Badlock

Lücke in Windows und Samba mit großer Tragweite

Die Samba-Entwickler haben vor vielen Monaten eine Sicherheitslücke entdeckt, die sich als Protokollschwäche erwies und Samba wie Windows gleichermaßen betrifft. Patches, die Samba- und Windows-Server-Betreiber unbedingt einspielen sollten, sind seit dem 12. April verfügbar – ein Blick auf die technischen Hintergründe und die Tragweite des Problems.

VON PETER SIERING

Der Entdecker der Badlock getauften Lücke, Stefan Metzmacher, entwickelt als Mitglied des Entwicklungs-Teams an der freien Windows-Server-Alternative Samba. Sein Arbeitgeber, die SerNet GmbH in Göttingen, bietet unter anderem Beratung, Anpassungen und Fehlersuche rund um Samba an. So entschied sich das Unternehmen, publikumswirksam drei Wochen vor Erscheinen der Patches auf deren Wichtigkeit aufmerksam zu machen. Man richtete eigens eine Internet-Seite ein und gestaltete ein Logo, verriet aber keine Details. Kurz vor Redaktionsschluss unterhielten wir uns mit Volker Lendecke, der SerNet mitgegründet hat und erstes deutsches Mitglied des Samba-Teams war, über die Hintergründe von Badlock.

Stress macht Stress

Er berichtete, dass Stefan Metzmacher den erschreckenden Erkenntnissen eines Stresstests nachging, den die Samba-Entwickler von einem spezialisierten Unternehmen an Teilen ihres Codes hatten durchführen lassen. Dieser Test betraf die Samba-eigene Implementierung der RPC-Dienste. Remote Procedure Calls (RPCs) dienen dazu, übers Netz gezielt Funktionen auf anderen Computern aufzurufen. In der Windows-Welt sind sie als MSRPC bekannt und wurden von der DCE/RPC-Implementierung abgeleitet. MSRPCs sind unter anderem gefragt, wenn Windows übers Netz gesteuert wird, etwa beim Bearbeiten der Registry, Ausführen der Datenträgerverwaltung oder Verwalten von Benutzerkonten. Sie spielen bis hin zu den aktuellen Windows-Versionen eine tragende Rolle und haben wie das Dateizugriffsprotokoll SMB eine bewegte Geschichte hinter sich.

Die Samba-Implementierung der MSRPC-Dienste wies diverse Mängel auf, wenn sie mit gezielt konstruiertem Datenmüll beschossen wurde (Fuzzing). Die Mängel zeigten sich nicht nur durch Abstürze oder hohe CPU-Last. Bei näherer Untersuchung stellte sich heraus, dass dabei aus sicheren Verbindungen mit aktiver Verschlüsselung und Integritätsprüfung plötzlich unsichere entstanden. Genau diesem Umstand forschte Stefan Metzmacher hinterher. Er fand heraus, dass aktuelle Windows-Server das gleiche Verhalten an den Tag legen: Sie lassen sich auf RPC-Aufrufe ein, die sensitive Daten manipulieren und ausliefern, selbst wenn die Verbindung weder verschlüsselt noch integritätsgeschützt ist. Die Lücke betrifft die Schnittstellen für Benutzeranmeldungen, Local Security Authority (LSA), und die Benutzerverwaltung, Security Account Manager (SAM).

Im regulären Betrieb kommen solche unsicheren Verbindungen nicht vor. Clients und Server geben sich Mühe, die Daten nur gesichert auszutauschen. Samba- und Windows-Server lassen sich aber durchaus dazu verführen. Nach aktuellem Kenntnisstand genügt ein Windows-Client-System, das Mitglied einer Active-Directory-Domä-

ne ist. Während sich ein Administrator an einem solchen System anmeldet, tätigt es einige RPC-Standardaufrufe gegen einen Domänen-Controller. Fängt ein Man-in-the-Middle die Aufrufe ab, kann er harmlose Namensanfragen ummodeln, indem er das Aufrufziel auf LSA oder SAM(R) abändert und den Verschlüsselungswunsch tilgt, den der Client beim Verbindungsaufbau mitgeschickt hatte. Ähnliche Übergriffe sind auch denkbar, wenn ein Administrator einen neuen Domänen-Controller in ein Active Directory hängt.

Für den Angriff ist wichtig, wie RPC-Aufrufe einen Server erreichen. Sie können verpackt in einer authentifizierten SMB-Verbindung ankommen, die dank SMB-eigener Integritätsprüfung und Verschlüsselung manipulationssicher ist. RPC-Aufrufe können aber auch direkt ohne SMB-Zwischenschicht via TCP-Verbindung eingehen. Die dabei transportierten Daten sind anfangs nicht verschlüsselt und nicht integritätsgeschützt. Bei RPC via TCP müssen Client und Server obendrein die Authentifizierung selbst erledigen. Die ersten RPC-Aufrufe, die Windows-Clients bei der Nutzeranmeldung an eine Domäne absetzen, laufen über TCP. Der Client erfragt zunächst via RPC beim End-Point-Mapper (Pförtner), auf welchem TCP-Port er das eigentliche Aufrufziel erreichen kann (DRSUAPI beim Anmelden). Auf diesen Port setzt er dann die RPC-Funktionsaufrufe ab und überträgt dabei Authentifizierungsdaten.

Genau hier könnte ein Man-in-the-Middle angreifen: Er ändert das Ziel von DRSUAPI auf SAMR und startet mit den enthaltenen Daten zur Authentifizierung eine Verbindung ohne Verschlüsselung zum Server. In der kann er walten und schalten, wie es ihm beliebt. Mit SAMR wäre es möglich, die Passwörter aller Benutzer im Active Directory zu ändern oder eigene Administrator-Accounts in der Domäne zu erstellen. Man mag sich nicht ausmalen, wie viel Bitcoin ein Großunternehmen berappen würde, um die Passwörter für alle Benutzerkonten zurückzubekommen, wenn die durch Zufallsdaten ersetzt wurden. Man-in-the-Middle kann ein übernommener Netzwerkdrucker, aber auch ein Software-Schläfer auf einem PC im Active Directory sein, auf dem sich zu Wartungszwecken ein Admin anmeldet.

Um die Lücke ausnutzen zu können, muss ein Angreifer ins lokale Netz eingedrungen sein und auf einem lokalen System administrative Rechte erlangt haben. Davon ausgehend könnte er unbeschränkter Herrscher im Windows-Netz werden, weil er sich zum Domänen-Administrator machen kann. Die Ursache ist der ungeschützte RPC-Verbindungsaufbau per TCP. Die dabei gesendeten Daten sind

nicht integritätsgeschützt und die enthaltenen Authentifizierungs-Tokens nicht mit der Verbindung assoziiert. Ein Man-in-the-Middle kann eingreifen, ohne dass die Anfrage ungültig wird.

Wo die Zeit blieb

Die Lücke steckt sowohl in Samba als auch in Windows Server. Zum Microsoft-Patchday am 12. April sind Patches für beide Umgebungen zu haben. Admins sollten sie bereits eingespielt haben. Das Gegenmittel ist kein Hexenwerk: Indem die SAMR- und LSA-RPC-Dienste keine TCP-Verbindung mehr annehmen, wenn sie unverschlüsselt antworten sollen, ist die Lücke geschlossen. Bei Samba führen die strikteren Einstellungen womöglich dazu, dass Admins über neue smb.conf-Optionen nachjustieren müssen, wenn ältere Client-Software Ärger machen sollte.

Bleibt die Antwort auf die Frage, warum ein so trivialer Fix so lange auf sich warten lässt: Die Samba-Entwickler haben in ihrer RPC-Implementierung gründlich aufgeräumt und alle beim Stresstest gefundenen Probleme beseitigt. Für die aktuelle Fassung 4.3 kommen 200 Patches zusammen, die nicht in ein paar Tagen geschrieben und getestet sind. Microsoft wusste seit September Bescheid und hat laut Lendecke viel Zeit gebraucht, um in Kompatibilitätstests sicher zu gehen, dass der Fix keine legitimen Anwendungen torpediert.

Während eingeweihten Entwicklern bei Microsoft und im Samba-Team die Probleme seit Monaten bekannt waren, hat die medial angespitzte Ankündigung drei Wochen vor Veröffentlichung der Patches der Göttinger SerNet einige Kritik eingebracht. Darunter waren diverse Vorwürfe zu hören: Da hätte jemand erst Sicherheitslücken eingebaut, um sie dann Jahre später zu entdecken. Die Ankündigung der Lücke würde böse Buben inspirieren, danach zu suchen und Schädlinge zu bauen, und so weiter. Der Name Badlock ist synthetisch, die Domain war frei und in den Suchmaschinen nicht vorbelegt. So hat SerNet all diejenigen ordentlich verladen, die im Locking-Code von Samba, der von Stefan Metzmacher stammt, die Lücke gesucht haben.

> **„Da schlummert noch einiges. Ich gehe davon aus, dass die Leute links und rechts gucken, was man mit RPC noch machen kann."**
>
> **Samba-Entwickler Volker Lendecke über die RPC-Dienste in Windows, auf denen auch DCOM, WMI & Co. aufbauen**

Das Samba-Team bot Samba-OEMs auch schon vor offiziellem Erscheinen der Patches und Erklärungen der Lücke die Möglichkeit, ihre Produkte entsprechend zu wappnen. Die zugehörigen Fehler im Bugtracker sind ihnen zugänglich, die Patches auf Anforderung per PGP-Key ebenfalls. Ob das allerdings genügt, alle aktiven Systeme rechtzeitig zu versorgen, darf bezweifelt werden: Patches gibt es nur für Samba 4.2 und neuer. Samba 4.1 erhält keine Bug-Fixes mehr, ebensowenig Version 3.6 – beide sind aus dem aktiven Support raus. Wie es um noch ältere Versionen bestellt ist, die eine andere Implementierung für MSRPC verwenden, hat niemand im Detail untersucht (die Entwickler haben MSRPC aus Samba 4 auf 3.6 rückportiert). Für NAS-Geräte mit alten Samba-Versionen gibt es also wenig Hoffnung.

Bei Samba besteht im Unterschied zu Windows zusätzlich das Problem, dass SAMR und LSA ständig erreichbar sind – selbst wenn Samba nicht als Active-Directory-Domänen-Controller läuft, sondern nur als Standard-Server. Inwieweit die Lücke auch dann ein Problem darstellt, hat bisher niemand untersucht. Am 12. April sind acht CVEs allein zu Samba erschienen (siehe c't-Link). Die Patches behandeln all diese Probleme. Bei Microsoft kann man das leider nicht im Detail nachlesen.

Apropos Microsoft: Ob und welche RPC-Dienste in Windows in ähnlicher Weise angreifbar sind, bleibt ebenfalls offen. An Badlock zeigt sich, dass die bei RPCs verwendeten Protokolle den heutigen Anforderungen an sichere Systeme kaum gewachsen sind. Aufgrund der Kompatibilität zu alten Anwendungen kann Microsoft hier auch keine großen Sprünge machen und etwa einen Integritäts-Check nachträglich einbauen. Stefan Metzmacher hat Ansätze erforscht, einen solchen Check abwärtskompatibel nachzurüsten, und möchte den Ansatz demnächst mit Microsoft diskutieren und in Samba einbauen. *(ps@ct.de)*

Patches und CVEs: *ct.de/yf35*

So könnte ein Badlock-Angriff ablaufen

An einem Windows-Client meldet sich ein Administrator an: Das System befragt via RPC den End Point Mapper nach dem Port für DRSUAPI (1). Mit einer weiteren TCP-Verbindung will es dorthin verbinden (2). Der Man-in-the-Middle lenkt auf SAMR um, schwätzt dem Server eine ungeschützte Verbindung auf und übernimmt so die ganze Kommunikation. Der Client wird keine Fehlermeldung bringen, sondern erneut versuchen, den Server zu kontaktieren.