Exploiting known vulnerabilities, misconfigurations and weaknesses in native protocols to compromise Windows Active Directory Domains with a focus on traceability and ease of use.

Søren Fritzbøger - s153753
Vejledt af Henrik Tange

20. Januar 2019

Danmarks Tekniske Universitet

Abstract

Abstract here

Table of contents

Abbreviations

AV Antivirus. 5

DC Domain Controller. 6

IDS Intrusion Detection System. 5

LLMNR Link-local Multicast Name resolution. 5, 6

NBNS NetBIOS Name Resolution. 5, 6

SIEM Security Information and Event Management. 5

1 Introduction

1.1 Problem background

When performing a pentest on Windows Active Directory environments, one of the goals is usually to obtain Domain Administrator privileges, usually in the form of a Domain Admin account. There are numerous ways to gain initial foothold in an Active Directory Domain, but the most common one is by exploiting the native protocols NetBIOS Name Resolution (NBNS) and Link-local Multicast Name resolution (LLMNR)[3] to gain crackable hashes which can be bruteforced offline. Hereafter a number of different techniques and exploits can be used to gain additional credentials, but it usually consists of dumping cached credentials on domain joined hosts in one way or another.

As one can easily figure out this is usually a manual job where many different tools are joined together to produce the right result. This usually means it is a trivial and easy job, which requires a lot of time that can be spent on more advanced tasks. This is often done with tools not written by the pentester themselves, which can pose a security risk as the tools can be backdoored or otherwise have security vulnerabilities.

One of the objectives of a pentest is usually to be as silent as possible and not trigger any alerts in any Intrusion Detection System (IDS), Security Information and Event Management (SIEM) or similar systems. The risk of using publicly available tools is therefore also that the tools are highly likely to be detected by Antivirus (AV) and will often trigger unwanted alerts.

Another issue of performing pentests is to document and remember the order of tasks that where done. A pentest usually concludes with a report where the necessary steps are explained to the customer, and this includes in what order tasks were done and which user credentials were used.

1.2 Problem brief

The purpose of this project is to determine whether a proper solution to the aforementioned problem can be found, and to analyze how such a tool can be developed. The problem is split in two, where the first goal is to gain an initial foothold and the second is to gain Domain Administrator privileges. To accomplish this the many techniques and methods will be discussed and evaluated in comparison to each other, and the most valid solution will be implemented in a piece of software that aims to be easy to use and contain a high level of traceability.

The developed piece of software must be able to be easy to use so that an incentive to use it instead of other tools is created. It should be constructed with AV evasion in mind, such that it will not be detected by AVs. Furthermore it must be designed with traceability in mind, such that a clear timeline can be constructed and documented.

1.3 Report structure

1.4 Pentesting Windows Domains

2 Initial foothold

In a Windows Active Directory Domain there are numerous ways of gaining an initial foothold. The following methods are the most used in modern penetration testing of Windows AD environments.

BRUTE User credential bruteforcing

SPRAY Password spraying

EXPL Exploiting known vulnerabilities on unpatched systems

CLEAR Clear text passwords stored on public shares

SPOOF NBNS and/or LLMNR spoofing

All of the above mentioned methods have their weaknesses and strengths, which should be taken into account when choosing the best method or methods to gain initial foothold in the domain. To make an educated guess of which method(s) to pursue further a comparison between the different methods is needed. Table 1 gives a comparison of the different methods, and shows which weaknesses and strengths each methods possess.

Strength	**BRUTE**	**SPRAY**	**EXPL**	**CLEAR**	**SPOOF**
Is it automatable?	+	+	-	-	+
Is it fast?	-	-	+	-	+
Account lockout issues?[1]	-	-	+	+	+
Communication with critical systems such as a DC?	-	-	+	+	+
Easy to detect?	-	-	+	+	+
Is it easy to do?	+	+	-	(+)	+
Points	2	2	4.5	5	7

Table 1: Comparison of different methods to gain initial foothold in a Windows AD environment

All of the above mentioned methods are valid and used by most hackers during penetration testing. Due to the nature of the project and the accompanying software

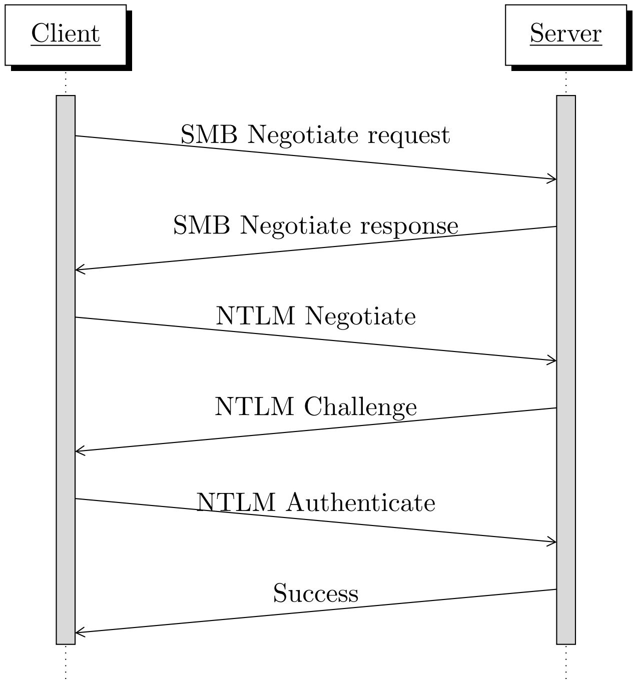

Figure 1: SMB NTLM authentication[2]

2.1 Spoofing

2.1.1 NBNS - NetBIOS Name Resolution

2.1.2 LLMNR - Link-local Multicast Name resolution

2.2 Credential acquiring

2.2.1 Credential types

LM
NT
NTLM
NetNTLMv1
NetNTLMv2

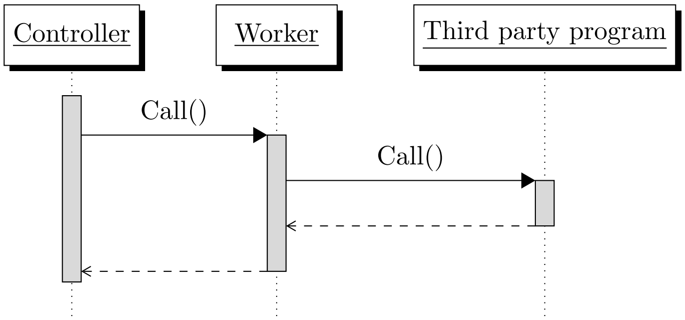

Figure 2: Test figure

2.2.2 SMB

2.2.3 HTTP

3 Attack methods

3.1 LSASS secrets

3.1.1 Impacket wmiexec

3.1.2 Mimikatz

3.2 secretsdump

4 Reconnaissance

5 Implementation

5.1 Technologies

5.1.1 ASP.NET Core

5.1.2 SignalR

5.1.3 VueJS

5.2 Considerations

5.2.1 Modularity

5.2.2 Ease of use

5.2.3 Traceability

5.3 Storage

6 Discussion

6.1 Ethics

7 Conclusion

List of Figures

References

[1] Microsoft. *Account lockout threshold.* URL: https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/account-lockout-threshold (visited on 11/17/2018).

[2] Microsoft. *NTLM Over Server Message Block (SMB).* URL: https://msdn.microsoft.com/en-us/library/cc669093.aspx (visited on 11/15/2018).

[3] Georgia Weidman. *Scenario-based pen-testing: From zero to domain admin with no missing patches required.* URL: https://www.computerworld.com/article/2843632/security0/scenario-based-pen-testing-from-zero-to-domain-admin-with-no-missing-patches-required.html (visited on 11/15/2018).