SUCCESSFUL (AND EASY) ATTACK VECTORS 2020

Prepared for : eForensics Magazine

Prepared by: LIFARS LLC, Offensive Security Services Department

Date: February/2021

Successful (And Easy) Attack Vectors 2020

How can attackers access your MFA-protected Company mailboxes? How can they move from one machine to another in your infrastructure? Which attack vectors were feasible in 2020? This article is written with the optics of Red Team. It summarizes the most successful attack vectors that repeatedly worked for us in 2020 Red Teaming engagements and penetration tests. The goal of this article is to summarize some of the weaknesses that are still being overlooked or insufficiently mitigated, and exploitation of which might have serious consequences.

Introduction

This article is written from the perspective of white hat hackers. Its goal is to point out what can still be easily exploitable in 2021, based on our experiences from last year. It describes a different set of attack vectors than what our colleagues from LIFARS DFIR department have commonly seen being misused during their recent engagements. And that is understandable, as security posture of companies that are willing to engage penetration testers or red teamers is probably somewhat better than security posture of many companies seeking reactive DFIR services.

In this article, we describe the most successful attack vectors that repeatedly worked for us (LIFARS Offensive Security Department) in 2020 infrastructure penetration tests and red team engagements. Spear phishing attacks are deliberately omitted from this summary. So, what was it?

- Weak passwords and password sprayable authentication endpoints.
- Mailboxes accessible without MFA through Exchange ActiveSync and EWS.
- Same local administrator credentials reused across multiple devices.
- Overprivileged accounts.
- Sensitive information stored on globally accessible network shares.
- Cached domain credentials and readable LSASS memory (without protections like Credential Guard and RunAsPPL).
- L2 attacks (ARP poisoning, LLMNR/NBNS poisoning).

Clients we assessed in 2020 were mostly large companies with 10.000+ employees, datacenters and branch offices across multiple continents, working in multiple time zones; or smaller firms with relatively large IT infrastructures.

Accessing Company systems from Internet (exploiting authentication weaknesses)

Having been forced to move to work-from-home environments during Spring 2020, companies made it possible for their employees to access virtually all internal systems remotely. Employee mailboxes, document storage, desktops, even administration interfaces are accessible after authentication either directly from the Internet or through a VPN tunnel. Business requires ease of use. Therefore, most of company systems are accessible by an

244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

employee that successfully authenticates using a single identity[1]. The trust in authentication process is critical for balancing ease of use and security. For this reason, companies deploy and require MFA (Multi-Factor Authentication) to access most of externally facing systems. Passwords may be stolen or guessed, whereas spoofing "something I have" (e.g. authentication app in Mobile phone) is generally significantly harder.

So, how do we gain our initial foothold, how do we "hack" into a mailbox of a Company employee? We need to know / have three things: username + valid password + bypass or respond to any second factor authentication request.

Based on our experience, usernames can often be enumerated (e.g. ADFS time-based enumeration[2]) or guessed with a high degree of confidence. Prerequisites are:

- performing OSINT (Open-Source Intelligence gathering)[3] and reconnaissance;
- generating a list of potentially valid usernames, a step that is easier the larger the Company is;
- identifying authentication targets (might be Office365 authentication or ADFS service or other).

The next step is to pair a username with a valid password.

Weak passwords, password spraying

TL;DR: Weak passwords are still a thing and stealthy password spraying[4] works nicely.

The larger the Company is, the higher the probability of finding passwords like "`February2021!`". Note that this password is 13 characters long and it includes all four-character sets. Is it strong enough? No. How does the user (and the attacker) adapt to 14-character requirement? "`February2021!!`".

Some of your users are probably using the following types of passwords. The attackers know it.

```
December2020
February2021!
Summer2021
Welcome2021
Welcome123
Password2021
{Company}123
{Company}123!
```

What did we target the most? From outside it was ADFS (Active Directory Federation Services) authentication portals and Lync servers (Skype for Business). Some of our clients did have the detection capabilities (e.g., Advanced Threat Analytics alerts), but in most cases they lacked automatic response that would block our attempts.

---

[1] or multiple identities in case of administrators – unprivileged and specialized administration account(s).
[2] https://cxsecurity.com/issue/WLB-2018100214
[3] https://en.wikipedia.org/wiki/Open-source_intelligence
[4] a type of brute force attack where the attacker tries to gain access to an organization's systems by testing out a small number of commonly used passwords on a large number of accounts

244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

What can you do?
@weak passwords: In addition to standard approaches, such as user training and password policy, *restrict use of common passwords and dictionary words*. To achieve this goal, you may give a try to Azure AD Password Protection[5] and create a Custom banned password list.

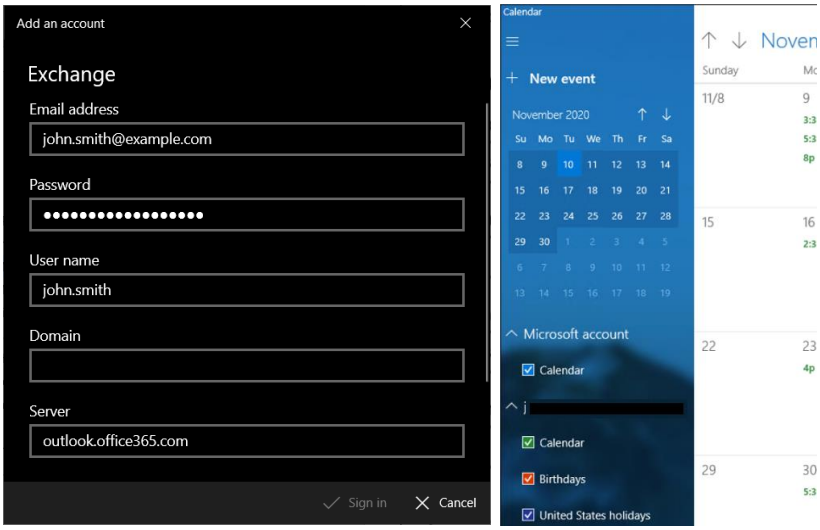
@password spraying: Monitoring, alerting and responding to those alerts. Lots of unsuccessful authentication attempts from a single source IP or weird behavior. Make sure your SOC always *detects and reacts to unanswered second factor authentication requests*.

Forgotten MFA – ActiveSync and Exchange Web Services

TL;DR: Employee mailboxes are assets protected by MFA. Or so the companies think. In reality Office365 mailboxes tend to be unintentionally accessible without MFA using specific protocols – ActiveSync and EWS.

Exchange ActiveSync is an Exchange synchronization protocol that allows clients to synchronize their mobile device with Exchange mailbox to access their email, calendar, contacts and tasks. By default, Exchange ActiveSync is enabled and it doesn't require MFA. Exchange Web Services (EWS) is a web API for Exchange server which also lets clients access mailboxes, calendars and contacts. It is also enabled by default.

Now, why is it a problem? Utilizing ActiveSync with no MFA, we can use Windows Mail App to conveniently access a user's mailbox and calendar just by authenticating with username and password. We even get to use official GUI application to interact with the mailbox.



Caption: Convenient access to mailbox using Windows Mail App and Exchange ActiveSync

Utilizing EWS, we can use attack tools (e.g. MailSniper[6]) to access and search mailboxes and dump Global address list (GAL[7]).

[5] https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-password-ban-bad
[6] https://github.com/dafthack/MailSniper
[7] https://docs.microsoft.com/en-us/exchange/email-addresses-and-address-books/address-lists/address-lists?view=exchserver-2019

```
PS C:\Users\IEUser\Desktop> Invoke-SelfSearch -Mailbox [redacted].com -ExchHostname outlook.office365.com -Remote -Output
Csv mailsearch1.csv -CheckAttachments -DownloadDir mailsearch1-down
[*] Trying Exchange version Exchange2010

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
[*] Using EWS URL https://outlook.office365.com/EWS/Exchange.asmx
[***] Found folder: Inbox
[*] Now searching mailbox: [redacted].com for the terms *password* *creds* *credentials*.
```

Caption: Using MailSniper to search the target Mailbox for emails containing words "password", "creds" and "credentials"

```
[*] Now cleaning up the list...
A total of 5560 email addresses were retrieved
```

Caption: Retrieving GAL using MailSniper

At this point we have a list of valid targets (identities), some information about internally used systems based on reviewed emails and we have access to mailboxes, from which we can perform spear phishing attempts or try to abuse the client-side Outlook features and gain a shell remotely[8].

What can you do?
Disable services that you do not use. Require MFA when you do use them. Tighten access rules specifying which devices can access Exchange using ActiveSync.
You can check your Company's exposure using tools like MFASweep[9].

We are inside the network, what now?

Objective: get access to other systems, find data, then do my thing that I came to do (this differs per threat actors).

Having obtained access to the network – to a Windows client system or to a Linux server in DMZ, or physically connecting a device into internal network – what next steps are most likely to succeed?

Same local administrator credentials across multiple devices

TL;DR: Local administrator credentials are still being reused across multiple devices. This allows us to move from host to host in pursuit of elevating privileges and accessing other systems.

Once we get local administrator access to a Windows system, we usually extract the local administrator's password hash from HKLM\SAM Windows Registry and try to use it to authenticate (pass-the-hash/overpass-the-hash) and access other reachable Windows machines to see whether the same credentials are reused there. It still works in majority of engagements. Fortunately, it is not an issue that affects an infrastructure globally anymore. However, in 2020 we still commonly found pools of machines using the same password. Local administrator credential reuse increases our chances of finding other credentials that are going to work for other systems.

Among notable countermeasures that we encountered during Red Team engagements was Microsoft ATA[10]. It didn't stop us but forced us to be more cautious and made our next steps more difficult.

[8] https://github.com/sensepost/ruler
[9] https://github.com/dafthack/MFASweep
[10] https://docs.microsoft.com/en-us/advanced-threat-analytics/what-is-ata

Note from our DFIR guys: We see the same thing during our investigations - LAPS is still NOT a thing, in majority of cases, we haven't seen it implemented.

What can you do?
The objective is using unique accounts for every system. Use Microsoft LAPS (*Local Administrator Password Solution*) to manage local admin account passwords.

Limit lateral movement possibilities to a minimum by restricting network access to other computers. *Isolate hosts on local network* using Private VLANs. Make sure that ports like TCP/445, TCP/5985-5986 (PowerShell Remoting) are not whitelisted across segments when not required.

Overprivileged Accounts

TL;DR: We still encounter standard domain accounts belonging to local administrators group on some endpoints. We also still sometimes see a service account with Domain Admin privileges.

By Overprivileged Accounts we mean several issues.
1) In 2020 we have not seen employees commonly being granted local administrators privileges on their computers. However, we still occasionally encountered users or the whole groups being assigned to local administrator group on specific servers or workstations.

How does this help us (the attackers)? Having already obtained a valid set of Domain credentials, we query the Domain Controller for details about AD (Active Directory) users, groups and computers. If the monitoring doesn't seem to be an issue, we proceed by enumerating local administrator members across reachable computers. The hunt for escalating privileges - possibly even to Domain admin level - becomes a matter of orientating in the collected data and jumping from machine to machine, dumping credentials and using them to access other machines.

2) Service accounts are needlessly overprivileged. Last year, we have seen several instances of service (i.e. not employee-bound) accounts belonging to Domain Admins group. These tend to be used across multiple devices by a specific software, and overlooked by monitoring.

3) We haven't seen the practice of routinely granting local administrator privileges to normal employee accounts for a few years already. This makes sense: A company willing to perform a pentest is probably going to have the security basics in place. On the other hand, our DFIR department sees a different story.

Note from our DFIR guys: We still encounter customers, especially smaller companies, whose employees accounts have local administrator privileges on their workstations.

What can you do?
Invest your time into auditing existing privileges and restricting them. There is no shortcut. Go over all accounts step by step, assess what privileges they have and whether there still is

244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

a business need for them. You can use tools like BloodHound[11] to help you during auditing. Prevent making policy exceptions and granting unnecessary privileges.

Sensitive information on accessible network shares

TL;DR: Disregarding business related documents, we have seen recent server VM backups and images and even the whole system drive C$ being shared and accessible to a standard domain user account.

It always surprises us what kind of data we can find on accessible network shares (SMB and NFS). We have seen VM backups and full system images and successfully used them to extract user account information. We even hit the jackpot once when we found backups of Domain Controller VMs.

Looking back on our last year's engagements, it turns out it is still not uncommon to find SMB or NFS shares with various backups (databases, Jenkins, etc.). Sometimes we even find the whole system drive being shared (either directly or as a result of Domain Users group belonging to the local administrators group on the target machine).

Note that during some engagements we encountered NFS shares which were configured to authenticate the user solely based on user supplied UID (user identifier). This allowed us to identify ourselves with spoofed UIDs, thus accessing shared files belonging to other users.

Note from our DFIR guys: Overexposed shares are an issue in our engagements too, however for a different reason – ransomware. We would like to see more infrastructures where workstations are isolated from each other.

What can you do?
Audit what shares really are in your infrastructure. SMB shares as well as NFS shares. What can a standard user account access? Tighten access privileges to a minimum level required for business purposes.

Isolate hosts on local network using Private VLANs. Make sure that related ports TCP/445, UDP/2049 are not whitelisted across segments when not required.

Cached domain credentials and readable LSASS

TL;DR: Dumping credentials became more difficult by adopting Windows 10. However, we still see infrastructures that do not have advanced memory protections enabled on Windows machines. Furthermore, some Windows 7 and Windows 2008 and 2012 can still be found in most infrastructures.

Dumping LSASS (Local Security Authority Subsystem Service) process memory turned out to be slightly more difficult during last year's engagements than before. Curiously, what most often prevented us from dumping and analyzing lsass.exe memory was EDR (Endpoint Detection and Response) agents installed on endpoints. Several of our clients also had LSA Protection turned on. On the other hand, our DFIR colleagues' experience suggests that Mimikatz can still run wild in some organizations.

[11] https://bloodhound.readthedocs.io/en/latest/

In cases when dumping LSASS memory is not feasible, extracting and analyzing SAM, SECURITY and SYSTEM registry hives from HKLM is still a fruitful approach. In addition to local users' hashes, it may - and sometimes did - contain cached credentials.

Note from our DFIR guys: Protected Accounts are still NOT a thing. We commonly see domain admin accounts being used to manage client machines, thus leaving at least a password hash in the memory. We also still sometimes encounter Mimikatz and alikes being successfully used directly on endpoints.

What can you do?
Enable LSA Protection[12] and Credential Guard on all Windows machines. Make sure that caching credentials in memory (WDigest) is turned off[13].

L2 attacks (ARP poisoning, LLMNR/NBNS poisoning)

TL;DR: Parts of customers' infrastructures still tend to be susceptible to Network Layer 2 attacks like ARP poisoning and LLMNR/NBNS poisoning.

The game changed a bit with regards to L2 attacks last year. It changed in two ways, mainly due to VDI (Virtual Desktop Infrastructure) being deployed in the cloud where the virtual desktops tend to be isolated from other machines on L2. The second thing that happened last year is that internal testing switched from on premise testing to remote testing. This has implications specifically regarding ARP spoofing attacks – we, the good guys attacking and assessing the infrastructure, are more way more cautious while performing ARP spoofing attacks using our remote access. Moreover, the attack requires elevated privileges on a computer connected to the infrastructure. However, L2 attacks are still possible in some infrastructures, mainly in on-premises networks.

ARP poisoning MitM (Man-in-the-Middle) attack[14] works by spoofing responses to ARP (Address Resolution Protocol) requests that ask to translate an IP address to a MAC address. The objective is tricking the target machine and local gateway into sending traffic destined to the other IP address to our machine. With some exceptions (like VDI in the cloud or specifically hardened segments), ARP poisoning is still a feasible option for attackers.

Some of the customers are still vulnerable to LLMNR and NBNS poisoning attacks[15], as well as WPAD (Web Proxy Auto-Discovery Protocol) MITM attacks.

How does it help us (the attackers)?
ARP spoofing allows us to sniff the traffic of other devices in local area network and capture information like interesting targets, passwords and secrets (e.g. SNMP strings, credentials submitted through HTTP, NetNTLMv2 hashes, VOIP related secrets, etc.).

Responding to LLMNR/NBNS requests and subsequent SMB/WPAD connect requests allows us to capture usernames and related NetNTLMv2 hashes. We cannot use them directly

[12] https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/configuring-additional-lsa-protection
[13] HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\WDigest -> UseLogonCredential = 0
[14] https://lifars.com/wp-content/uploads/2020/02/case-study-NAC-Bypass-and-ARP-Spoofing.pdf
[15] https://attack.mitre.org/techniques/T1557/001/

244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

(apart from relaying attacks), but we can try to crack them offline. In case of WPAD, we can conduct web proxy MITM attacks.

What can you do?
Isolate hosts on local network using Private VLANs. This is a true bang-for-your-money gem. Host isolation makes attacker's job significantly more difficult.

Endnote: What makes attackers' life more difficult?

When they have reason to fear being easily detected. The combination of low-privileged user account and EDR agent on an endpoint (employee desktop) significantly obstructs the attacker's proceeding. It gets somewhat harder to remain undetected with a quality EDR installed on endpoint and reporting back to a SOC (Security Operations Center).

Let us repeat several of the controls recommended previously. The following are inexpensive controls that are definitely going to make the attacker life considerably harder:

- Isolate hosts on LAN using Private VLANs.
- Use unique local administrator passwords - implement LAPS.
- Enable LSA Protection and Credential Guard on all Windows machines.
- Invest your time into auditing existing privileges and restricting them.

About the author

LIFARS Offensive Security Department. LIFARS is a highly technical, New York City based incident response and digital forensics firm specializing in proactive and reactive services.

244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com