =~ "RegSvcs.exe"
   and InitiatingProcessFileName in~ ("wscript.exe","powershell.exe")
| project Timestamp, DeviceName, FileName,
   InitiatingProcessFileName, ProcessCommandLine
```

```
union
DeviceFileEvents
| where FolderPath contains @"\Windows\Temp\"
   and FileName matches regex @"pid_\d+_seq_\d+\.js\.pe"
| project Timestamp, DeviceName, FileName, FolderPath, InitiatingProcessFileName
```

6.5 Firma de red (Suricata / Snort)

```
/alert tcp $HOME_NET any -> $EXTERNAL_NET 4782 (
  msg:"COLCERT BlindEagle Overlord RAT C2 beacon";
  content:"xGKGL-4447DF1C"; offset:0; depth:16;
  threshold:type limit, track by_src, count 1, seconds 60;
  sid:9260922001; rev:1;
  metadata:affected_product Windows, attack_target Client_Endpoint,
    tlp AMBER, reference COLCERT-INC-2026-0922;)
```

```
alert dns $HOME_NET any -> any any (
  msg:"COLCERT BlindEagle C2 domain lookup";
  dns.query; content:"duckdns.org"; endswith;
  pcre:"/^(envio14-9|qsar20)\.duckdns\.org$/i";
  sid:9260922002; rev:1;)
```

7. Recomendaciones

7.1 Acciones inmediatas – 72 horas



- ☐ Bloquear en firewall perimetral y proxy web los dominios, IPs y puertos listados en numeral 5, Indicadores de compromiso (IOC), **Prioridad máxima:** 199.16.199.2 y envio14-9.duckdns.org en puerto 4782/TCP.
- ☐ Desplegar firmas YARA de numeral 6.1–6.2 en herramienta de análisis de archivos (EDR, sandbox, correo gateway). Prioridad: escanear adjuntos .xhtml, .html, .svg en correos entrantes.
- ☐ Activar regla Sigma de numeral 6.3 en SIEM para detección de process hollowing en RegSvcs.exe.
- ☐ Desplegar query KQL de numeral 6.4 como hunting activo y como regla de alerta en Sentinel / Defender XDR.
- ☐ **Alertar a todos los usuarios sobre el señuelo:** NO ejecutar archivos .xhtml / .html / .svg que lleguen adjuntos, aunque provengan de cuentas conocidas .gov.co.
- ☐ Verificar en EDR/antivirus si existe tráfico al puerto 4782/TCP o conexiones a duckdns.org en los últimos 30 días.

7.2 Investigación de compromiso (si hay sospecha)

- ☐ Buscar proceso RegSvcs.exe con conexiones externas activas o procesos hijo no esperados.
- ☐ Revisar HKCU\Software\Microsoft\Windows\CurrentVersion\Run en equipos del parque — entradas con nombre que contenga "NetGuard" son indicador fuerte.
- ☐ Buscar archivos *.vbs en %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\ no reconocidos.
- ☐ Buscar archivos pid_*_seq_*.js.pe en C:\Windows\Temp\ — su presencia confirma ejecución del dropper.
- ☐ Si se confirma compromiso: aislar el equipo de red ANTES de cualquier otro paso, preservar imagen RAM y disco, y reportar a ColCERT con la evidencia.
- ☐ Revisar logs de proxy/firewall para conexiones históricas a 181.235.13.254 y 199.16.199.2 — puede haber equipos comprometidos sin síntomas visibles (HVNC es completamente silencioso).

7.3 Corto plazo – 15 días

- ☐ Implementar bloqueo de extensiones .xhtml, .svg y .html en gateway de correo. Cuarentena automática de adjuntos con esas extensiones.
- ☐ Habilitar PowerShell Constrained Language Mode (CLM) y Script Block Logging para detectar ejecuciones codificadas (-EncodedCommand).
- ☐ Revisar configuración DMARC/DKIM/SPF en todos los dominios .gov.co propios. Las cuentas comprometidas aprovechan la ausencia de controles de autenticación de correo.
- ☐ Evaluar despliegue de solución con kernel callbacks (EDR de nivel kernel) para detectar NT direct syscalls usadas en el process hollowing.
- ☐ Cargar hashes de §5.2 en MISIP, OpenCTI o plataforma CTI propia para correlación con otros incidentes nacionales.

8. Ampliación Técnica

El informe técnico completo de este caso —con la cadena forense detallada, análisis estático y dinámico de cada artefacto, reglas de detección (YARA, Sigma, KQL, Suricata) e indicadores de compromiso completos— está disponible para las entidades del sector público colombiano en la sección **Boletines e Informes Técnicos** del sitio web del ColCERT: www.colcert.gov.co. Para solicitar acceso escríbanos a contacto@colcert.gov.co con el código **COLCERT-INC-2026-0922** en el asunto.

9. Reporte y contacto

Ante cualquier detección de los IoC descritos o comportamiento sospechoso compatible con esta campaña, **reporte de inmediato a ColCERT:**

Correo institucional: contacto@colcert.gov.co

Incluya en el reporte: nombre de la entidad, descripción del incidente, capturas de pantalla, logs de red/endpoint relevantes y artefactos (comprimidos con contraseña). **Indique "INC-2026-0922" en el asunto para correlación inmediata.**



La información contenida en este documento, bajo clasificación TLP:CLEAR - -Pública puede ser utilizada y compartida libremente con fines informativos, técnicos y de prevención, siempre que se cite como fuente al **Equipo de Respuesta a Emergencias Cibernéticas de Colombia (ColCERT)**.
Uso permitido con atribución. © ColCERT, 2026.

