COMPROMITATION OF NETWORK INFRASTRUCTURE THROUGH THE DEFAULT SETTINGS OF WINDOWS OS

Daniel Paučo
Doctoral Degree Programme (1st), FEEC BUT
E-mail: xpauco00@stud.feec.vutbr.cz

Supervised by: Lukáš Malina
E-mail: malina@feec.vutbr.cz

Abstract: The number of cyber-attacks grows every single day. In fact, in 2020, the Microsoft reported about 64% increment of the number of reported vulnerabilities in the last 5 years. In this article, we present and demonstrate some recent cyber-attacks and security threats that can compromise whole domain network in many current enterprises. These attacks use mainly default Windows OS services, settings and protocols.

1 INTRODUCTION

Computer networks are nowadays expanded almost everywhere. From households or shops to factories, there are computers that are connected to the Internet from the majority or they are at least interconnected in the Local Area Network (LAN). As these systems are connected to the internet, the risk of compromitation is increasing. If they are not connected to the internet but still communicating in a LAN, there is still risk of compromitation for example from a perspective of a dissatisfied employee who has access to the network.

The main objective of this article is to describe and analyze default settings of Windows OS protocols, analyze the attacks aimed at these protocols, and describe the countermeasures to mitigate the critical information leakage leading to system compromitation. These attacks were chosen because they are considered as "low-hanging fruits". It means, that they are easy to realize and could potentionaly bring sensitive information. The attacks performed during this research were aimed at Windows Server 2008, 2012 and 2016.

This article describes a network infrastructure based on Windows OS and its protocols. It comes to protocols that could lead to information leakage when set up improperly. These information could be used for example to authenticate to the system. Scenarios and attacks described in this article simmulate the attacker located in the internal network of the given infastructure. The main objective of the attacker is to compromise Domain Controller (DC) that runs as a control point of the whole domain. The role of the domain controller is to store authentication data for the whole domain, control the access to the domain services, etc.

2 ATTACK SCENARIO PREPARATION

To demonstrate described attacks there was designed and implemented cyber exercise with a scenario of penetration test of certain company. This exercise was implemented in virtualization platform VMWare. There is a network infrastructure consisting of Demilitarized Zone (DMZ) containing Linux servers and internal network (SRV) consisting of Windows Server 2012 and Windows Server 2016.

The Figure 1 represents the scheme of the laboratory network. The simulation of penetration tester starts out of the company network by a reconnaissance phase where the pentester explores Domain

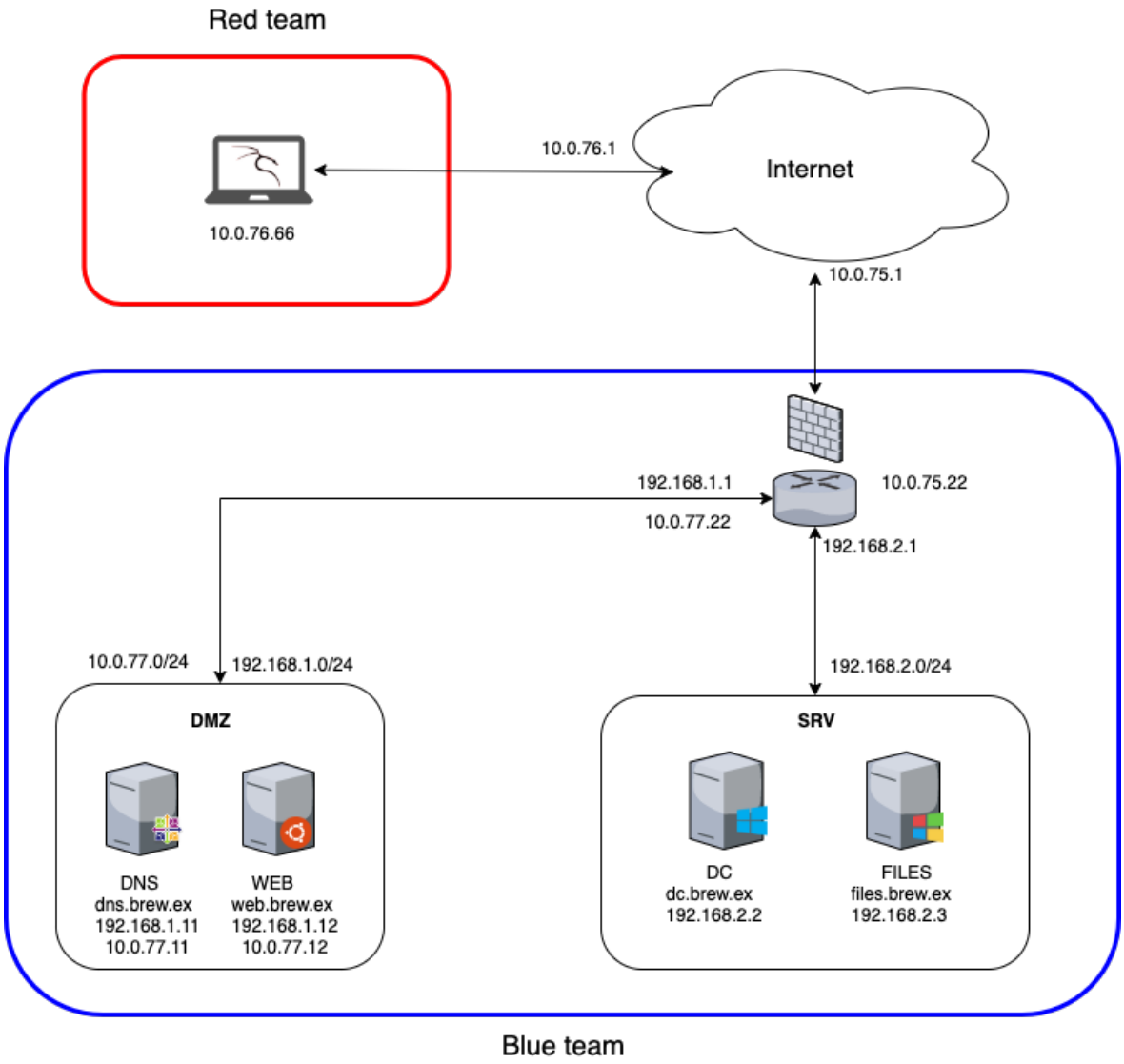

Figure 1: Scheme of the network infrastructure

Name System (DNS) server and web server situated in DMZ. This web server contains preinstalled advanced web shell that has to be found and used to get reverse shell to the pentesters machine (10.0.76.66). This compromised machine could be used as a pivot to the internal network of the company what is the next step of the exercise. This intrusion is possible due to improper system configurations and its protocols. Files server has Remote Desktop (RDP) service enabled that is not secured well and can be used through the proxychains that forwards connection to internal network. This service allows connections from the DMZ so as the pentester connects to the system, he/she can abuse prepared vulnerability in Utilman.exe. All systems are interconnected in the domain BREW.EX. The main goal of the pentester is to compromise DC and dump the credentials of each domain user. The pentester gets the access to the DC by using the Pass the Hash attack in which he uses credentials of domain administrator who was logged in the files server to configure some stuff. The last step of the exercise is to create a Golden ticket that serves to sign kerberos tickets and grant the pentester access to the all domain services.

3 WINDOWS OS DEFAULT SETTINGS

The most of current ICT systems connected to the network uses name resolution to be capable of communicate between each other and to work properly as well. To give an example, if a client wants to communicate with a name address of "example.com", it must be done a name resolution in the network to set up a proper routing of data flow. This communication by default looks like following. The client who wants to communicate through the network firstly tries to resolve the name to the IP address using its cache memory. If the cache memory doesn't contain required record, the client sends the query to the DNS server. If neither DNS server does not resolve the name, the client sends the query across the whole network segment. To achieve this, it uses Link-Local Multicast Name Resolution (LLMNR), NetBIOS-NS and mDNS protocols. In a case that no system knows the answer, the resolutions fails. However if there is an attacker listening in the network, he/she accepts the query, responds to the client and deceive it that he/she knows the address as can be seen in Figure 2. The motivation behind this behaviour is gaining authentication data from the client who sent the query.

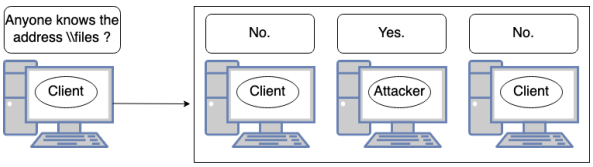

Figure 2: Client is sending the query to the network

3.1 INTERCEPTION OF AUTHENTICATION DATA

As it can be seen from Figure 2 if there is the attacker in the network, he can spoof an authoritative source resolving names to the IP addresses and force the clients to communicate with his/her system. The attacks aimed at these services are called LLMNR/NBT-NS Poisoning and Server Message Block (SMB) Relay and they are classified as Man-in-the-Middle (MitM) attacks.

The attack scenario focused on LLMNR/NBT-NS/mDNS is following. The attacker spoofs the authoritative source serving to resolve names in the network and answers to sent queries pretending he knows the identity of the destination system. By this manner he poisons mentioned services and the clients will communicate with him. If the destination service requires authentication, the client sends its username and NT Lan Manager (NTLM) hash to the address provided by attacker. The first option what the attacker could do with the gained hash is to crack it offline. The second option is to forward the credentials to the destination system and create a session with it allowing him to run a code. This option can be done within the poisoned communication or independently from it. See [1][2] for more details.

Figure 3 shows the process of poisoning. The client sent query for the name SERVICE and the attacker responded with poisoned answer. The client accepted spoofed answer and tried to authenticate to his bogus service. The attacker obtained his IP address, username and NTLMv2 hash that he can use for offline cracking.

The attack scenario focused on Web Proxy Auto-Discovery (WPAD) protocol works in similar way. Each modern web browser has an option to set up proxy. As the article is focused on Windows domain, we used up-to-date Microsoft Edge web browser which is a default web browser of Windows 10 OS. After entering the web address the browser always checks the proxy setttings. The default configuration is set to set up proxy automatically so with the each query for the new address there is a query sent to the network that is looking for proxy configuration file. If there is not DNS record for

```
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 172.16.0.3 for name SERVICE (service: File
Server)
[*] [LLMNR]  Poisoned answer sent to 172.16.0.3 for name service
[*] [MDNS] Poisoned answer sent to 172.16.0.3      for name service.local
[*] [MDNS] Poisoned answer sent to 172.16.0.3      for name service.local
[*] [LLMNR]  Poisoned answer sent to 172.16.0.3 for name service
[SMB] NTLMv2-SSP Client   : 172.16.0.3
[SMB] NTLMv2-SSP Username : WINDEV2008EVAL\User
[SMB] NTLMv2-SSP Hash     : User::WINDEV2008EVAL:4dfe7bebe0d8f40f:E6DA585E5D82C6
EE8B7649B84D0B7EDA:0101000000000000C0653150DE09D201D52EC3E8E21E06B10000000002000
80053004D004200330001001E00570049004E002D005000520048003400390032005200510041004
6005600040014005300 4D00420033002E006C006F00630061006C0003003400570049004E002D005
0005200480034003900320052005100410046005600 2E0053004D00420033002E006C006F0063006
1006C0005001400530 04D00420033002E006C006F00630061006C0007000800C0653150DE09D2010
6000400020000000800300030000000000000000100000000200000 17605C5FFE0262F3AA0A9E2CB
F8DC5011E8554E8AE1ACA94E979F630DC073F530A00100000000000000000000000000000000000
900180063006900660073002F00730065007200760069006300650000000000000000
```

Figure 3: Interception of NTLM hash using Responder

the address "wpad.domain.ex", the query is sent through the LLMNR/NBT-NS/mDNS protocols as can be seen in the Figure 4.

1	0.000000	192.168.0.235	213.46.172.38	DNS	69	Standard query 0x36be A wpad.home
2	0.000578	192.168.0.235	224.0.0.251	MDNS	70	Standard query 0x0000 A wpad.local, "QM" question
3	0.001037	fe80::192f:fc99:d0b…	ff02::fb	MDNS	90	Standard query 0x0000 A wpad.local, "QM" question
4	0.001397	192.168.0.235	224.0.0.251	MDNS	70	Standard query 0x0000 A wpad.local, "QM" question
5	0.001817	fe80::192f:fc99:d0b…	ff02::fb	MDNS	90	Standard query 0x0000 A wpad.local, "QM" question
6	0.002557				98	<Ignored>
7	0.003776				98	<Ignored>
8	0.013364	213.46.172.38	192.168.0.235	DNS	144	Standard query response 0x36be No such name A wpad.home SOA a.root-servers.net
9	0.013925	192.168.0.235	192.168.0.255	NBNS	92	Name query NB WPAD<00>
10	0.014212	192.168.0.235	224.0.0.251	MDNS	70	Standard query 0x0000 A wpad.local, "QM" question
11	0.014736	fe80::192f:fc99:d0b…	ff02::fb	MDNS	90	Standard query 0x0000 A wpad.local, "QM" question
12	0.015528	fe80::192f:fc99:d0b…	ff02::1:3	LLMNR	84	Standard query 0x4ec7 A wpad
13	0.015706	192.168.0.235	224.0.0.252	LLMNR	64	Standard query 0x4ec7 A wpad
14	0.015955	192.168.0.235	192.168.0.255	NBNS	92	Name query NB WPAD<00>

Figure 4: Interception of requests for WPAD in Wireshark

These queries are abused by the attacker who runs the HTTP server and answers the queries. The web victim's browser tries to connect to the attacker's HTTP server to download the *wpad.dat* file however this action requires the authentication what means that client sends its username and NTLM hash.

Figure 4 shows the common procedure what is done in the background when the user enters the address to the address bar. The procedure corresponds with the procedure described in Section 3.

3.2 Pass the Hash attack

This attack is focused on bypassing the authentication mechanism and most often is used against Windows OS. The attack depends on a Single Sign-On (SSO) functionality of authentication mechanism used by authentication protocols as NTLM or Kerberos. With SSO, users can enter their passwords once to be able to use resources they have been given rights to, without prompting them for their passwords again. For more details see [3].

In authentication process, the client's password is converted to the hash and after this action it is discarded. In the following steps it is used only the obtained NTLM hash. This is the essence of the Pass the Hash (PtH) attack. The attacker does not need to have knowledge of the user's password. Everything he needs to authenticate to the server is the knowledge of user password's hash and the username.

To perform PtH attack we chose Metasploit and it's module named psexec to perform the attack with. Psexec is the utility from the Sysinternals package that serves to administer systems remotely [4].

4 MITIGATION

The best way to mitigate these type of attack in which the sensitive data are leaked is to disable use of protocols like LLMNR, NBT-NS and mDNS. These protocols are used only in a case that the DNS server is not available or the DNS record does not exist but despite of this are these protocols enabled by default in Windows OS. All mentioned protocols are nowadays used only behalf of legacy systems in the network. So if there are no systems that run Windows 2000 and older, these protocols are not necessary [5]. The next recommendation linked to this threat is to set up the DNS record for the WPAD protocol.

The second step could be to completely disable NTLM authentication in the network and deploy protocols like Kerberos. Another option is to enforce SMB signing at all devices in the network. SMB signing serves to authenticate the source and the authenticity of each SMB packet. So if the authentication fails, the packet is dropped and this would mitigate the scenario of MitM attack focused on SMB service [2].

There are more recommendations how to mitigate this type of attacks as for example the network segmentation or the use of the model with the lowest privileges. See the explicit description of this model at [6].

5 CONCLUSION

This article described and analyzed the methods of Windows domain compromitation through the default Windows OS settings. The main objective of the article was to analyze the protocols of Windows OS that are enabled by default and could lead to the compromitation of the whole domain. Our motivation behind writing up this article was an annual increase of cyber attacks and an effort to teach users and administrators how to mitigate this simple attacks with possible critical impact.

REFERENCES

[1] SALVATI, M. (2017, June 2). *Practical guide to NTLM Relaying in 2017 (A.K.A getting a foothold in under 5 minutes)* [online]. [cit. 2020-11-25]. Available at: https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html

[2] KUEHN, E. (2018, April 11). *Ever Run a Relay? Why SMB Relays Should Be On Your Mind* [online]. [cit. 2020-11-25]. Available at: https://blog.secureideas.com/2018/04/ever-run-a-relay-why-smb-relays-should-be-on-your-mind.html

[3] EWAIDA, B. (2010, January 21). *Pass-the-Hash attacks: Tools and Mitigation* [online]. [cit. 2020-12-12]. Available at: https://www.sans.org/reading-room/whitepapers/testing/pass-the-hash-attacks-tools-mitigation-33283

[4] MICROSOFT: *Windows Name resolution.* Microsoft 29.06.2016 [online]. [cit. 2020-12-12]. Available at: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec

[5] MICROSOFT: *Windows Name resolution.* Microsoft 29.06.2016 [online]. [cit. 2020-12-12]. Available at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc728457(v=ws.10)?redirectedfrom=MSDN

[6] BEYOND TRUST: *Least Privilege.* BeyondTrust [online]. [cit. 2020-12-16]. Available at: https://www.beyondtrust.com/resources/glossary/least-privilege