Kompetenzzentrum für angewandte
Sicherheitstechnologie
Karlsruher Institut für Technologie


KASTEL-Praktikum „Security“
Wintersemester 2017/2018

# SMB, das (Einfalls-)Tor zur Welt

Autor: Thomas Czogalik, Maximilian Schick
Matrikelnr.: 1696853, 1709879

Betreuer: Florian Patzer (IOSB), Dr. Christian Haas (IOSB)

# Erklärung

Wir versichern wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, Wintersemester 2017/2018

.....
Thomas Czogalik, Maximilian Schick

# Inhaltsverzeichnis


**1 Motivation** **1**

**2 Grundlagen** **2**
2.1 Server-Message-Block (SMB) . . . 2
2.1.1 Microsoft-SMB-Protokoll . . . 2
2.1.2 SMB-Verbindung . . . 3
2.1.3 SMB-Sicherheitsmodell . . . 5
2.1.4 NTLM . . . 5
2.2 Angriffsarten . . . 6
2.2.1 Remote-Code-Execution (RCE) . . . 6
2.2.2 Man-In-The-Middle (MITM) . . . 6
2.2.3 Denial-Of-Service (DOS) . . . 6
2.3 Metasploit . . . 8

**3 Angriffe auf SMB** **8**
3.1 Pass-The-Hash . . . 8
3.1.1 Einführung . . . 8
3.1.2 Angriffsszenario . . . 9
3.1.3 Technische Details . . . 9
3.1.4 Schutzmöglichkeiten . . . 9
3.2 SMBRelay . . . 10
3.2.1 Einführung . . . 10
3.2.2 Angriffsszenario . . . 10
3.2.3 Technische Details . . . 12
3.2.4 Schutzmöglichkeiten . . . 12
3.3 EternalBlue . . . 13
3.3.1 Einführung . . . 13
3.3.2 Angriffsszenarien . . . 13
3.3.3 Technische Details . . . 13
3.3.4 Schutzmöglichkeiten . . . 17
3.4 SMBLoris . . . 17
3.4.1 Einführung . . . 17
3.4.2 Angriffsszenario . . . 17
3.4.3 Technische Details . . . 18
3.4.4 Schutzmöglichkeiten . . . 18

**4 Evaluierung** **19**

**5 Zusammenfassung und Ausblick** **20**

# 1 Motivation

Sobald man Datei-, Druck- und andere Serverdienste in lokalen Rechnernetzen nutzen möchte, wird ein Netzwerkprotokoll benötigt. Ein Protokoll, das dies ermöglicht, ist das Server-Message-Block-Protokoll (SMB). SMB wird in nahezu jedem privaten und firmeninternen Netz verwendet. Jedes Windows-System enthält eine Microsoft Implementierung des Protokolls. Auf Unix basierten Systemen wird der Zugriff auf Ressourcen mithilfe des freien Softwarepaket Samba ermöglicht.

Viele große Angriffe aus den letzten Jahren nutzten Sicherheitslücken im SMB-Protokoll aus. Einer dieser Angriffe war der Sony Picture Hack vom 24. November 2014 [1]. Bei diesem veröffentlichte eine Hackergruppe vertrauliche Daten aus dem Sony Pictures Filmstudio. Zu den Daten gehörten persönliche Informationen über die Mitarbeiter, E-Mails zwischen Mitarbeitern, Informationen über die Gehälter der Führungskräfte und Kopien von damals unveröffentlichten Sony-Filmen.

Ein weiterer SMB bezogener Angriff ereignete sich im Mai 2017. Hierbei nutzten Hacker eine Sicherheitslücke im SMB-Protokoll, um das Schadprogramm WannaCry auf fremden Rechnern einzuschleusen [2]. Bei WannaCry handelt es sich um eine Ransomware, also ein Schadprogramm, das Dateien auf dem befallenen Rechner verschlüsselt und Lösegeld für die Entschlüsselung fordert. Die Malware infizierte rund 200.000 Rechner in 150 Ländern, wie in Abbildung 1 zu sehen ist. Darunter befanden sich auch Systeme mehrerer global tätiger Unternehmen wie Server und medizinische Ausrüstung des britischen National Health Service sowie der US-Logistikkonzern FedEx. In Deutschland war hauptsächlich die Deutsche Bahn betroffen.

Mit dieser Arbeit erstellen wir einen Überblick über mehrere Angriffe und die damit einhergehenden Sicherheitslücken des SMB-Protokolls. Durch falsche Konfigurationen oder nicht aktualisierte Systeme, können Angreifer weiterhin altbekannte Sicherheitslücken ausnutzen. Dies führt dazu, dass diese Angriffe trotz ihres Alters immer noch gefährlich sein können. Dabei werden wir uns auf die Microsoft Implementierung des SMB-Protokolls beschränken, da diese am weitesten verbreitet ist. Bei der Untersuchung haben wir versucht, die folgenden Fragen zu beantworten:

- Zu welcher Angriffsart gehören die verschiedenen Angriffe auf das SMB-Protokoll?
- Wie funktionieren diese?
- Wie sieht ein verwundbares System aus? Welche Windows Versionen sind betroffen?
- Wie kann man sich schützen? Wie muss das System konfiguriert werden? Muss ein Patch installiert werden?

Abschließend wird evaluiert, ob die Angriffe, die teilweise über 20 Jahre alt sind, heute noch von Bedeutung sind. Dazu wurden Testsysteme mit unterschiedlichen Betriebssystemen, in verschiedenen Versionen aufgesetzt. Mithilfe von frei verfügbaren Implementierungen der Exploits, die zum Aufzeigen einer Sicherheitslücke entwickelt wurden, soll festgestellt werden, ob beim Betreiben eines bestimmten Betriebssystems die jeweilige Sicherheitslücke ausgenutzt werden kann.

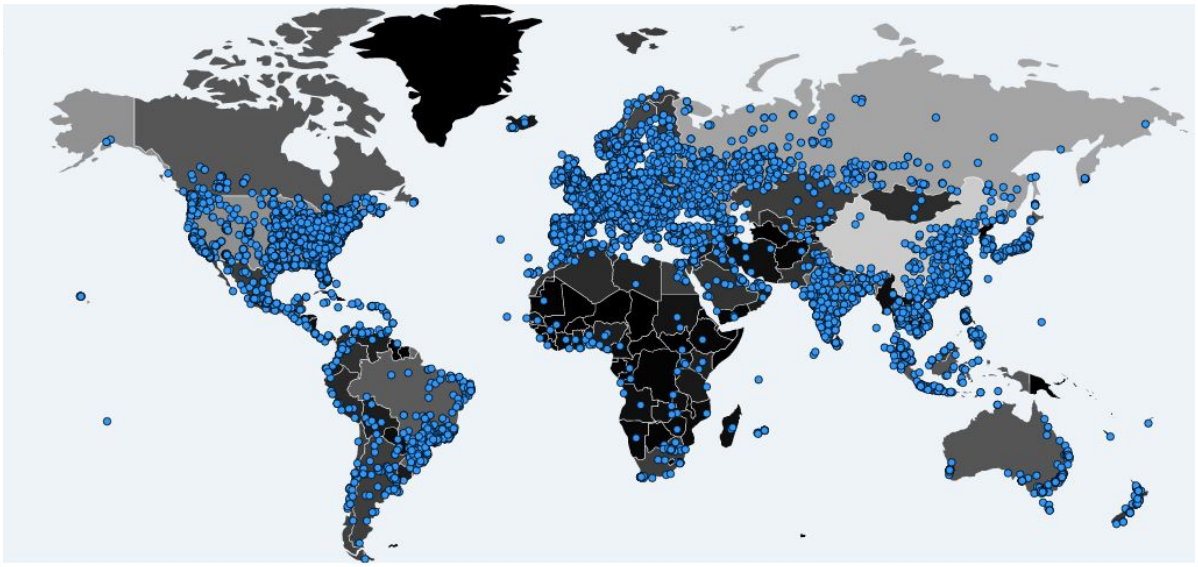

Abbildung 1: Ausbreitung von WannaCry [2].

Die Arbeit ist wie folgt aufgebaut: In Abschnitt 2 werden für diese Arbeit wichtige Grundlagen erläutert. Abschnitt 3 behandelt die in dieser Arbeit betrachteten Angriffe auf das SMB-Protokoll. In Abschnitt 4 werden verschiedene Testsysteme mithilfe von Exploits angegriffen und evaluiert. Schließlich fasst Abschnitt 5 die Arbeit zusammen.

# 2 Grundlagen

## 2.1 Server-Message-Block (SMB)

Das SMB-Protokoll (Server-Message-Block) ist ein Netzwerkprotokoll auf Anwendungsebene. Es wird hauptsächlich für den gemeinsamen Zugriff auf Dateien, Drucker, serielle Schnittstellen und sonstige Kommunikation zwischen Knoten in einem Netzwerk verwendet. Die bekanntesten Implementierungen des Protokolls sind SAMBA [3] und das Microsoft-SMB-Protokoll [4]. Bei SAMBA handelt es sich um eine freie multiplattform Implementierung des SMB-Protokolls. Es wird hauptsächlich auf Unix-Systemen verwendet. Die Microsoft Implementierung ist dagegen in allen Windows-Systemen enthalten. Im Folgenden wollen wir uns auf die Microsoft Implementierung beschränken.

### 2.1.1 Microsoft-SMB-Protokoll

Die erste SMB-Version (SMB1) stammt aus dem Jahr 1990. SMB2 wurde 2006 mit Windows Vista eingeführt. 2012 wurde SMB3, die mittlerweile aktuellste SMB-Version, mit Windows 8 veröffentlicht. Neben SMB3 sind auch die älteren Versionen, SMB1 und SMB2, standardmäßig auf jeder aktuellen Windows Version installiert. Das mehrere Versionen parallel installiert sind, hat den Grund, dass damit auch mit Geräten kommuniziert werden kann, die ausschließlich eine ältere Version des Protokolls unterstützen. Ein

Angreifer kann dies aber ausnutzen und sein Opfer dazu bringen über SMB1 mit ihm zu kommunizieren. Dadurch erhält ein Angreifer viel mehr Informationen über den Rechner, mit dem er kommunizieren will, da SMB1 über 100 Kommandos zur Kommunikation zur Verfügung stellt. Erst mit SMB2 wurde die Kommunikation eingeschränkt, indem aus über 100 Kommandos 19 wurden [5]. Außerdem wurden weitere Sicherheitsmaßnahmen implementiert, um den Anwender vor verschiedenen Angriffen zu schützen. Microsoft rät deshalb dazu SMB1 nicht mehr zu verwenden und zu deinstallieren [6]. Deshalb ist seit dem Windows 10 Fall Creators Update und der Windows Server Version 1709 SMB1 nicht mehr standardmäßig installiert [7].

### 2.1.2 SMB-Verbindung

Im OSI-Netzwerkmodell lässt sich das SMB-Protokoll in die Anwendungs- oder Präsentationsschicht einordnen. Zum Transport verwendet es untergeordnete Protokolle. Eine Verbindung zwischen den Kommunikationspartnern wird meist direkt über den TCP-Port 445 aufgebaut oder bei Verwendung der NetBIOS-Api über die TCP-Ports 137 und 139. Nachdem die Kommunikationspartner verbunden sind, kann nun eine SMB-Session aufgebaut werden. Dies ist in Abbildung 2 veranschaulicht. Zunächst müssen der Client und der Server einen SMB-Dialekt wählen [8]. Dieser Dialekt beschreibt die höchste Funktionalität, die sowohl der Client als auch der Server unterstützen. Dies geschieht indem der Client eine Anfrage an den Server sendet, um einen Dialekt auszuhandeln (1). Dabei hängt er der Anfrage zusätzlich eine Liste mit seinen unterstützten Dialekten an. Im nächsten Schritt (2) entscheidet sich der Server für einen Dialekt und teilt diesen dem Client mit. Nun signalisiert der Client, dass er sich beim Server authentifizieren will (3). Die Authentifizierung wird in Abschnitt 2.1.4 genauer betrachtet. Ist die Authentifizierung erfolgreich, signalisiert der Server dies dem Client und schickt ihm eine UID, mit der er ihn eindeutig identifizieren kann (4). Schlägt die Authentifizierung fehl, sendet der Server hier einen Fehlercode. Nach einer erfolgreichen Authentifizierung haben der Client und der Server eine SMB-Session aufgebaut. Jetzt kann der Client beispielsweise versuchen auf eine freigegebene Datei zuzugreifen. Zunächst muss er dem Server dazu signalisieren, dass er Zugriff auf eine bestimmte Freigabe haben will (5). Wird ihm diese gewährt erhält er eine TID (Tree ID) (6). Eine TID stellt eine offene Verbindung zu einer Freigabe dar. Eine offene TID muss innerhalb einer SMB-Verbindung eindeutig sein. Der Client kann den Server nun auffordern die Freigabe zu öffnen (7). Wenn der Client dazu berechtigt ist, sendet der Server eine ID um die Freigabe identifizieren zu können (8). Schließlich kann der Client den Server auffordern diese zu lesen (9). Der Server öffnet die Freigabe und schickt den Inhalt an den Benutzer (10).

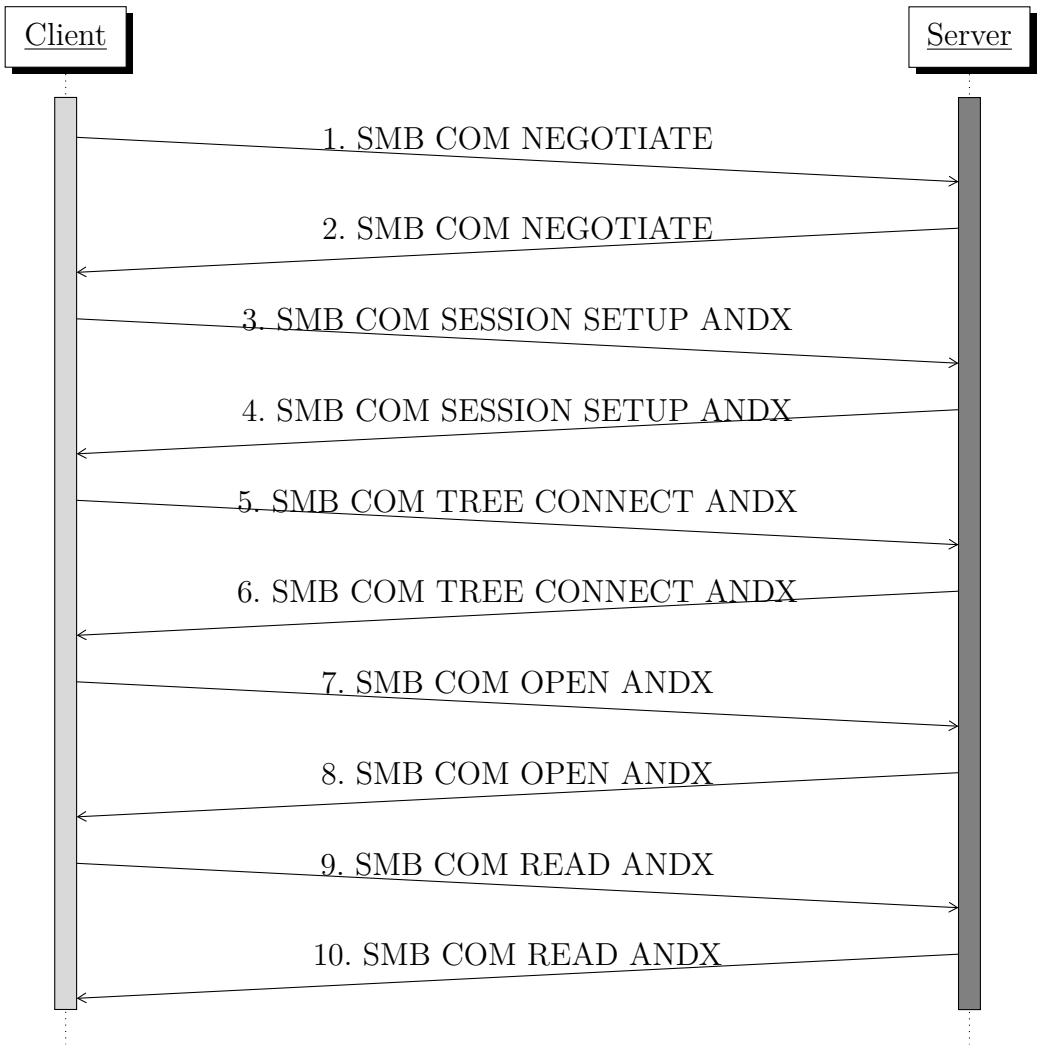


Abbildung 2: Darstellung einer SMB-Verbindung mit Öffnen einer Freigabe, basierend auf [9]

### 2.1.3 SMB-Sicherheitsmodell

Das Sicherheitsmodell unterscheidet zwischen zwei Arten der Authentifizierung um auf Freigaben (Shares) zuzugreifen [10]. Eine Freigabe ist hierbei eine Datei, ein Verzeichnis oder ein Drucker.

- Benutzer-Authentifizierung: Damit können Systemadministratoren gezielt festlegen, welche Benutzer und Gruppen auf eine Freigabe zugreifen dürfen. Der Benutzer, der versucht auf eine Freigabe zuzugreifen, muss einen Benutzernamen und ein Kennwort angeben. Nachdem er sich authentifiziert hat, kann er auf alle Freigaben zugreifen, die für ihn freigegeben sind und nicht ebenfalls durch eine Freigabe-Authentifizierung geschützt sind.

- Freigabe-Authentifizierung: Im Gegensatz zur Benutzer-Authentifizierung wird hier keine Benutzeridentität festgelegt. Der Zugriff auf Freigaben wird durch ein Passwort gesteuert. Möchte man auf eine Freigabe zugreifen, muss man das jeweilige Passwort eingeben.

### 2.1.4 NTLM

NTLM (NT LAN Manager) [11] ist eines der möglichen Authentifizierungsverfahren des SMB-Protokolls. Es verwendet ein Challenge-Response-Protokoll. Im Folgenden wird eine Benutzer-Authentifizierung mithilfe von NTLM betrachtet. An dieser sind in der Regel drei Systeme beteiligt: ein Client, ein Server und ein Domänen-Controller. Hierbei möchte sich der Client beim Server authentifizieren. Der Domänen-Controller ist beteiligt, da er alle Anmeldedaten der Domäne kennt. Diese bestehen hierbei aus einem Benutzernamen und dem Passwort, des zu authentifizierenden Benutzers. Möchte sich der Client mittels eines lokalen Nutzers des Servers anmelden, kann auch der Server die Rolle des Domänen-Controllers übernehmen. Während der Authentifikation wird niemals das Passwort in Klartext durch das Netzwerk übertragen. Stattdessen muss der Client eine Berechnung durchführen, die beweist, dass er Zugriff auf die benötigten Anmeldeinformationen hat.

Abbildung 3 zeigt den Ablauf einer erfolgreichen NTLM-Authentifizierung. Zunächst sendet der Client Alice dem Server Bob ihren Benutzernamen zu und signalisiert damit, dass sie sich bei ihm authentifizieren möchte. Bob generiert daraufhin eine zufällige Bitfolge und schickt sie Alice zurück. Diese Bitfolge ist die Challenge. Im Folgenden muss Alice mit Hilfe ihres Passwort-Hashes und der Challenge eine Response berechnen. Mit dieser Response muss Alice Bob beweisen, dass sie der Client ist, für den sie sich ausgibt. Für die Berechnung wird bei NTLM, je nach Version, die NTOWF[1] oder LMOWF[2] Funktion verwendet. Nachdem Alice die Response berechnet hat, schickt sie diese an Bob. Nachdem Bob die Response erhalten hat, muss er die Anmeldung verifizieren. Dazu sendet er den Benutzernamen, die Challenge und die Response weiter an den Domänen-Controller. Dieser holt sich zunächst den Hash des Benutzerpassworts aus der Datenbank des Account-Managers. Im Anschluss berechnet er selbst die Response

[1] https://msdn.microsoft.com/en-us/library/cc236699.aspx
[2] https://msdn.microsoft.com/en-us/library/cc236700.aspx

und vergleicht sein Ergebnis mit dem von Alice. Stimmen diese überein, hat Alice sich erfolgreich beim Server authentifiziert. Stimmen die Ergebnisse nicht überein, wird ihr der Zugriff verweigert.

## 2.2 Angriffsarten

### 2.2.1 Remote-Code-Execution (RCE)

Unter Remote-Code-Execution versteht man die Möglichkeit auf einem fremden System Programmcode auszuführen. Dadurch kann beispielsweise Schadsoftware installiert werden oder eine Backdoor eingerichtet werden. Es ist somit sogar möglich, die vollständige Kontrolle über das betroffene System zu übernehmen. Angriffe, die Remote-Code-Execution erlauben, gehören somit zur mächtigsten Angriffsart.

Angriffe mit dieser Möglichkeit sind meistens durch Sicherheitslücken innerhalb verwendeter Programme bedingt. Somit ist es empfehlenswert, diese stets aktuell zu halten. Bei wichtigen Netzwerken, die besonderen Schutz benötigten, sollte man zusätzlich einen Überblick über bekannte Exploits, innerhalb verwendeter Programme, haben. Diese sollten anschließend getestet werden und falls Angriffe gelingen, sollten Gegenmaßnahmen getroffen werden. Hiermit kann die Wahrscheinlichkeit erhöht werden, dass diese das Netzwerk nicht beeinträchtigen können.

### 2.2.2 Man-In-The-Middle (MITM)

Ein Man-In-The-Middle-Angriff (MITM) ist ein Angriff, bei dem der Angreifer heimlich die Kommunikation zwischen Kommunikationspartnern, die glauben direkt miteinander zu kommunizieren, abfängt, weiterleitet und möglicherweise liest oder verändert [12]. Um sich vor einem MITM-Angriff zu schützen, hat ein Anwender wenig Spielraum, da die meisten Schutzmaßnahmen serverseitig unternommen werden müssen.

Eine Verteidigungsmöglichkeit, um sich vor dem Mitlesen zu schützen, ist das Verwenden einer starken Verschlüsselung zwischen den Kommunikationspartnern. Dafür muss eine gegenseitige Authentifizierung stattfinden. Im Anschluss kann ein verschlüsselter Kanal aufgebaut werden. Um sicherzugehen, dass die Kommunikation nicht manipuliert wird, können die Nachrichten von dem jeweiligen Kommunikationspartner signiert werden.

### 2.2.3 Denial-Of-Service (DOS)

Denial-Of-Service (DOS) bezeichnet die Nichtverfügbarkeit eines Dienstes, obwohl dieser verfügbar sein sollte [13]. Häufig ist ein DOS die Folge eines Angriffs. Dabei wird die Maschine, die den Dienst anbietet, mit so vielen Verbindungsversuchen überflutet, dass sie den Dienst vorübergehend einstellt.

Mithilfe von strengeren Firewall Einstellungen kann das Risiko eines Angriffs reduziert werden. Dazu zählt zum Beispiel die Beschränkung der Bandbreitennutzung auf authentifizierte Benutzer. Auch Überwachungstools, die im Netzwerk nach ungewöhnlichen Aktivitäten Ausschau halten, können dabei helfen, im Falle eines Angriffs schnell reagieren zu können.

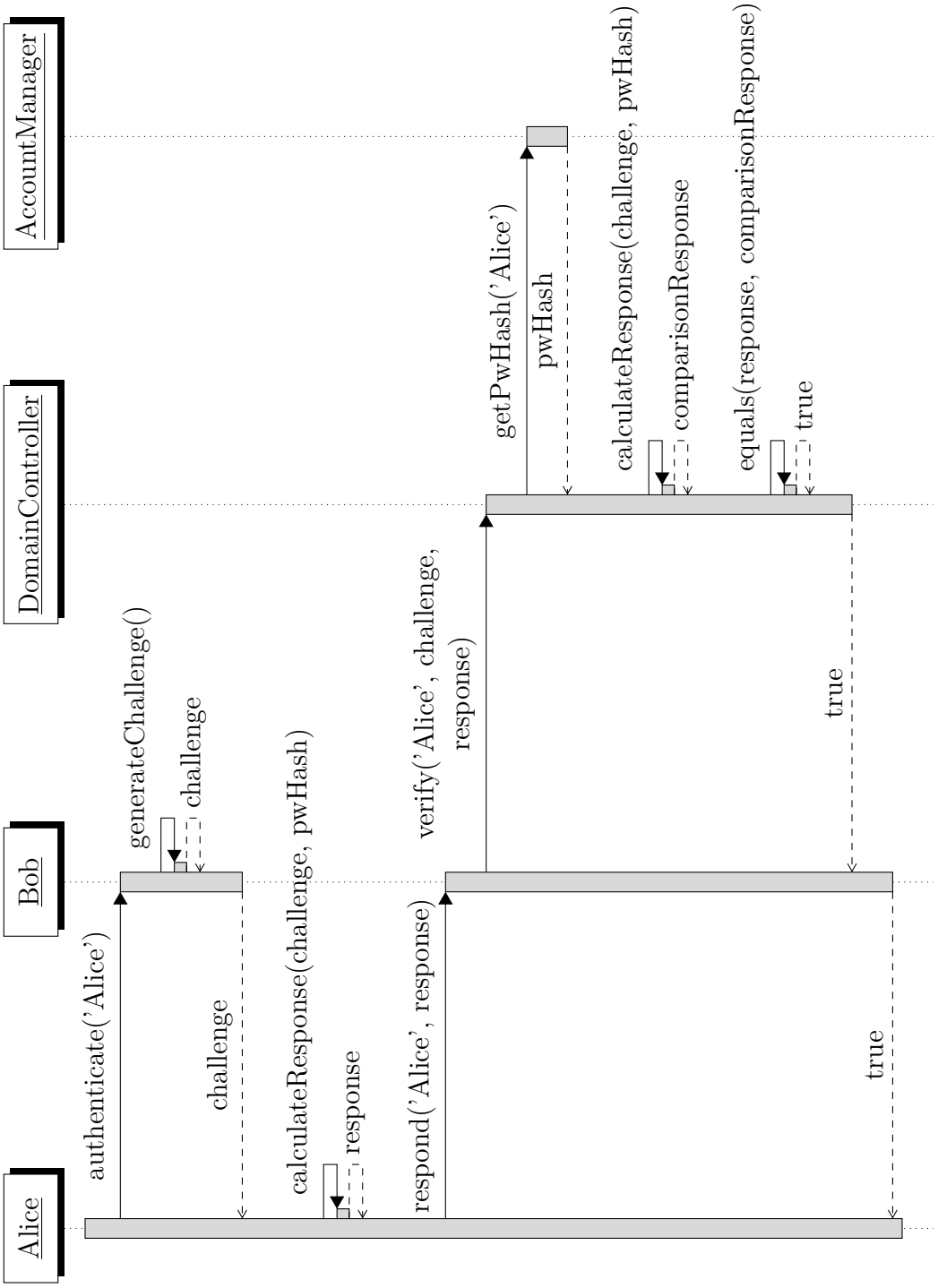


Abbildung 3: Darstellung eines erfolgreichen NTLM-Authentifizierungsprozesses.

## 2.3 Metasploit

Metasploit[3] ist ein freies Open-Source Projekt der IT-Sicherheitsfirma Rapid7. Der Hauptbestandteil des Projekts ist das Metasploit-Framework. Dieses ist ein Werkzeug, welches Penetration-Tests ermöglicht. Es besteht aus vielen einzelnen Modulen, die erlauben, bekannte Exploits, wie in einem Baukastenprinzip, zu konfigurieren und auf ein erreichbares System anzuwenden. Für die meisten Module reicht es aus, ausschließlich Netzwerk bedingte Informationen wie IP-Adressen anzugeben. Somit ist es möglich bekannte Systeme auf Sicherheitslücken gegen bekannte Exploits zu testen. Aufgrund des Baukastenprinzips wird in den meisten Fällen kein detailliertes Wissen über die einzelnen Exploits benötigt. Bei der Evaluierung der Angriffe wird unter anderem das Metasploit-Framework verwendet.

# 3 Angriffe auf SMB

Im Folgenden sollen einige der bekanntesten SMB Angriffe genauer vorgestellt werden. Dazu wird jeweils zunächst eine Einführung über die Herkunft des Angriffs gegeben. Danach werden mögliche Angriffsszenarien beschrieben. Anschließend wird erklärt, warum der Angriff möglich ist. Abschließend werden Möglichkeiten genannt, wie man sich schützen kann.

## 3.1 Pass-The-Hash

### 3.1.1 Einführung

Bei Pass-The-Hash handelt es sich um einen Angriff auf das Challenge-Response-Authentifizierungsprotokoll NTLM von SMB. Dieser ermöglicht eine Authentifikation ohne ein dafür notwendiges Passwort in Klartext zu besitzen. Grund hierfür ist ein Designfehler innerhalb NTLM, da zur Authentifizierung das Passwort ausschließlich in gehashter Form benötigt wird. Die Theorie hinter diesem Angriff wurde erstmals 1997 von Paul Ashton veröffentlicht [14].

Dieser Angriff wird oft in Kombination mit dem Werkzeug PsExec[4] verwendet. Dies ist Teil von PsTools[5], einer Sammlung von Befehlszeilenprogramme zur Systemadministration. Nach einer vorangegangener Authentifikation ermöglicht PsExec, Programme auf einem fremden Client über SMB auszuführen. Hierfür muss im Vorfeld nichts auf dem Client installiert werden. Mit dem Passwort-Hash eines lokalen Administrators ist es möglich sich bei diesem System zu authentifizieren und Code auszuführen. Es wird also Remote-Code-Execution erreicht. Für PsExec in Kombination mit Pass-The-Hash existiert ein Metasploit-Modul[6].

[3] https://www.metasploit.com/
[4] https://technet.microsoft.com/de-de/sysinternals/bb897553.aspx
[5] https://technet.microsoft.com/de-de/sysinternals/pstools.aspx
[6] https://www.offensive-security.com/metasploit-unleashed/psexec-pass-hash/

### 3.1.2 Angriffsszenario

Im Folgenden soll ein mögliches Angriffsszenario zeigen, wie eine Angreifer mittels Pass-The-Hash ein gesamtes Netzwerk übernehmen kann [15].

1. Ein beliebiger Benutzer mit lokalen Administratorrechten infiziert versehentlich sein Gerät. Dies kann mittels Malware innerhalb einer Phishing-Mail passieren oder beispielsweise durch das Verwenden eines gefundenen USB-Sticks.

2. Wenn sich nun zu einem späteren Zeitpunkt ein Domänen-Administrator lokal an dem Gerät anmeldet, um beispielsweise das Gerät zu konfigurieren, kann die Malware dessen Passwort-Hash stehlen.

3. Mit dem Passwort-Hash eines Domänen-Administrators kann sich der Angreifer nun an allen Geräten in der Domäne Administratorrechte verschaffen.

4. Der Angreifer hat somit das Netzwerk übernommen.

### 3.1.3 Technische Details

Das zuvor vorgestellte Angriffsszenario funktioniert, da nach dem lokalen Anmelden der Passwort-Hash eines Benutzers im Arbeitsspeicher des Geräts abgelegt wird. Der Grund hierfür ist das Komfortprinzip Single-Sign-On in Microsoft Windows. Dieses besagt, dass nach einem einmaligen Anmelden der Benutzer dauerhaft Zugriff auf seine Rechte hat. Um dies zu erlauben müssen für die Authentifikation notwendige Anmeldedaten zeitweise auf dem Rechner gespeichert werden. Windows ist standardmäßig so konfiguriert, dass die Daten der letzten zehn angemeldeten Benutzer aufbewahrt werden [16]. Mithilfe lokaler Administratorrechte und bestimmter Tools ist es somit immer möglich einen Passwort-Hash herauszulesen.

Wie in der Abbildung 3 ersichtlich, wird an keinem Punkt innerhalb des Authentifizierungsprozesses das Passwort in Klartext benötigt. Somit kann der Hashwert direkt genutzt werden, ohne zuvor durch hohen rechnerischen Aufwand aus dem Hashwert den Klartext des Passwortes zurückzurechnen.

### 3.1.4 Schutzmöglichkeiten

Es existiert keine einfache Möglichkeit, um ein Netzwerk gezielt vor diesem Angriff zu schützen. Grundsätzlich sollte man sparsam mit den Rechten der einzelnen Benutzer umgehen und immer beachten mit welchen Rechten man sich an welchem Gerät anmeldet. Beispielsweise sollten Nutzer nur dann lokale Administratorrechte bekommen, wenn sie diese wirklich benötigen. Ebenfalls sollte sich ein Domänen-Administrator niemals lokal an einem Gerät anmelden. Es sollte stets ein Benutzer mit lokalen Administratorrechten benutzt werden, wenn Änderungen vorgenommen werden müssen [16].

Aktuelle Windows Versionen verfügen mittlerweile dennoch über Features, die Pass-The-Hash Angriffe erschweren. So verwendet Windows 10 eine Virtualisierungstechnik, den sogenannten Credential-Guard [17], um den Zugriff auf zwischengespeicherte

Hash-Werte zu blockieren. Zusätzlich verhindern spezielle Einträge in der Windows-Registrierdatenbank, dass sich ein lokaler Administrator von einem anderen System aus über SMB als Administrator authentifizieren kann [18].

## 3.2 SMBRelay

### 3.2.1 Einführung

Der SMBRelay Angriff stammt bereits aus dem Jahr 2001. Dabei handelt es sich um einen MITM-Angriff. Ziel des Angreifers ist es, sich beim Opfer (Server) als ein bereits bekannter Benutzer (Client) anzumelden. Dabei kommuniziert der Angreifer mit dem Client, als wäre er der Server und mit dem Server, als wäre er der Client. Der Angriff wurde vom Hacker `Sir Dystic` der Hacker Organisation `Cult of the Dead Cow` veröffentlicht. Ein Patch seitens Microsoft wurde erst sieben Jahre später veröffentlicht. Trotz des Patches kann die Schwachstelle, unter bestimmten Umständen, immer noch ausgenutzt werden. Mithilfe des SMBRelay-Metasploit-Moduls[7] kann der Angriff, mit ein wenig Vorbereitung, reproduziert werden.

### 3.2.2 Angriffsszenario

Beim SMBRelay Angriff fügt sich der Angreifer in die Mitte der Verbindung zwischen Client und Server ein. Dies ist in Abbildung 3 veranschaulicht. Angreifer Mallory gibt sich dem Client Alice gegenüber als eigentlicher Verbindungspartner aus. Alle Anfragen von Alice an den Server Bob werden von Mallory an Bob weitergeleitet. Diesem gibt sich Mallory als Alice aus. Zu Beginn muss Mallory darauf warten, dass Alice sich bei Bob authentifizieren will. Nachdem sie die Anfrage abgefangen hat, leitet sie diese an Bob weiter und gibt sich dabei als Alice aus. Bob generiert eine Challange für Alice und schickt diese an Mallory. Mallory leitet nun die Challenge an Alice weiter, damit diese eine Response berechnet. Nachdem Alice ihre Response berechnet hat, sendet sie diese zurück an Mallory. Mallory leitet diese an Bob weiter. Bob überprüft Benutzername, Challenge und die Response, wie bereits in Abbildung 3 beschrieben. Nach erfolgreicher Überprüfung gewährt er Mallory Zugriff auf die Freigaben für Alice. Im letzten Schritt sendet Mallory eine Fehlernachricht an Alice.

[7] https://www.rapid7.com/db/modules/exploit/windows/smb/smb_relay

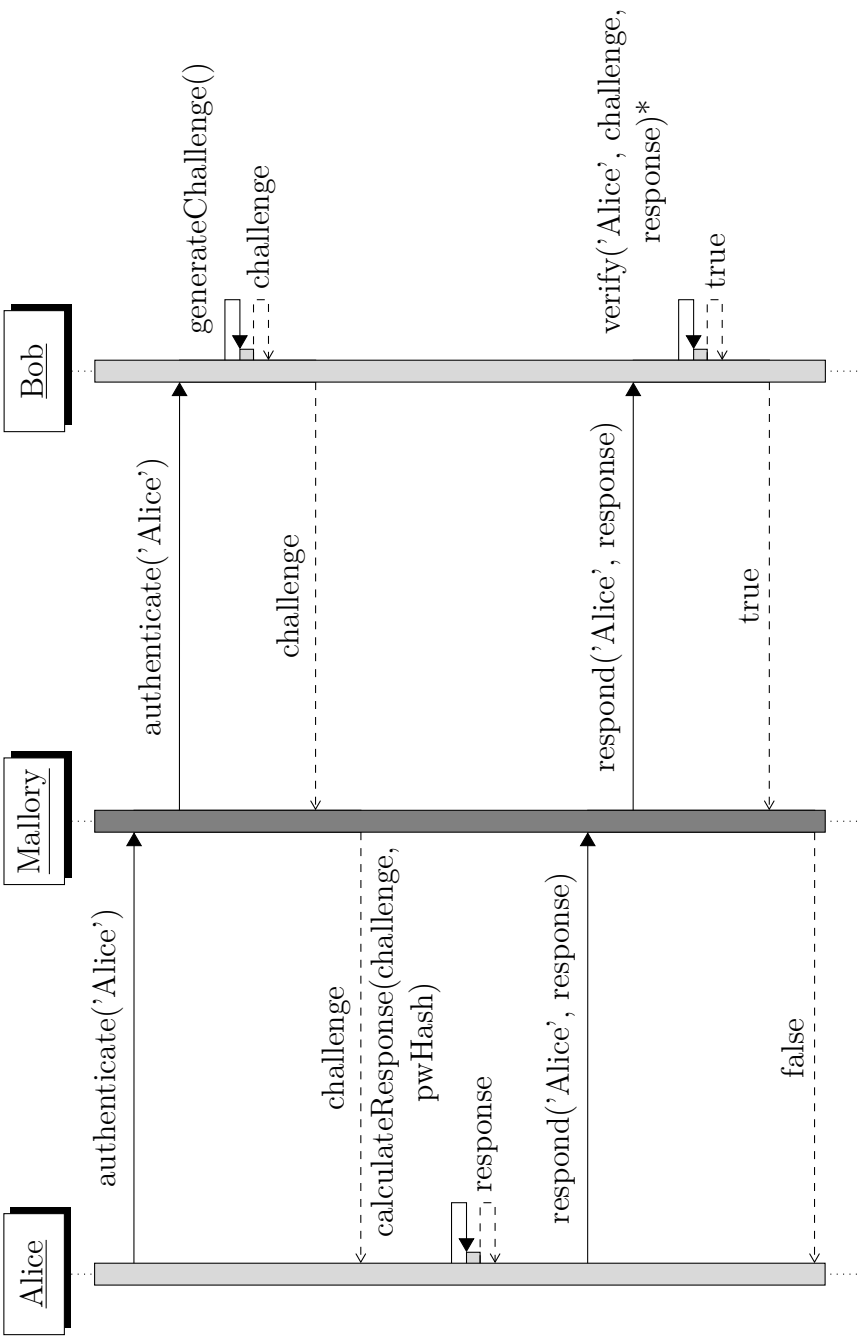


Abbildung 4: Darstellung eines erfolgreichen SMBRelay Angriffs.
* siehe Abbildung 3

### 3.2.3 Technische Details

SMBRelay nutzt Designfehler des SMB-Protokolls aus. Konkret werden Schwächen des Authentifizierungsprotokolls NTLM ausgenutzt. Dabei leitet der Angreifer die Kommunikation über sich um und kann diese auch verändern, falls nötig. Der SMBRelay Angriff läuft auf Port 139. Das liegt daran, dass SMB vor Windows 2000 mit NetBIOS über den TCP/IP-Port 139 lief. Daher war eine NetBIOS-Sitzung erforderlich, um eine SMB-Verbindung aufzubauen. Da es sich bei Port 139 um einen privilegierten Port handelt, benötigt der Client Administratorrechte. Der Angriff funktioniert auch unter dem TCP-Port 445, der Client benötigt aber auch hier Administratorrechte. Noch dazu benötigt er Rechte um in den Administrativen Freigabeordner `ADMIN$` schreiben zu dürfen und um neue Windows Dienste zu erstellen und zu starten. Wenn das alles vorliegt, kann ein Angreifer, wie im zuvor dargestellten Szenario, sich als jemand anderes bei seinem Opfer anmelden. Im nächsten Schritt kann er eine ausführbare Datei erstellen, die sich wie ein Windows Dienst verhalten soll. Dieser Dienst enthält zusätzlich Shellcode, der dem Angreifer schließlich eine Shell auf dem Opfer anbieten soll. Die Datei wird zunächst im `ADMIN$` Ordner abgelegt. Der enthaltene Dienst kann nun mithilfe des Service-Control-Managers gestartet werden. Der Shellcode wird ausgeführt, stoppt und löscht den Dienst und bietet dem Angreifer eine Shell auf dem Opfer an.

### 3.2.4 Schutzmöglichkeiten

Da der Angriff Designfehler des SMB-Protokolls ausnutzt, gibt es kein Allheilmittel. Sowohl der erste Patch aus dem Jahr 2008, als auch alle folgenden, die sich diesem Angriff widmen, lieferten nur einen kurzfristigen Schutz. Es gibt jedoch einige Möglichkeiten, um die Wahrscheinlichkeit eines Angriffes zu minimieren:

- Wähle ein sichereres Authentifizierungsprotokoll, zum Beispiel Keberos
- Das NTLM-Protokoll deaktivieren. Dies schützt vor allem gegen einen Angriff mit dem SMBRelay-Metasploit-Modul.
- Aktiviere SMB-Signing [19] um sicherzugehen, dass SMB-Pakete während der Übertragung nicht manipuliert werden. Dabei wird aber die Abwärtskompatibilität eingeschränkt.
- Da der Angriff einen Client mit Administratorrechten erfordert, sollten im Netzwerk verschiedene Domänen mit unterschiedlichen Sicherheitslevel eingerichtet werden. Jede Domäne bekommt dann nur die nötigen Rechte und die mögliche Angriffsfläche verkleinert sich dadurch.

## 3.3 EternalBlue

### 3.3.1 Einführung

EternalBlue ist ein Exploit, der mehrere Programmierfehler im SMB-Protokoll von Microsoft ausnutzt. Der Exploit erlaubt es dem Angreifer Schadcode auf fremden Rechner auszuführen. Er ist somit in der Lage den attackierten Rechner, sowie alle enthaltenen Daten zu kompromittieren. EternalBlue erfuhr im Mai 2017 großes mediales Interesse, als die Schwachstelle in Kombination mit dem Schadprogramm WannaCry ausgenutzt wurde. Im Common-Vulnerabilities-And-Exposures Industriestandard wird diese Schwachstelle als `CVE-2017-0144` bezeichnet [20].

Am 14. April 2017 erstellte die Hackergruppe `The Shadow Brokers` einen Blogeintrag mit dem Titel `Lost in Translation` [21]. In diesem veröffentlichten sie das Fuzzbunch-Framework, welches angeblich von der NSA nahen Equation-Group geschrieben wurde. Dieses enthielt ausführbare Versionen von mehreren bis dahin unbekannten Exploits. Unter diesen befand sich auch ein Modul, welches EternalBlue ausnutzt, um auf anfälligen Systemen eine Hintertür einzurichten. Durch die Veröffentlichung wurde der Exploit erstmals publik. Der Aussage eines NSA Mitarbeiters zufolge benutzte der Geheimdienst die Sicherheitslücke aber schon seit mehreren Jahren [22].

Durch Aufzeichnungen des vom EternalBlue-Modul erzeugten Datenpaketverkehr konnte das RiskSense-Cyber-Security-Research-Team die Funktionsweise nachvollziehen [23]. Mit diesem Wissen konnten sie den Exploit für das Metasploit-Framework portieren. Systemadministratoren können mit dem portierten Modul[8] ihre zugewiesene Systeme auf die EternalBlue Schwachstelle testen, ohne versehentlich eine Backdoor einzubauen. Das Modul ist dabei genauso wie das Vorbild aus dem Fuzzbunch-Framework nur mit Microsofts Windows XP (Server 2003) und Windows 7 (Server 2008 R2) kompatibel. Der Risikoanalyse vom RiskSense-Cyber-Security-Research-Team zufolge existiert aber auch ein nicht veröffentlichtes Modul für Windows 10. Deshalb wird im Folgenden nur die Windows 7 Implementierung betrachtet.

### 3.3.2 Angriffsszenarien

Durch EternalBlue kann ein System in jedem Netzwerk, das SMB nutzt, beliebigen Code auf jedem anderen System im Netzwerk ausführen. Durch die hohe Verbreitung von SMB gilt EternalBlue somit als ein sehr mächtiger Exploit. In Verbindung mit WannaCry wurde der Exploit genutzt, sobald ein Rechner in einem fremden lokalen Netz infiziert war. Der Schadcode konnte sich im Anschluss im Netzwerk verteilen und mehr Systeme infizieren.

### 3.3.3 Technische Details

Im Weiteren ist die Funktionsweise des Exploits für Windows 7 beschrieben. Es werden dabei zwei Parteien betrachtet. Der Angreifer und das angegriffene Windows 7 System auf dem der zum SMB-Protokoll zugehörige TCP-Port geöffnet ist. Da der Exploit

[8] https://www.rapid7.com/db/modules/exploit/windows/smb/ms17_010_eternalblue

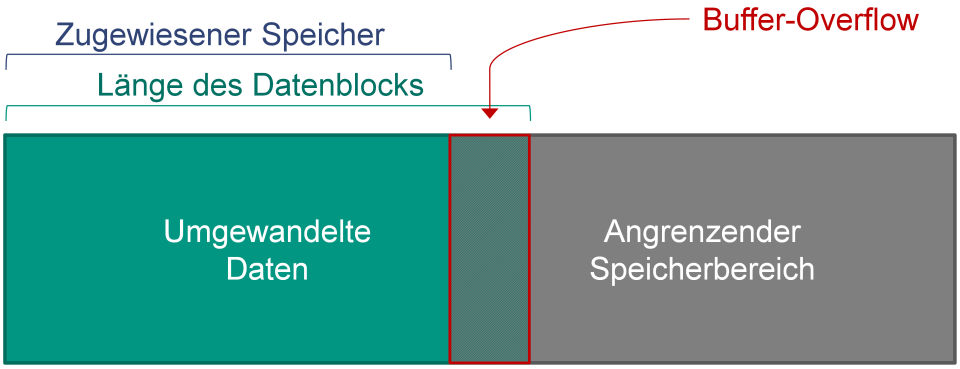


Abbildung 5: Darstellung eines Buffer-Overflows.

aus mehreren Komponenten besteht, werden zuerst die einzelnen Bestandteile für sich behandelt. Abschließend wird erklärt wie sich die Schwachstelle daraus zusammensetzt. Die Funktionsweise ist dem Forschungsbericht von Check Point Research [24] entnommen.

**Buffer-Overflow** Bis Windows 7 erlaubt jedes Windows System eine anonyme Authentifizierung über SMB1. Dies ist standardmäßig erlaubt, um jedem anderen Nutzer im lokalen Netz alle freigegebenen Ordner und Dateien auflisten zu können. Der authentifizierte Nutzer kann über die somit erstellte Session aber nicht direkt auf das System zugreifen. Es ist ihm aber erlaubt Datenpakete zu senden. Der Angreifer nutzt diese Möglichkeit aus, um Daten in einer nicht Windows NT kompatiblen Form zu senden. Dies ist wegen einer in das SMB-Protokoll eingebaute Umwandlung der Daten in eine kompatible Form möglich. Nach dem Empfangen der Daten werden diese, in umgewandelter Form, im Hauptspeicher abgelegt. Problem hierbei ist ein Programmierfehler innerhalb der Umwandlungsfunktion. Bei der Berechnung des benötigten Speicherplatzes für die umgewandelten Daten wird ein vom Sender kontrolliertes DWORD (4-Bytes) nur als WORD (2-Bytes) interpretiert. Dadurch werden die beiden höchstwertigen Bytes nicht beachtet, was zu einem zu kleinen Buffer für die zu schreibende Datenmenge führt. Deshalb entsteht ein Buffer-Overflow und Daten werden im angrenzenden Speicherblock überschrieben. Dies ist in Abbildung 5 dargestellt.

**Pool-Grooming** Zusätzlich zu der authentifizierten SMB1-Verbindung benötigt der Exploit noch mehrere SMB2-Verbindungen. Hierbei ist keine Authentifizierung nötig, da nur eine grundlegende Verbindung benötigt wird. Grund hierfür ist, dass für jede aufgebaute Verbindung ein zugehöriger Header-Block im Hauptspeicher abgelegt wird. Diese Header-Blöcke sind für den Exploit aus zwei Gründen notwendig:

- Erstens füllen diese Blöcke kleinere Lücken im fragmentierten Hauptspeicher auf, was zu einem zusammenhängenden Speicher führt. Es bleiben somit nur wenige größere Lücken übrig. Dies ist wichtig, da man erreichen will, dass der zuerst erwähnte Buffer-Overflow gezielt in einen von uns eingesetzten Block schreibt. Diese Technik nennt man Pool-Grooming.

- Zweitens enthält ein solcher Block zwei wichtige Adressen. Die eine Adresse referenziert eine Speicheradresse, an welcher alle Pakete, die über die zugehörige Verbindung empfangen werden, abgelegt werden sollen. Die andere referenziert eine Funktion, die nach Abbruch der zugehörigen Verbindung ausgeführt werden soll. Diese Funktion soll normalerweise den über die Dauer der Verbindung zugewiesenen Speicher aufräumen.

**Hole-Connection** Der letzte Bestandteil ist ein weiterer Programmierfehler innerhalb des Authentifizierungsprozesses in SMB1. Ein Angreifer kann eine zur Authentifizierung notwendige Anfrage so beantworten, dass im Hauptspeicher des Angegriffenen ein Speicherblock reserviert wird, über dessen exakte Größe der Angreifer Kontrolle hat. Durch Abbruch der Verbindung kann im Anschluss eine Lücke im Hauptspeicher erzeugt werden. Diese Verbindung wird deswegen als Hole-Connection bezeichnet.

**Exploit** Um aus diesen Komponenten einen zuverlässigen Exploit zu erstellen, werden spezielle Kenntnisse über undokumentierte Kernel-Strukturen und Implementierungsdetails innerhalb des SMB-Protokolls vorausgesetzt. Als erstes werden mehrere SMB2-Verbindungen aufgebaut. Dies führt dazu, dass viele kleinere Lücke im Hauptspeicher des angegriffenen Systems aufgefüllt werden und nur noch wenige größere Lücken übrig bleiben. In eine dieser Lücken wird anschließend ein Speicherblock durch die Hole-Connection abgelegt. Als nächster Schritt wird erneut Pool-Grooming ausgeführt. Dabei besteht eine hohe Wahrscheinlichkeit, dass einer der erzeugten Headerblöcke direkt hinter dem Speicherblock der Hole-Connection abgelegt wird. Nun wird die Hole-Connection abgebrochen. Dies führt, bei richtiger Anordnung, zu einer Speicherlücke mit festgelegter Größe vor einem Headerblock. Jetzt können, über eine anonym authentifizierte SMB1-Verbindung, Daten in einer nicht unterstützten Form, gesendet werden. Hierbei ist es wahrscheinlich, dass der umgewandelte Datenblock in die davor erzeugte Lücke abgelegt wird. Die Daten sind dabei so präpariert, dass der folgende Buffer-Overflow gezielt die zwei wichtigen Adressen im angrenzenden Header überschreibt. Überschreibt der Buffer-Overflow einen vom Angreifer nicht vorgesehenen Speicherblock, kann es zu nicht vorhersehbaren Verhalten, wie ein Bluescreen, führen. Die beiden Adressen im Header werden mit einer Adresse aus einem statischen Bereich, der Ausführung erlaubt, überschrieben. Anschließend wird über alle SMB2-Verbindungen Shellcode gesendet. Bei der Verbindung, dessen Header manipuliert wurde, landet der Shellcode direkt in einem ausführbaren Bereich. Der Abbruch aller SMB2-Verbindungen führt dadurch direkt zu einer Remote-Code-Execution. Damit kann beispielsweise eine Backdoor eingebaut werden. Diese Funktionsweise ist in Abbildung 6 dargestellt


Schritt 0: Ausgangszustand des Hauptspeichers.


Schritt 1: Pool-Grooming führt zu Headerblöcken im Hauptspeicher.


Schritt 2: Die Hole-Connection belegt einen Bereich im Hauptspeicher.


Schritt 3: Zweites Pool-Grooming erstellt zusätzliche Headerblöcke.


Schritt 4: Abbruch der Hole-Connection führt zu freiem Speicherplatz.


Schritt 5: Ein Umgewandeltes Datenpaket wird im erzeugten freien Speicher abgelegt. Es folgt ein Buffer-Overflow in den anliegenden Header.

Abbildung 6: Schrittweise Darstellung des EternalBlue-Exploits.

### 3.3.4 Schutzmöglichkeiten

Genau einen Monat vor der Veröffentlichung des Fuzzbunch Frameworks erschien der `MS17-010`-Patch von Microsoft [25]. Dieser verhindert auf allen gängigen Windows Versionen den EternalBlue-Exploit. Aufgrund der schwere der Sicherheitslücke, veröffentlichte Microsoft den Patch auch für nicht mehr unterstütze Windows Versionen wie Windows XP [26]. Der Patch beseitigt dabei nicht den zugrunde liegenden Programmierfehler, sondern fügt zusätzliche Authorisierungsabfragen beim Übermitteln von Daten ein. Es ist somit über das Netzwerk ermittelbar, ob auf einem System dieser Patch installiert ist. Hierfür gibt es Scanner im Metasploit-Framework[9], in Python[10] und für NMAP[11]. Eine weitere Möglichkeit, um sich vor dem Exploit zu schützen, ist das Deaktivieren von SMB1. Dies verhindert alle für den Exploit nötigen Schritte über dieses Protokoll. Microsoft hat bereits selbst die Gefahren, die durch das alte Protokoll ausgehen, erkannt und es mit dem Windows 10 Fall Creators Update standardmäßig auf Windows 10 deaktiviert [7].

## 3.4 SMBLoris

### 3.4.1 Einführung

Dieser Angriff wurde während der Analyse von EternalBlue entdeckt. Dabei handelt es sich um einen DOS-Angriff, d.h. im Gegensatz zu den vorherigen Angriffen gibt es hier keine Codeausführung. Der SMBLoris Angriff verhält sich ähnlich zum SlowLoris Angriff [27], weshalb er auch dessen Name geerbt hat. Beim SlowLoris öffnet eine einzelne Maschine mehrere HTTP-Verbindungen zu einem Server und hält diese so lang wie möglich offen. Andere Verbindungen können während dieser Zeit nicht verarbeitet werden. Die Besonderheit dabei ist, dass der Server nicht abstürzt, lediglich der HTTP-Server ist von dem Angriff betroffen. Von SMBLoris sind alle Windows Versionen betroffen. Auch hier ermöglicht ein Designfehler, im SMB-Protokoll, den Angriff. Dieser Fehler bezieht sich auf die Speicherallokierung beim Verbindungsaufbau einer SMB-Verbindung.

### 3.4.2 Angriffsszenario

Obwohl SMBLoris keine Code-Ausführung erlaubt, gibt es dennoch einige Szenarien in denen der Angriff Schaden anrichten kann. Da die Speicherallokierung vor dem Authentifizierungsprozess stattfindet, kann ein Angreifer beliebig viele Verbindungen zum Opfer aufmachen. In Abbildung 7 ist dies veranschaulicht. Dadurch, dass der Angreifer die Verbindung halboffen lässt, wird das Opfer so sehr ausgelastet, dass es aufhört zu funktionieren. Der Angriff eignet sich beispielsweise dazu andere zu erpressen und diese so lange anzugreifen, bis diese der Forderung nachkommen. Außerdem kann der Angriff auch genutzt werden, um für Ablenkung zu sorgen. Während alle sich um den SMBLoris Angriff kümmern, können andere Angriffe ausgeführt werden, mit der Hoffnung unentdeckt zu bleiben.

[9] https://www.rapid7.com/db/modules/auxiliary/scanner/smb/smb_ms17_010
[10] https://github.com/nixawk/labs/blob/master/MS17_010/smb_exploit.py
[11] https://nmap.org/nsedoc/scripts/smb-vuln-ms17-010.html

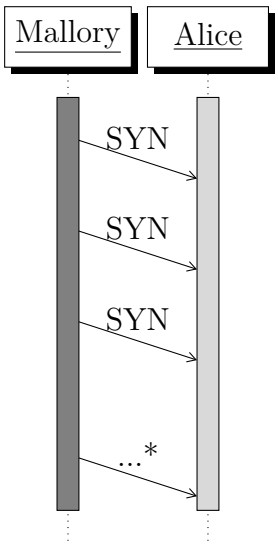


Abbildung 7: Darstellung von mehreren halboffenen SMB-Verbindungen.
* insgesamt 65,535 Verbindungen

### 3.4.3 Technische Details

SMBLoris nutzt einen Designfehler im SMB-Protokoll aus, der in der Speicherallokierung liegt. Der Angriff nutzt dabei den SMB-Port 445. Dabei wird beim Starten einer SMB Session zunächst der NetBIOS-Session-Service (NBSS) Header über TCP übermittelt. Dies geschieht noch bevor irgendeine Authentifikation stattgefunden hat. Der Header ist in Abbildung 8 zu sehen. Dieser benötigt 4 Bytes im Speicher des Angreifers. Das Opfer hingegen allokiert beim Empfang die angeforderte Länge von 128 KiB ($2^{17}$ = 131.072-Bytes = 128 KiB). Dieser Speicher wird in einem non-paged-pool angelegt, d.h. die Daten können nicht ausgelagert werden. Das führt dazu, dass der Speicher und die CPU die gesamte Verbindung über in Anspruch genommen werden und schließlich das System zum Absturz bringen. Der Angriff nutzt dabei die Eigenschaft, dass das Öffnen einer Verbindung günstiger ist, als diese zu halten. Tabelle 1 zeigt die Auswirkungen auf den Speicher während des Angriffs. Dabei werden die Anzahl an aufgebauten Verbindungen skaliert. Die Tabelle zeigt zunächst, dass mit einer IP 65,535 Verbindungen aufgebaut werden können. Die Anzahl der Verbindungen ist dabei durch den TCP-Port limitiert. Dieser erlaubt 65,535 ($2^{16}$) Verbindungen. Bei dieser Anzahl an Verbindungen verbraucht der Angreifer lediglich 256 KiB an Speicher. Beim Opfer werden dagegen 8 GiB Speicher fällig. Bereits mit nur zehn IPs kann ein Angreifer mit nur 2,5 MiB Speicherverbrauch 80 GiB Speicher beim Opfer allokieren und ein Windows System somit in die Knie zwingen.

### 3.4.4 Schutzmöglichkeiten

Obwohl alle Windows Systeme angreifbar sind plant Microsoft keinen Patch, da man grundsätzlich den ganzen Speicherallokierungsprozess ändern müsste, der seit über 20 Jahren in Betrieb ist [29]. Außerdem sieht Microsoft in diesem Angriff keinen Sicherheits-

```
struct NBSS_HEADER
{
    char   MessageType   : 8;
    char   Flags         : 7;
    int    Length        : 17;
};
```

Abbildung 8: Der NBSS Header [28]

| Szenario | Verbindungen | Speicher (Angreifer) | Speicher (Opfer) |
|---|---|---|---|
| Ausgangswert | 1 | 4 bytes | 128 KiB |
| Eine IPv4 | 65,535 | 256 KiB | 8 GiB |
| Eine IPv6 | 65,535 | 256 KiB | 8 GiB |
| Zwei IPv4/IPv6 | 131,070 | 512 KiB | 16 GiB |
| 10 IPs | 655,535 | 2,5 MiB | 80 GiB |

Tabelle 1: Speicher Allokationen während des Angriffs [28]

fehler [30], auch wenn es durchaus Angriffszenarien gibt. Auch Linux Systeme sind in ihrer Standardkonfiguration verwundbar. Um sich vor dem Angriff zu schützen, setzt man die maximale Anzahl der SMB-Verbindungen herunter, zum Beispiel auf 1000. Diese Konfiguration befindet sich i.d.R. in `/etc/samba/smb.conf`. Indem `max smbd processes = 1000` gesetzt wird, schützt man sich gegen den SMBLoris Angriff.

# 4 Evaluierung

Dieser Abschnitt soll zeigen, ob und wie relevant die in Abschnitt 3 beschriebenen Angriffe heute noch sind. Dazu wurde ein Angreifersystem mit Kali Linux (Version 2017.3) aufgesetzt. Das Linux Image hierfür stammt von Offensive Security[12], einer Penetration-Testing-Website und ist bereits mit einigen Angriffswerkzeugen ausgestattet. Eins davon ist das Metasploit-Framework, mit dem die meisten unsere Angriffe durchgeführt wurden. Insgesamt gab es vier verschiedene Zielsysteme. Ein Windows 7 Professional mit Service Pack 1 (SP1), ein Windows 7 Professional mit SP1 und neuesten Updates, ein aktuelles Windows 10 mit ebenfalls allen Updates und ein Beckhoff Embedded-PC mit Windows 7 Embedded. Alle Systeme wurden mit den in Abschnitt 3 beschriebenen Angriffen attackiert. Für die Angriffe Pass-The-Hash in Kombination mit PsExec, SMBRelay und EternalBlue wurde das jeweilige Metasploit-Modul verwendet. Für Pass-The-Hash wurde dazu das standardmäßig enthaltene `RID 500` Administratorenkonto aktiviert und dessen, mittels eines Hashdump[13] ausgelesener, Passwort-Hash verwendet. Bei SMB-Loris wurde auf eine C-Implementierung[14] zurückgegriffen. Die Ergebnisse sind in Tabelle 2

[12]https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-hyperv-image-download/
[13]https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/
[14]https://gist.github.com/marcan/6a2d14b0e3eaa5de1795a763fb58641e

| Betriebssystem | PSExec | SMBRelay | EternalBlue | SMBLoris |
|---|---|---|---|---|
| Windows 7 Professional SP1 | X | X | X | X |
| Windows 7 Professional SP1 * | X | | | X |
| Windows 10 ** | X | | | X |
| Beckhoff Embedded-PC | - | - | | X |

Tabelle 2: Übersicht darüber, welches System für welchen Angriff anfällig ist.
* Inklusive Update KB4054518[15]
** Inklusive Update KB4053580[16]

zusammengefasst. Dort ist zu sehen, dass die Windows 7 Maschine ohne aktuelle Updates und trotz SP1 für alle Angriffe anfällig war. Anders sah es bei dem aktualisierten Windows 7 mit SP1 aus. Durch die Updates wurde der EternalBlue-Exploit verhindert. Außerdem wurde NTLM durch eine neuere Version ersetzt, wodurch SMBRelay nicht mehr funktionierte. Die Windows 10 Maschine war, ohne weitere Konfigurationen, gegen beiden Angriffe immun. Pass-The-Hash mittels PsExec funktionierte auf beiden neueren Versionen weiterhin. Beim Embedded-System konnten nur EternalBlue und SMBLoris untersucht werden. Für Pass-The-Hash fehlte der Passwort-Hash und für SMBRelay die Möglichkeit eine Authentifikation abzufangen und weiterzuleiten. Das System war ohne Veränderung von uns unempfindlich. SMBLoris war der einzige Angriff, der auf jeder Maschine erfolgreich war. Dieser funktionierte sowohl auf dem alten Windows 7, dem aktuellsten Windows 7, Windows 10 und Windows 7 Embedded. Die Gründe dafür wurden bereits in Unterabschnitt 3.4 erläutert.

Vor allem bei den Windows 7 Versionen wird sichtbar, dass ein aktuelles System, mit richtiger Konfiguration, vor Angriffen schützt. Für die Angriffe aus dem Metasploit-Frameworks ist nicht auszuschließen, dass diese mit anderen Implementierungen, auch auf neueren Systemen funktionieren.

# 5 Zusammenfassung und Ausblick

In dieser Arbeit haben wir zunächst das SMB-Protokoll vorgestellt. Ein Netzwerkprotokoll, das verwendet wird, um Zugriff auf Freigaben in einem Netzwerk zu erhalten. Für dieses Protokoll haben wir vier Angriffe vorgestellt, die Design- oder Implementierungsfehler der zugehörigen Microsoft Implementierung ausnutzen. Schließlich haben wir die jeweiligen Angriffe auf Testsystemen ausgeführt, um zu prüfen, wie relevant diese heute noch sind. Das Fazit war, dass nicht aktualisierte oder falsch konfigurierte Systeme immer noch anfällig sein können. Deshalb ist es wichtig sein Netzwerk richtig abzusichern. Mögliche Sicherheitsmaßnahmen wurden bei den jeweiligen Angriffen besprochen.

Obwohl die Evaluierung gezeigt hat, dass richtig konfigurierte oder aktualisierte Systeme nicht mehr für die jeweiligen Exploits anfällig sind, schließt das andere Implementierungen der Angriffe nicht aus. Neben den verwendeten Metasploit-Modulen gibt es noch weitere

Implementierungen der Angriffe, wie zum Beispiel `SMBRELAYX`[17]. Dabei handelt es sich um eine SMBRelay Implementierung in Python, die auch mit der aktuellen NTLM Version funktioniert. Zukünftige Arbeiten könnten weitere Implementierungen der Angriffe untersuchen und prüfen, ob auch neuere Windows Versionen angreifbar sind. Außerdem wurde in dieser Arbeit nur das Microsoft-SMB-Protokoll untersucht. Weitere Arbeiten könnten die freie multiplattform Implementierung SAMBA fokussieren und untersuchen, ob und unter welchen Umständen SAMBA angreifbar ist.

[17] https://github.com/CoreSecurity/impacket/blob/master/examples/smbrelayx.py

# Abbildungsverzeichnis


1 Ausbreitung von WannaCry [2]. . . . . 2
2 Darstellung einer SMB-Verbindung mit Öffnen einer Freigabe, basierend auf [9] . . . . . 4
3 Darstellung eines erfolgreichen NTLM-Authentifizierungsprozesses. . . . . 7
4 Darstellung eines erfolgreichen SMBRelay Angriffs. . . . . 11
5 Darstellung eines Buffer-Overflows. . . . . 14
6 Schrittweise Darstellung des EternalBlue-Exploits. . . . . 16
7 Darstellung von mehreren halboffenen SMB-Verbindungen. . . . . 18
8 Der NBSS Header [28] . . . . . 19

# Literatur


[1] Kim Zetter. *The Sony Hackers Were Causing Mayhem Years Before They Hit the Company*. 16. Feb. 2016. URL: https://www.wired.com/2016/02/sony-hackers-causing-mayhem-years-hit-company/.

[2] rain-1. *Wannacrypt0r Factsheet*. 23. Mai 2017. URL: https://gist.github.com/rain-1/989428fa5504f378b993ee6efbc0b168.

[3] *Samba: An Introduction*. Samba Team. 27. Nov. 2001. URL: https://www.samba.org/samba/docs/SambaIntro.html.

[4] *Microsoft SMB Protocol and CIFS Protocol Overview*. Microsoft. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365233(v=vs.85).aspx.

[5] *SMB2, a complete redesign of the main remote file protocol for Windows*. Microsoft. 9. Dez. 2008. URL: https://blogs.technet.microsoft.com/josebda/2008/12/09/smb2-a-complete-redesign-of-the-main-remote-file-protocol-for-windows/.

[6] *Stop using SMB1*. Microsoft. 16. Sep. 2016. URL: https://blogs.technet.microsoft.com/filecab/2016/09/16/stop-using-smb1/.

[7] *SMBv1 is not installed by default in Windows 10 Fall Creators Update and Windows Server, version 1709*. Microsoft. 5. Feb. 2018. URL: https://support.microsoft.com/en-us/help/4034314/smbv1-is-not-installed-windows-10-and-windows-server-version-1709.

[8] *Microsoft SMB Protocol Dialects*. Microsoft. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365235(v=vs.85).aspx.

[9] *Microsoft SMB Protocol Packet Exchange Scenario*. Microsoft. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365236(v=vs.85).aspx.

[10] *Microsoft SMB Protocol Authentication*. Microsoft. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365234(v=vs.85).aspx.

[11] *Microsoft NTLM*. Microsoft. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/aa378749(v=vs.85).aspx.

[12] *Was ist eine Man-in-the-Middle-Attacke?* Kaspersky. 10. Apr. 2013. URL: https://www.kaspersky.de/blog/was-ist-eine-man-in-the-middle-attacke/905/.

[13] *Denial of Service Attack: What it is and how to prevent it*. The Windows Club. 31. März 2017. URL: http://www.thewindowsclub.com/dos-denial-of-service-attack.

[14] Paul Ashton. *NT Pass-the-Hash with Modified SMB Client Vulnerability*. 1997. URL: https://www.securityfocus.com/bid/233/discuss.

[15] Dagmar Heidecker. *Pass-the-Hash Angriffe (PtH) – kurz zusammengefasst*. 16. Sep. 2015. URL: https://blogs.technet.microsoft.com/austria/2015/09/16/pass-the-hash-angriffe-pth-kurz-zusammengefasst/.

[16] Daniel Stirnimann. *Windows Attack — Gain Enterprise Admin Privileges in 5 Minutes*. 9. Aug. 2010. URL: https://www.hacking-lab.com/misc/downloads/event_2010/daniel_stirnimann_pass_the_hash_attack.pdf.

[17] Jürgen Schmidt. *Windows 10 mit Schutz vor Pass-the-Hash-Angriffen*. 28. Juli 2016. URL: https://www.heise.de/security/artikel/Windows-10-mit-Schutz-vor-Pass-the-Hash-Angriffen-3280610.html.

[18] harmj0y. *Pass-the-Hash Is Dead: Long Live LocalAccountTokenFilterPolicy*. 16. März 2017. URL: http://www.harmj0y.net/blog/redteaming/pass-the-hash-is-dead-long-live-localaccounttokenfilterpolicy/.

[19] Microsoft. *Using SMB Packet Signing*. URL: https://technet.microsoft.com/en-us/library/cc180803.aspx.

[20] Rapid7. *Microsoft CVE-2017-0144. "Windows SMB Remote Code Execution Vulnerability"*. URL: https://www.rapid7.com/db/vulnerabilities/msft-cve-2017-0144.

[21] Shadow Brokers. *Lost in Translation*. 14. Apr. 2017. URL: https://steemit.com/shadowbrokers/@theshadowbrokers/lost-in-translation.

[22] Craig Timberg Ellen Nakashima. „NSA officials worried about the day its potent hacking tool would get loose. Then it did.“ In: *The Washington Post* (16. Mai 2017). URL: https://www.washingtonpost.com/business/technology/nsa-officials-worried-about-the-day-its-potent-hacking-tool-would-get-loose-then-it-did/2017/05/16/50670b16-3978-11e7-a058-ddbb23c75d82_story.html.

[23] Dylan Davis Sean Dillon. *ETERNALBLUE. Exploit Analysis and Port to Microsoft Windows 10*. Techn. Ber. Version 1.2. RiskSense, 5. Juli 2017.

[24] Nadav Grossman. *EternalBlue – Everything There Is To Know*. 29. Sep. 2017. URL: https://research.checkpoint.com/eternalblue-everything-know/.

[25] *Microsoft Security Bulletin MS17-010 - Critical*. Microsoft. 14. März 2017. URL: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/ms17-010.

[26] Michael Kranawetter. *WannaCrypt: Microsoft schützt auch ältere Windows-Versionen*. Microsoft. 13. Mai 2017. URL: https://blogs.technet.microsoft.com/michaelkranawetter/2017/05/13/wannacrypt-microsoft-schutzt-auch-altere-windows-versionen/.

[27] Zach Banks. *Slowloris HTTP denial of service*. URL: https://hackaday.com/2009/06/17/slowloris-http-denial-of-service/.

[28] JennaMagius zerosum0x0. *SMBLoris - Windows Denial of Service Vulnerability*. URL: https://smbloris.com/.

[29] Michael Mimoso. *Windows SMB Zero Day to Be Disclosed During DEF CON*. URL: https://threatpost.com/windows-smb-zero-day-to-be-disclosed-during-def-con/126927/.

[30] Eslam Medhat. *Microsoft refused to fix SMB vulnerability.* URL: https://latesthackingnews.com/2017/08/05/microsoft-refused-fix-smb-vulnerability/.