Autodiscovering the Great Leak

by Amit Serper (https://www.guardicore.com/author/amit-serper/)

Executive Summary

- Autodiscover, a protocol used by Microsoft Exchange for automatic configuration of clients such as Microsoft Outlook, has a design flaw that causes the protocol to "leak" web requests to Autodiscover domains outside of the user's domain but in the same TLD (i.e. Autodiscover.com).
- Guardicore Labs acquired multiple Autodiscover domains with a TLD suffix and set them up to reach a web server that we control. Soon thereafter, we detected a massive leak of Windows domain credentials that reached our server.
- Between April 16th, 2021 to August 25th, 2021 we have captured:
  - **372,072 Windows domain credentials** in total.
  - **96,671 UNIQUE** credentials that leaked from various applications such as Microsoft Outlook, mobile email clients and other applications interfacing with Microsoft's Exchange server.
- This is a severe security issue, since if an attacker can control such domains or has the ability to "sniff" traffic in the same network, they can capture domain credentials in plain text (HTTP basic authentication) that are being transferred

over the wire. Moreover, if the attacker has DNS-poisoning capabilities on a large scale (such as a nation-state attacker), they could systematically syphon out leaky passwords through a large-scale DNS poisoning campaign based on these Autodiscover TLDs.
- Additionally, we have developed an attack – "The ol' switcheroo" – which downgrades a client's authentication scheme from a secure one (OAuth, NTLM) to HTTP Basic Authentication where credentials are sent in clear text.

Introduction

As a part of the ongoing security research efforts by the Guardicore Labs team, we have discovered an interesting case of credential leak affecting a large number of people and organizations worldwide.

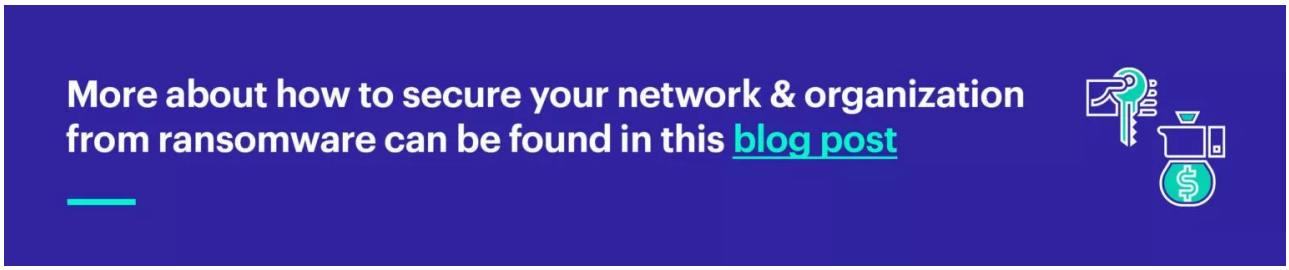
The credentials that are being leaked are valid Windows domain credentials used to authenticate to Microsoft Exchange servers. The source of the leaks is comprised of two issues:

1. The design of Microsoft's **Autodiscover** protocol (and the "back-off" algorithm, specifically).
2. Poor implementation of this protocol in some applications.

As mentioned, Microsoft's Autodiscover protocol (https://docs.microsoft.com/en-us/exchange/client-developer/exchange-web-services/autodiscover-for-exchange) was meant to ease the configuration of Exchange clients such as Microsoft Outlook. The protocol's goal is to make an end-user be able to completely configure their Outlook client solely by providing their username and password and leave the rest of the configuration to Microsoft Exchange's Autodiscover protocol. It is important to understand that since Microsoft Exchange is part of the "Microsoft domain suite" of solutions, the credentials that are necessary to login to one's Exchange-based inbox are in most cases their domain credentials. The implications of a domain credential leak in such scale are massive, and can put organizations in peril. Especially in today's ransomware-attacks ravaged-world – the easiest way for an attacker to gain entry into an organization is to use legitimate and valid credentials.

In 2017, researchers from Shape Security published a paper (https://www.blackhat.com/docs/asia-17/materials/asia-17-Nesterov-All-Your-Emails-Belong-To-Us-Exploiting-Vulnerable-Email-Clients-Via-Domain-Name-Collision-wp.pdf) about how Autodiscover implementations on email clients on mobile phones (such as Samsung's mail client on Android and Apple Mail on iOS) can cause such leaks (CVE-2016-9940, CVE-2017-2414). The vulnerabilities disclosed by Shape Security were patched, yet, here we are in 2021 with a significantly larger threat landscape, dealing with the exact same problem only with more third-party applications outside of email clients. These applications are exposing their users to the same risks. We have initiated responsible disclosure processes with some of the

the same risks. We have initiated responsible disclosure processes with some of the vendors affected. More details on that aspect will be released as a second part to this paper.



(https://www.guardicore.com/blog/stopping-ransomware-with-segmentation/)

This document will detail how the aforementioned protocol's design issues cause severe credential leak that gives us the ability to receive **tens-of-thousands of valid Windows domain credentials without sending a single packet.**

What is Autodiscover?

Exchange's Autodiscover protocol was made to provide a way for clients to easily configure their Exchange client applications. Usually, in order to configure a mail client, the user has to configure multiple settings:

- Username and password.
- The hostnames/IP addresses of the mail/Exchange servers.
- In some cases, additional settings are required (Miscellaneous LDAP settings, WebDAV calendars, etc.).

The protocol has several iterations, versions and modes – their full documentation can be found on Microsoft's website (https://docs.microsoft.com/en-us/exchange/client-developer/exchange-web-services/autodiscover-for-exchange), however, in this article, we will discuss a specific implementation of Autodiscover based on the POX XML protocol. Once the user adds a new Microsoft Exchange account to Outlook, the user will receive a prompt that asks for their username and password:

Microsoft Outlook auto account setup

Once the user fills in all of the details, Outlook will then try to use Autodiscover in order to configure the client. This stage in the process looks like this:

Microsoft Outlook auto account setup process

However, in order to truly understand how Autodiscover works, we need to know what happens “behind the scenes”:

1. The client parses the email address supplied by the user – amit@example.com.
2. The client then tries to build an Autodiscover URL based on the email address with the following format:

- https://Autodiscover.example.com/Autodiscover/Autodiscover.xml
- http://Autodiscover.example.com/Autodiscover/Autodiscover.xml
- https://example.com/Autodiscover/Autodiscover.xml
- http://example.com/Autodiscover/Autodiscover.xml

In the case that none of these URLs are responding, Autodiscover will start its **"back-off"** procedure. This "back-off" mechanism is the culprit of this leak because it is always trying to resolve the Autodiscover portion of the domain and it will always try to "fail up," so to speak. Meaning, the result of the next attempt to build an Autodiscover URL would be: **http://Autodiscover.com/Autodiscover/Autodiscover.xml.** This means that whoever owns **Autodiscover.com** will receive all of the requests that cannot reach the original domain. For more information about how Autodiscover works, please check out Microsoft's documentation (https://docs.microsoft.com/en-us/exchange/client-developer/web-service-reference/pox-autodiscover-request-for-exchange?redirectedfrom=MSDN).

Autodiscover "back-off" process

Abusing the Leak

In order to see if the Autodiscover leak scenario is even a viable one, we have purchased the following domains:

- Autodiscover.com.br – Brazil
- Autodiscover.com.cn – China
- Autodiscover.com.co – Columbia
- Autodiscover.es – Spain
- Autodiscover.fr – France
- Autodiscover.in – India
- Autodiscover.it – Italy

- Autodiscover.it – Italy
- Autodiscover.sg – Singapore
- Autodiscover.uk – United Kingdom
- Autodiscover.xyz
- Autodiscover.online
- Autodiscover.cc
- Autodiscover.studio
- autodiscover.capital
- autodiscover.club
- autodiscover.company
- autodiscover.jp
- autodiscover.me
- autodiscover.mx
- autodiscover.ventures

Later, these domains were assigned to a webserver in our control and we were simply waiting for web requests for various Autodiscover endpoints to arrive. To our surprise, we started seeing significant amounts of requests to Autodiscover endpoints from various domains, IP addresses and clients. The most notable thing about these requests was that they requested the relative path of */Autodiscover/Autodiscover.xml* with the Authorization header already populated with credentials in HTTP basic authentication.

example of a simple HTTP GET request with the Authorization header already populated with credentials

Generally, web requests should not be sent blindly pre-authenticated, but rather following the HTTP authentication process:

1. A client requests access to a protected resource.
2. The web server returns a dialog box that requests the username and password (in accordance with the supported authentication methods, in our case, basic authentication).
3. The client submits the username and password to the server.

4. The server authenticates the user and returns the requested resource.

HTTP basic authentication process illustrated

As can be seen in the following excerpt, the hostnames appearing in the log (scrubbed for privacy reasons) are the domains from which the Autodiscover clients were trying to authenticate to, along with their respected username and passwords:

```
2021–05–18 03:30:45 W3SVC1 instance-2 10.142.0.4 GET /Autodiscover/Autodiscover.xml – 80 – <IP address scrubbed> HTTP/1.1 Microsoft+Office/16.0+ (Windows+NT+10.0;+Microsoft+Outlook+16.0.13901;+Pro) – -<Victim domain scrubbed> 404 0 2 1383 301 265 <Victim domain scrubbed> Basic+<base64 encoded credentials scrubbed>= – –
2021–05–18 03:30:52 W3SVC1 instance-2 10.142.0.4 GET /Autodiscover/Autodiscover.xml – 80 – <IP address scrubbed> HTTP/1.1 Microsoft+Office/16.0+ (Windows+NT+10.0;+Microsoft+Outlook+16.0.13901;+Pro) – – <Victim domain scrubbed> 404 0 2 1383 301 296 <Victim domain scrubbed> Basic+<base64 encoded credentials scrubbed> – –
2021–05–18 03:30:55 W3SVC1 instance-2 10.142.0.4 GET /Autodiscover/Autodiscover.xml – 80 – <IP address scrubbed> HTTP/1.1 Microsoft+Office/16.0+ (Windows+NT+10.0;+Microsoft+Outlook+16.0.13901;+Pro) – – <Victim domain scrubbed> 404 0 2 1383 296 328 <Victim domain scrubbed> Basic+<base64 encoded credentials scrubbed> – –
2021–05–18 03:31:19 W3SVC1 instance-2 10.142.0.4 GET /Autodiscover/Autodiscover.xml – 80 – <IP address scrubbed> HTTP/1.1 Microsoft+Office/16.0+ (Windows+NT+10.0;+Microsoft+Outlook+16.0.13901;+Pro) – – <Victim domain scrubbed> 404 0 2 1383 306 234 <Victim domain scrubbed> Basic+<base64 encoded credentials scrubbed> – –
```

The interesting issue with a large amount of the requests that we received was that there was no attempt on the client's side to check if the resource is available, or even exists on the server, before sending an authenticated request. Usually, the way to implement such a scenario would be to first check if the resource that the client is requesting is valid, since it could be non existent (which will trigger an HTTP 404

error (https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404)) or it may be password protected (which will trigger an HTTP 401 error code (https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/401)) as seen in the above diagram.

Between Apr 16, 2021 to Aug 25, 2021 we have captured a large number of credentials this way, needless to say, without sending a single packet other than what's required to establish an HTTP/HTTPS session between our server and the miscellaneous clients.

The following data was collected between April 20th 2021 to August 25th 2021:

While examining the domains from these leaked credentials, we were able to find credentials from various companies across multiple verticals:

The ol' switcheroo

Illustation of the ol' switcheroo attack

When observing the logs of the HTTP server, we can clearly see that the client is successfully downgraded after receiving the HTTP 401 response from the server, telling it to use HTTP Basic Authentication:

(https://www.guardicore.com/wp-content/uploads/switcheroo.jpg)
Successful ol' switcheroo attack (Click image to enlarge)

Note: The empty bearer token is sent as a part of the realm Autodiscovery process which is discussed further in Microsoft's documentation (https://docs.microsoft.com/en-us/openspecs/exchange_server_protocols/ms-xoauth/f622dee1-a9a7-436e-b44e-6ecf8f536f77). On the victim's side ,however, it is difficult to even realize that the user is experiencing any sort of attack. When the victim is being redirected to our Autodiscover server due to the leak, a security alert pops up:

Security alert prompted by Microsoft Outlook

This warning indicates that while the certificate is valid, it is self-signed and should not be trusted. However, this could be easily avoided by deploying an actual SSL certificate. In our case, we deployed a LetsEncrypt (https://letsencrypt.org/) certificate within seconds, which remediated the issue of the SSL warning being displayed.

Installing LetsEncrypt SSL certificates with WinAcme

autodiscover.sg is now secured with a valid certificate

Once a secure session has been established, the victim will now see this legitimate authentication prompt displayed by Microsoft Outlook:

Basic authentication dialog displayed in Outlook as a result of a successful ol' switcheroo attack

This is the last stage of this attack – the victim will input their credentials into this dialog box, which in turn, will send the credentials to our web server. The credentials now appear in our logs.

Mitigation

Mitigating the issue of Autodiscover leaks is important as we have previously demonstrated. In order to mitigate this issue, two separate approaches are required:

1. One approach needs to be implemented by the general public who use Exchange-based technologies such as Outlook or ActiveSync (https://en.wikipedia.org/wiki/ActiveSync) (Microsoft's mobile Exchange synchronization protocol) and the other approach should be implemented by software developers/vendors who are implementing the Autodiscover protocol in their products:
2. For the general public:Make sure that you are actively blocking Autodiscover. domains (such as Autodiscover.com/Autodiscover.com.cn, etc) in your firewall.

Guardicore Centra's DNS Security allows creating block rules for Autodiscover domain names

- When deploying/configuring Exchange setups, make sure that support for basic authentication is disabled – using HTTP basic authentication is the same as sending a password in clear text over the wire.
- A comprehensive textual list of all top level domains can be found in the following url: https://data.iana.org/TLD/tlds-alpha-by-domain.txt (https://data.iana.org/TLD/tlds-alpha-by-domain.txt)
    - We have prepared a txt file with all possible Autodiscover.TLD domains which can be added to your local hosts file or firewall configuration in order to mitigate the risk of having such Autodiscover domains resolve. Please check our github repository for more information: https://github.com/guardicore/labs_campaigns/tree/master/Autodiscover (https://github.com/guardicore/labs_campaigns/tree/master/Autodiscover)

3. For software vendors and developers:

- Make sure that when you are implementing the Autodiscover protocol in your product you are not letting it "fail upwards", meaning that domains such as "Autodiscover." should never be constructed by the "back-off" algorithm.

Conclusion

In this document, we discussed the implications of the basic design flaw within the Autodiscover protocol (the "back-off" algorithm) and demonstrated that if an attacker controls top-level Autodiscover domains (or if the attacker has the ability to conduct a DNS-poisoning attack using these domains), they can easily consume valid domain credentials from these leaky Autodiscover requests.

Oftentimes, attackers will try to cause users to send them their credentials by applying various techniques, whether technical or through social engineering. However, this incident shows us that passwords can be leaked outside of the organization's perimeter by a protocol that was meant to streamline the IT department's operations with regards to email client configuration without anyone from the IT or security department even being aware of it, which emphasises the importance of proper segmentation and Zero trust.

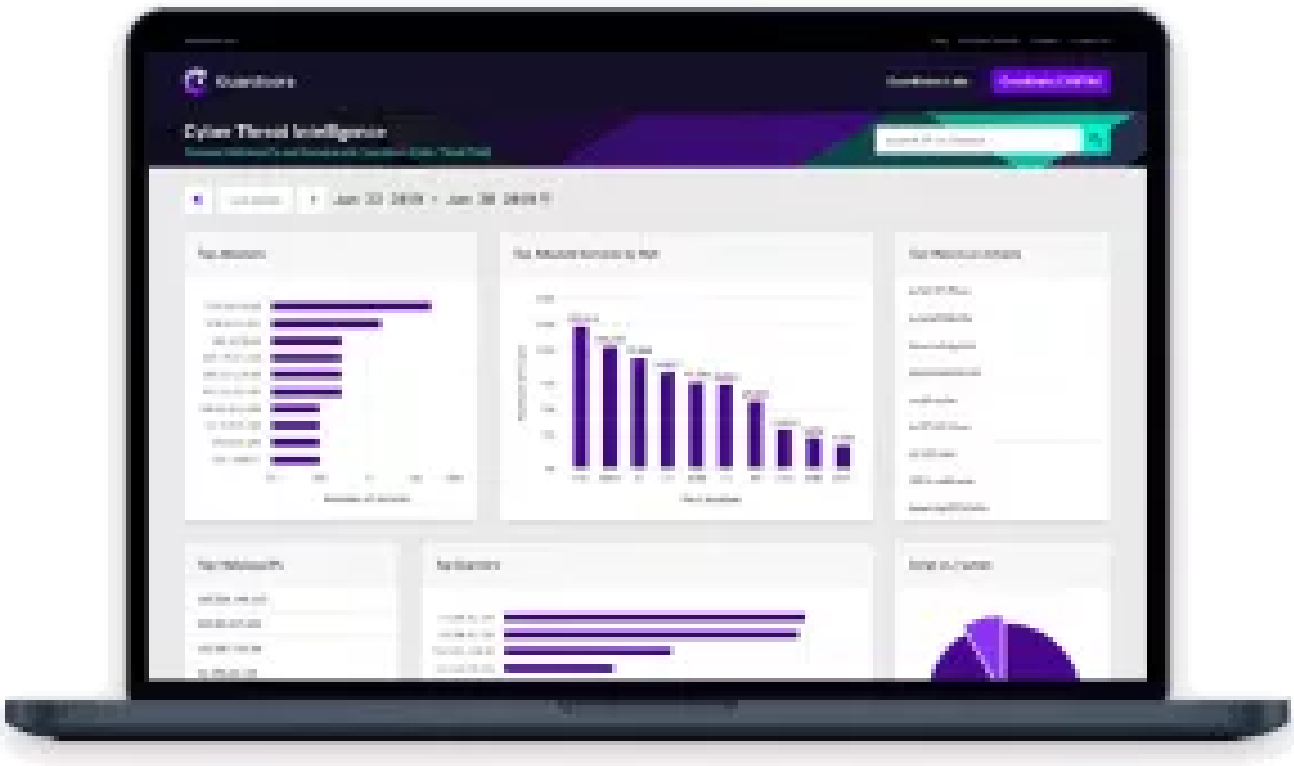
We, at Guardicore Labs, are continuing our ongoing efforts to secure networks, applications, and protocols alike by finding, alerting and disclosing such issues.

SUBSCRIBE

FOLLOW US ON

(https://www.linkedin.com/company/guardicore)
(https://twitter.com/guardicore) (https://www.facebook.com/guardicore/)

(https://www.guardicore.com/infectionmonkey/breach-and-attach-simulation/)



Cyber Threat Intelligence

Get unique information on malicious Internet assets – IP addresses and domain – detected by Guardicore.

View Dashboard (https://threatintelligence.guardicore.com/)

SHARE THIS ARTICLE:

(https://www.guardicore.com/labs/validate-

(https://www.linkedin.com/company/guardicore/) (https://twitter.com/guardicore) (https://www.facebook.com/guardicore) (https://www.youtube.com/channel/UCxu1wG7lO4t