Angriffstechnik im lokalen Netz: ARP-Spoofing und -Poisoning

Interner Zugriff

Im LAN kann jeder Rechner den Netzverkehr belauschen und manipulieren. An dieser Tatsache ändert auch ein Switch wenig: Durch ARP-Spoofing und -Poisoning lenkt der Angreifer jeden Verkehr auf seinen Rechner. Wie das funktioniert und welche Gegenmaßnahmen dem Admin bleiben, zeigt dieser Artikel. Thomas Demuth, Achim Leitner

Neugier, Mobbing, Konkurrenzkampf oder Wirtschaftsspionage – es gibt viele Gründe, warum sich Insider unberechtigt Zugang zu Daten und Systemen verschaffen. Laut Statistik kommen 70 bis 80 Prozent aller Angriffe aus dem internen Netz **[1]**. Admins haben es schwer, dies zu verhindern: Der innere Schutz ist deutlich schwieriger als die Verteidigung gegen externe Attacken.

Gefahr durch Innentäter

Den Innentätern bleibt die Wahl unter vielen Angriffstechniken. Die meisten setzen einiges Fachwissen beim Angreifer und eine spezielle Betriebssystem- und Software-Umgebung beim Angegriffenen voraus. Es gibt aber auch Verfahren, die fast immer funktionieren: Per ARP-Spoofing setzen sich Angreifer in eine Position, in der sie alle Datenströme im lokalen Netz abhören und manipulieren können. Diese als Man-in-the-Middle-Angriff bekannte Technik ist einfacher als viele IT-Verantwortliche und Admins glauben. Dank ausgeklügelter Software gelingt dies auch dem Laien in Sachen Netzwerk und Sicherheit.

Um Hintergründe und mögliche Verteidigungsstrategien zu verstehen, sind einige Grundkenntnisse in Sachen ARP (Address Resolution Protocol) nötig. Dieses alte und relativ einfache Protokoll bildet IP- auf MAC-Adressen ab (siehe **Kasten „Grundlagen: Adressen im LAN"**). Während der Anwender nur den Namen »www.linux-magazin.de« oder die IP-Adresse »62.245.157.216« kennt, wird der Rechner im LAN ausschließlich mit seiner MAC-Adresse angesprochen »00:C1:26:07:CF:F3«.

Das ARP-Protokoll wurde im November 1982 von David C. Plummer als RFC 826 veröffentlicht **[2]**. Trotz des Titels „An Ethernet Address Resolution Protocol" ist der Ansatz generisch gehalten und nicht auf Ethernet oder TCP/IP beschränkt. Da IT-Sicherheit im Jahre 1982 noch keine wesentliche Rolle spielte, war das Ziel lediglich, eine bestimmte Funktionalität bereitzustellen. Ob und inwieweit diese Funktion angreifbar ist, wurde erst später diskutiert.

ARP stellt die Zuordnung von IP- zu MAC-Adresse fest. Wenn Client C ein Paket an Server S senden will, muss er die MAC-Adresse von S kennen, wenn beide im selben Subnetz stehen. Selbst wenn S in einem fremden Netz steht, braucht C eine MAC-Adresse, dann aber die des nächsten Routers (meist das Standard-Gateway), siehe **Abbildung 1**. Der Router kümmert sich dann um alles Weitere.

So funktioniert ARP

Um die MAC-Adresse des Zielsystems zu erfahren, schickt C einen ARP-Request per Broadcast an alle Maschinen im lokalen Netz mit der Frage „Wer hat die IP-Adresse a.b.c.d?". Der Rechner mit der passenden Nummer antwortet und teilt dem Client die gesuchte MAC-Adresse mit (**Abbildung 2**).

Wie in **Abbildung 3** zu sehen ist, überträgt ein Ethernet-Frame das ARP-Paket direkt als Payload (Nutzlast), es sind keine weiteren Protokolle zwischengeschaltet. Dazu ist im Type-Feld des Frame-Headers der Wert »0x0806« eingetragen – das sagt dem Empfänger, dass der Frame ein ARP-Paket enthält. **Tabelle 1** führt auf, welche Werte in welchen Feldern des Ethernet-Frame und des ARP-Pakets zu finden sind.

Da es viel zu aufwändig wäre, vor dem Absenden jedes Pakets eine ARP-Anfrage zu stellen und die Antwort abzuwarten, verfügt jeder IP-Stack über eine ARP-Tabelle, auch ARP-Cache genannt (**Abbildung 4**). Der Cache listet IP- und zugehörige MAC-Adressen tabellarisch. Er nimmt statische (vom Anwender generierte) und dynamische (durch das

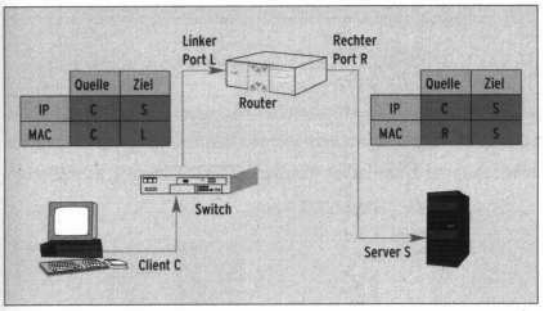

Abbildung 1: Client C sendet ein Paket an Server S. Als Ziel-IP-Adresse setzt er S ein, als MAC-Adresse das linke Router-Interface L. Der Router leitet das Paket weiter; die IP-Adressen lässt er, setzt aber die MAC-Adressen neu. Ein Switch verändert das Paket nicht.

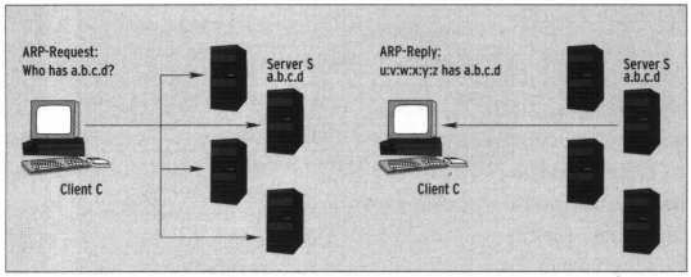

Abbildung 2: Mittels ARP stellt der Client im LAN die MAC-Adresse des Servers fest, bevor er ein Paket an ihn sendet. Die Frage „Who has ..." geht per Broadcast an alle Rechner im LAN. Der gesuchte Partner sendet seine Antwort direkt zum Fragenden.

ARP-Protokoll gelernte) Einträge auf. Dynamische Einträge sind in der Regel nur eine begrenzte Zeit gültig, oft nur wenige Minuten.

Angriffe auf die Adressierung im LAN

Da das Address Resolution Protocol keinerlei Anstrengung unternimmt, um sich vor gefälschten Paketen zu schützen, ist es anfällig für eine Reihe von Angriffen. Bekannt und verbreitet sind drei Varianten: MAC-Spoofing, MAC-Flooding (auch CAM-Flooding genannt) sowie ARP-Spoofing. Beim MAC-Spoofing nutzt der Angreifer eine falsche MAC-Absenderadresse. Das ist sinnvoll, wenn bestimmte Rechte mit einer MAC-Adresse verbunden sind. Viele WLAN-Betreiber (Wireless LAN) setzen beispielsweise auf eine Zugriffsliste, die die MAC-Adressen der berechtigten Benutzer verzeichnet.

Dieser schwache Sicherheitsmechanismus ist leicht zu umgehen, wenn der Angreifer eine der berechtigten Adressen kennt und selbst einsetzt, solange der rechtmäßige Besitzer inaktiv ist. MAC-Spoofing ist auch für Angreifer nützlich, die ihre Identität verbergen wollen.

Im drahtgebundenen Netz gibt es ein probates Mittel dagegen: Viele Switches kennen ein Feature namens Port Security. Der Switch lernt jede MAC-Adresse nur einmal und speichert sie dann dauerhaft. Fortan akzeptiert er keine andere Absender-MAC-Adresse mehr am jeweiligen Port. Dies unterbindet MAC-Spoofing-Angriffe wirksam. Der Nachteil: Bei jeder absichtlichen Änderung am Netz muss der Admin auch in die Konfiguration des Switch eingreifen. ►

Grundlagen: Adressen im LAN

Wollen in einem Netz zwei Rechner miteinander kommunizieren, müssen sie sich eindeutig adressieren. Im Ethernet dient dazu eine 48-Bit-Nummer (6 Byte), die der Hersteller vorgibt. Diese so genannte MAC-Adresse (Media Access Control) ist weltweit eindeutig. Damit kann der Anwender beinahe beliebig viele Ethernet-Adapter zu einem LAN verkabeln.

Ethernet funktioniert ohne Switches und Bridges als Broadcast-Medium, sprich: Jedes Paket erreicht jeden Teilnehmer in diesem Netz. Aber nur der adressierte Empfänger nimmt das Paket an, alle anderen ignorieren es. Das Netz funktioniert ohne weitere Konfiguration, echtes Plug & Play.

Einfache Zustellung, aber nur lokal tauglich

Diese Vorgehensweise ist bestechend einfach, skaliert aber schlecht. Schließlich teilen sich alle Teilnehmer das Netz (Shared Medium) und damit dessen Übertragungskapazität. Linderung verschaffen Bridges und Switches: Diese Geräte teilen das Netz in mehrere Segmente und lernen, welche MAC-Adressen an welchem Anschluss zu finden sind (CAM-Tabelle, Content-addressable Memory). Sie senden dann Pakete nur noch in das Segment, in dem sie den gewünschten Empfänger erwarten. Innerhalb eines Segments senden sich die Teilnehmer Datenpakete, ohne die Kommunikation in einem anderen Segment zu stören.

Für weltweite Netze taugt dieses Prinzip immer noch nicht. Jeder Switch müsste für jeden einzelnen Zielrechner wissen, wo er sich befindet. Daher haben die Väter des Internets eine eigene Adressierung eingeführt, die IP-Adresse. Sie ist 32 Bit lang (4 Byte) und besteht aus einem Netz- und einem Host-Anteil. Welcher Teil der Adresse das Netz und welcher den Host festlegt, lässt sich aus der Netzmaske ablesen.

Die einzelnen Netze sind im Internet durch Router verbunden. Die Router müssen nur die Netzadressen kennen und leiten jedes Paket damit in die richtige Richtung. Durch systematisches Vergeben der Adressen gelingt es, IP-Pakete weltweit korrekt und zügig zuzustellen.

MAC und IP: Zwei Adressen für einen Rechner

Während für das Routing ausschließlich die IP-Adressen relevant sind, arbeitet das LAN weiterhin nur mit MAC-Adressen (Abbildung 1). Es wäre aber ungeschickt, wenn jedes Programm sowohl IP- als auch MAC-Adresse eines Rechners kennen müsste, um mit ihm zu kommunizieren. Daher sorgt ARP (Address Resolution Protocol) für die Zuordnung: Zu einer IP-Adresse liefert es die passende MAC-Adresse. Eine Konfiguration ist dazu nicht nötig, der Admin muss also nicht selbst die Adresspärchen IP/MAC einstellen. Die Automatik führt aber zu schweren Sicherheitsproblemen, die dieser Artikel genauer erklärt.

Namen und Nummern

Zusätzlich zu MAC- und IP-Adresse spielen noch Portnummer und Rechnername eine wichtige Rolle. Während MAC und IP nur den Rechner adressieren, unterscheidet der Port die gleichzeitig laufenden Clients und Server (Dienste) innerhalb eines Computers. Der DNS-Name nimmt Rücksicht auf die menschliche Merkfähigkeit: »www.linux-magazin.de« ist einfacher zu behalten als »62.245.157.216«. Hier sorgt der Domain Name Service für die Abbildung von Name auf Nummer, ganz nach dem Vorbild der Telefonbücher.

Neben ARP gibt es noch RARP (Reverse ARP, [3]). Ähnlich wie DHCP teilt ein RARP-Server einer Maschine, die nur ihre eigene MAC-Adresse kennt, eine IP-Adresse zu. Da RARP keine weiteren Parameter überträgt (Nameserver, Gateway-Adresse, Netzmaske), kommt es kaum noch zum Einsatz.

Noch ein Angriffstyp lässt sich mit Port Security abwehren: Das Ziel von MAC-Flooding ist es, die CAM-Tabelle (Content-addressable Memory) des Switch zu deaktivieren. Sie zeigt für jede aktive MAC-Adresse, an welchem Port der jeweilige Rechner hängt.

Der Switch sendet jedes Paket nur an jenen Port, hinter dem er den passenden Rechner kennt. Angreifer setzen diese Funktion außer Kraft, indem sie den Switch mit Adressen überfluten – die CAM-Tabelle nimmt nur eine begrenzte Menge auf. Der Switch arbeitet danach nur noch als Hub, jede Kommunikation ist an jedem Port sichtbar.

Der dritte Angriff ist nicht so leicht zu erkennen, für ihn existieren auch keine einfachen Gegenmaßnahmen. Er beruht auf ARP-Spoofing: Der Angreifer versendet gefälschte ARP-Pakete. Eine besondere Form des ARP-Spoofing ist das ARP-Poisoning. Es verfolgt das Ziel, die ARP-Tabellen anderer Systeme zu manipulieren (zu vergiften).

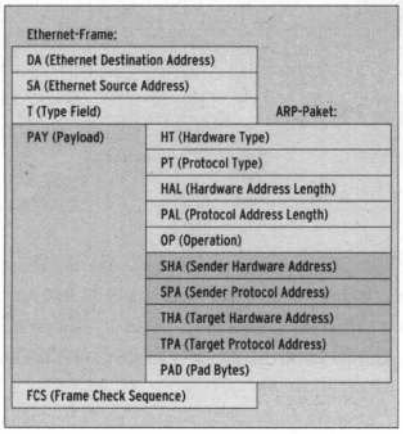

Abbildung 3: Ein ARP-Paket wird direkt als Nutzlast eines Ethernet-Frame übertragen, benutzt also selbst kein IP. Nach den Feldern, die Typ und Länge der Adressen beschreiben, enthält jedes Paket die Angaben für Sender und Empfänger.

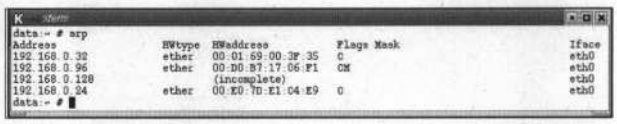

Abbildung 4: Die ARP-Tabelle eines Linux-Systems mit einem unvollständigen, einem statischen und zwei dynamischen Einträgen (Flag C: komplett, M: statisch).

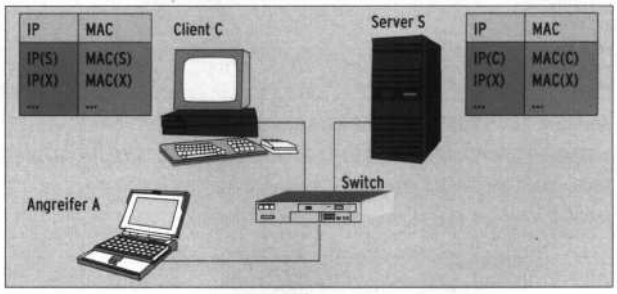

Abbildung 5a: Die Ausgangssituation: Der Client C hat die MAC-Adresse des Servers S korrekt in seiner ARP-Tabelle verzeichnet, ebenso kennt der Server die Adresse des Clients. Der Eintrag X steht stellvertretend für weitere Rechner.

Gift für die ARP-Tabelle

Da Betriebssysteme in der Regel nicht prüfen, ob ein ARP-Reply tatsächlich auf einen kürzlich versandten ARP-Request zurückzuführen ist, übernehmen sie die Adressinformationen des Replys. Bei Windows-Betriebssystemen können Saboteure sogar Einträge verändern, die der Anwender explizit als statisch eingegeben hat.

Ein Datenspion kann somit die Kommunikation zwischen Client und Server angreifen und sich selbst als Man in the Middle in die Kommunikation einklinken. Er manipuliert den Eintrag für den

Tabelle 1: Paketformat

Abkürzung	Feld	Länge in Byte	Erklärung
Ethernet-Frame			
DA	Ethernet Destination Address	6	ARP-Requests werden in der Regel an die Broadcast-Adresse »FF:FF:FF:FF:FF:FF« versandt, während die Antwort gezielt an den Anfragenden geht
SA	Ethernet Source Address	6	Adresse des Absenders
T	Type Field	2	Gibt an, welches Protokoll im Payload-Feld transportiert wird; bei ARP lautet der Eintrag 0x0806
PAY	Payload	46	Nutzlast; ARP-Pakete sind immer 46 Byte lang
FCS	Frame Check Sequence	4	Prüfsumme; damit kann der Empfänger Übertragungsfehler bemerken
ARP-Paket			
HT	Hardware Type	2	Art der Hardware-Adresse; bei Ethernet steht hier der Wert 1 (MAC-Adresse)
PT	Protocol Type	2	Für welches Protokoll die Adresszuordnung gesucht wird; bei IP steht hier 0x0800
HAL	Hardware Address Length	1	Eine MAC-Adresse (Ethernet) hat 6 Byte
PAL	Protocol Address Length	1	Eine IPv4-Adresse hat 4 Byte
OP	Operation	2	Die gewünschte Aktion; bei einem ARP-Request steht hier der Wert 1, ein ARP-Reply hat den Operation-Wert 2
SHA	Sender Hardware Address	6	Hardware-Adresse des Absenders, die Länge entspricht dem Wert im HAL-Feld (bei Ethernet: MAC-Adresse, Länge: 6 Byte)
SPA	Sender Protocol Address	4	Protokoll-Adresse des Absenders, die Länge entspricht dem Wert im PAL-Feld (üblicherweise die IP-Adresse, Länge 4 Byte)
THA	Target Hardware Address	6	Die Hardware-Zieladresse eines ARP-Requests ist »0:0:0:0:0:0«; da ARP-Replys direkt zugestellt werden, steht dort die MAC-Adresse des Anfragenden
TPA	Target Protcol Address	4	IP-Adresse des Ziels
PAD	Pad Bytes	18	Jedes ARP-Paket ist 46 Bytes lang; bei der Kombination Ethernet und IPv4 bleiben 18 Byte übrig, die durch Padding aufgefüllt werden

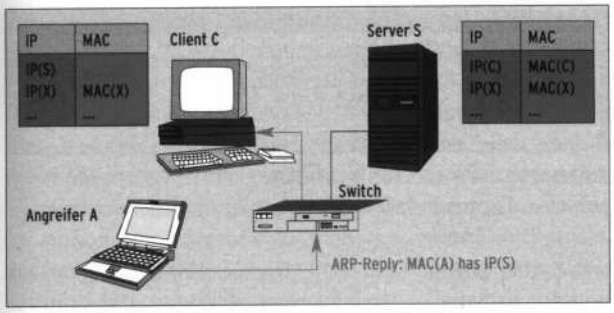

Abbildung 5b: Im ersten Schritt des Man-in-the-Middle-Angriffs sendet der Angreifer A ungefragt einen ARP-Reply an Client C. Der Client trägt folgsam die MAC-Adresse von A in die ARP-Tabelle ein, und zwar als Adresse für den Server S.

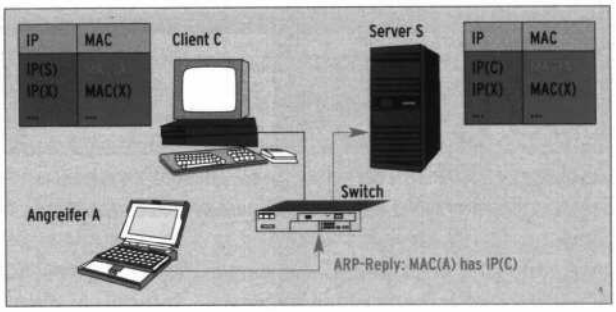

Abbildung 5c: Ein zweiter ARP-Reply – diesmal an den Server S – vervollständigt den Angriff. Jetzt denkt der Server, dass die MAC-Adresse von A dem Client C gehört. Für C gedachte Pakete wird er daher fälschlich an A senden.

Server in der ARP-Tabelle des Clients so, dass seine eigene MAC-Adresse der IP-Adresse des Servers zugeordnet wird. Die entsprechende Manipulation erledigt er beim Server (Abbildungen 5a bis 5c).

Unerwünschter Router

Will der Client mit dem Server kommunizieren, wird er seiner manipulierten ARP-Tabelle folgen und das Paket an die MAC-Adresse des Angreifers schicken. Der Saboteur kann es lesen und ändern, bevor er es an den Server weiterleitet. Als Man in the Middle gleicht er einem Router (Abbildung 1), der nur über ein Netzinterface verfügt. Der Empfänger geht davon aus, dass das Paket direkt vom Client kommt. Die Antwort des Servers geht ebenfalls ungewollt an den Angreifer, der sie wieder an den Client weiterleitet. Falls der Server in einem anderen Subnetz steht, wird einfach die Verbindung zwischen Client und lokalem Router angegriffen.

Statt Daten abzuhören erreicht der Angreifer sehr einfach einen Denial of Service, wenn er die zu ihm umgeleiteten Pakete verwirft. Auch Informationen zu manipulieren ist leicht, der Saboteur muss nur andere Daten weiterleiten als er empfängt. Das Sammeln von Passwörtern ist ebenfalls möglich, aus der Portnummer lassen sich Rückschlüsse auf das verwendete Protokoll ziehen und mit dieser Kenntnis Benutzername und Passwort finden.

Vorsicht bei SSL und SSH

Sogar verschlüsselte Verbindungen sind nicht automatisch immun, wie sich mit einigen ARP-Tools belegen lässt. Diese Programme sind für verschiedene Betriebssysteme verfügbar (siehe Kasten „ARP-Angriffssoftware"). Neben der ARP-Poisoning-Funktion enthalten sie Client- und Server-Implementierungen für SSL (Secure Socket Layer), TLS (Transport Layer Security), SSH (Secure Shell) oder PPTP (Point to Point Tunne-

ARP-Angriffssoftware

Die folgende Liste zeigt einige Beispiele für Software, mit der Angreifer die ARP-Probleme ausnutzen oder Admins ihre eigenen Netze testen. Die Tools eignen sich gut, um Ungläubigen die Tragweite der ARP-Attacken zu verdeutlichen. Nicht die Existenz dieser Programme ist das Problem, sondern ARP.

ARP-SK: Die Programmierer bezeichnen ihr Tool als Schweizer Messer für ARP, es ist als Unix- und Windows-Version verfügbar. Das Programm kann die ARP-Tabellen verschiedenster Geräte manipulieren. [http://www.arp-sk.org]

Arpoc und WCI: Dieses Unix- und Windows-Programm fängt Verbindungen im LAN ab (Man in the Middle). Es reagiert auf jeden ARP-Request, der den Rechner erreicht, mit einem gefälschten ARP-Reply. Pakete, die nicht lokal zugestellt werden sollen, leitet das Programm an den passenden Router weiter. [http://www.phenoelit.de/arpoc/]

Arpoison: Kommandozeilentool, das gefälschte ARP-Pakete erzeugt. Der Anwender legt die Absender-IP/MAC und Ziel-IP/MAC beliebig fest. [http://arpoison.sourceforge.net]

Brian: Dieses sehr einfache Tool (nur ein C-File) benutzt ARP-Poisoning, um das Switching in einem LAN zu deaktivieren. Damit lässt sich der gesamte Datenverkehr abhören. [http://www.bournemouthbynight.co.uk/tools/]

Cain & Abel: Die ausgefeilte Windows-Software war ursprünglich ein Passwort-Recovery-Tool. Das Programm hört das Netzwerk ab und versucht mit unterschiedlichen Techniken verschlüsselte und getarnte Passwörter zu dechiffrieren. Seit Version 2.5 nutzt das Tool auch ARP-Poisoning, um im geswitchten LAN den IP-Verkehr abzufangen. Das Programm greift auch SSH- und HTTPS-Verbindungen an. [http://www.oxid.it/cain.html]

Dsniff: Die einzelnen Programme dieser Tool-Sammlung erfüllen verschiedene Aufgaben. Dsniff, Filesnarf, Mailsnarf, Msgsnarf, Urlsnarf und Webspy belauschen das Netz und fischen interessante Daten heraus (etwa Passwörter, E-Mail und Dateien). Mit Arpspoof, Dnsspoof und Macof kommen Admins und Angreifer an Daten, die ihnen das Switching im LAN eigentlich vorenthält. Per Sshmitm und Webmitm sind auch Man-in-the-Middle-Angriffe auf SSH und HTTPS möglich (hier übrigens Monkey in the Middle genannt). [http://naughty.monkey.org/~dugsong/dsniff/]

Ettercap: Ein recht mächtiges Programm mit bequemer Textmodus-Oberfläche (siehe Abbildung 7), die aktuelle Version bringt auch ein GTK-Interface mit. Alle Aktionen laufen automatisch, mögliche Angriffsziele listet das Tool in einem Fenster. Neben Sniffing, ARP-Attacken und dem automatischen Passwortsammeln kann Ettercap auch die Daten innerhalb einer Verbindung manipulieren. Das Programm greift auch SSHv1- und SSL-Verbindungen an. [http://ettercap.sourceforge.net]

Hunt: Dringt in eine Verbindung ein, hört sie ab und kann sie übernehmen (Session Hijacking). Das Tool benutzt unter anderem ARP-Spoofing. [http://packetstormsecurity.nl/sniffers/hunt/]

Juggernaut: Das Phrack-Magazin veröffentlichte 1997 mit Juggernaut den Vorläufer vieler heute verfügbarer Sniffer mit ARP-Cache-Poisoning-Funktion. [http://www.phrack.org/show.php?p=50&a=6]

Parasite: Der Parasite-Daemon hört das LAN ab und reagiert auf jeden ARP-Request mit einem gefälschten ARP-Reply. Im Laufe der Zeit fungiert der Rechner als Man in the Middle für alle Kommunikation im LAN. [http://www.thc.org/releases.php]

Abbildung 6a: Die ARP-Tabelle des Rechners »odo« enthält für die IP-Adresse 192.168.1.124 vor dem Angriff die korrekte MAC-Adresse »00:D0:B7:0B:B6:4A«.

Abbildung 6b: Während des Angriffs ändert sich die zu 192.168.1.124 gehörende MAC-Adresse auf »00:60:97:B4:44:07« – sie gehört dem Angreifer-Rechner.

ling Protocol). Beim Zugriff auf einen SSL-Webserver warnt der Browser zwar, dass mit dem Zertifikat dieser Verbindung etwas nicht stimmt. Viele Anwender können die Warnung aber nicht richtig deuten und ignorieren sie.

Dass viele Webserver mit einem selbst generierten oder abgelaufenen Zertifikat arbeiten und die Warnung daher bekannt ist, verstärkt diesen Warnung-wegklicken-Effekt noch. Durch einen Bug in manchen Internet-Explorer-Versionen ist es sogar möglich, SSL-Verbindungen anzugreifen, ohne dass der Browser eine Warnung anzeigt.

Der Angriff auf SSH erfolgt ähnlich (Abbildung 7). Wenn der Client den Host-Key des Servers bereits kennt, gibt er eine sehr deutliche Warnung aus (Abbildung 8). Aber sogar die wird von vielen Benutzern und Admins ignoriert – sie vermuten, dass jemand den SSH-Schlüssel des Servers neu erzeugt hat. Nur wenige Protokolle (genauer: deren Implementierungen) sind immun, ein Beispiel dafür ist IPsec. Es verweigert pauschal den Dienst, wenn bei der Authentifizierung etwas schief läuft.

Angriffe erkennen

Für den Netzbetreiber bedeutet dies, dass beinahe jede interne Kommunikation angreifbar ist. Es gibt sogar Einsteiger-taugliche Software, die Passwörter in über 50 Protokollen analysiert. Da die Angriffe auf ARP-Ebene arbeiten, aber meist nur IP-Zugriffe geloggt werden, darf sich der Angreifer heutzutage sogar recht sicher sein, dass niemand seine Taten aufdeckt. Damit drängt sich die Frage auf, wie ein Admin verhindern kann, dass interne Mitarbeiter ARP-Angriffe durchführen.

Ein Ansatz ist, das Herunterladen und Ausführen fremder Software zu unterbinden. Das ist aber kaum durchzusetzen: Die Internet-Anbindung wäre stark einzuschränken, schließlich ist es mit HTTP, HTTPS, FTP und E-Mail leicht, Software in das interne Netz zu holen. Auch mobile Datenträger wie Disketten oder CDs müssten die Admins verbieten, ebenso mobile Geräten wie Notebooks oder PDAs. Wegen der gravierenden Einschränkungen eignet sich diese Lösung nur in Ausnahmefällen.

Wer im internen Netz nur Linux einsetzt und seinen Benutzern keine Root-Rechte gibt, vermeidet die meisten Angriffe von vornherein, denn das Senden gefälschter ARP-Paket erfordert Root-Rechte. Allerdings sollte sich kein Admin zu sicher sein, dass nicht doch ein Mitarbeiter einen Rechner von CD bootet oder sein privates Notebook anschließt.

Statische Einträge in den ARP-Tabellen

Auch statische ARP-Einträge (per »arp -s« gesetzt) können ARP-Angriffe scheitern lassen. Wegen des hohen Aufwands wird kaum jemand mehr als die wichtigsten Systeme (Router und Server) manuell eintragen. Bei Microsoft-Betriebssystemen können Angreifer selbst diese Einträge durch ARP-Poisoning ändern – es gibt letztlich keinen Schutz.

Der Ansatz ist aber nur für kleine Netze sinnvoll, da die Anzahl der ARP-Einträge quadratisch mit der Zahl der Netzwerkkarten steigt: Für 100 Systeme wären 9900 Einträge erforderlich (je 99 pro System). Der Administrationsaufwand ist besonders beim Suchen und Beheben von Netzwerkfehlern enorm.

Wachsames Auge

Mit Arpwatch [4] gibt es seit vielen Jahren ein Open-Source-Tool für Unix-Plattformen, das auffällige ARP-Aktivitäten aufspürt. Jedem ARP-Paket, das den Arpwatch-Rechner erreicht, entnimmt es Adressinformationen und speichert sie in einer Datenbank. Stimmen die Daten nicht mit früher gesammelten Adresspaarungen überein, warnt das Tool den Admin per E-Mail. Arpwatch soll Geräte auch per SNMP abfragen können. Das hat im Test eines der Autoren dieses Artikels aber nicht funktioniert.

Viele Netze verwenden heute dynamische IP-Adressen, die ein interner Server per DHCP (Dynamic Host Configuration Protocol) verteilt. In diesem Umfeld liefert Arpwatch häufig Fehlalarme (false positive), da es jede geänderte IP-MAC-Zuordnung meldet.

ARP-Guard [5] ist ein recht neues Produkt der Firma ISL. Es arbeitet mit einer Sensor-Management-Architektur: Mehrere Sensoren beobachten an beliebiger Stelle im Netz die ARP-Informationen und leiten sie an das Managementsystem weiter, das die Meldungen analy-

Abbildung 7: Ettercap wartet auf Verbindungen zwischen 192.168.1.120 und 192.168.1.124 (Source und Destination, links oben). Unverschlüsselte Protokolle wie Telnet und FTP kann es problemlos abhören. Bei SSHv1 greift es als Man in the Middle aktiv ein, um die Verbindung zu entschlüsseln.

siert und im Angriffsfall die Administratoren alarmiert. Das Programm verfügt über einen LAN- und einen SNMP-Sensor. Der LAN-Sensor arbeitet ähnlich wie Arpwatch oder ein IDS-Sensor. Er analysiert Pakete, die ihn erreichen. Der SNMP-Sensor greift dagegen per SNMP auf vorhandene Endgeräte zu und fragt deren ARP-Tabelle ab.

Intrusion-Detection-Systeme (siehe **Kasten „Snort und ARP"**) könnten zwar ebenfalls ARP-Attacken erkennen, sie werden meist aber nur am Übergang zu fremdem Netzen eingesetzt. Für viele Firmen rechnet sich der Einsatz im internen Netz nicht. Zudem hat der Betriebsrat meist etwas dagegen, weil der Administrator des IDS-Systems als Big Brother handeln kann: Er sieht den gesamten Netzverkehr und damit alle Zugriffe der Mitarbeiter. Der Nutzen wäre auch begrenzt, da viele IDS-Produkte das ARP-Protokoll nicht beachten. Spätestens an ARP-Poisoning-Angriffe bei dynamischen IP-Adressen scheitern sie.

Kryptographie hilft

Wenn Vertraulichkeit, Authentizität und Integrität der Daten durch kryptographische Protokolle (vor allem IPsec) garantiert werden, beschränken sich ARP-Attacken auf Denial-of-Service-Angriffe. Abhör- und Manipulationsangriffe scheitern dagegen. Es wird aber noch einige Zeit dauern, bis IPsec und andere Kryptoprotokolle umfassend zum Einsatz kommen, korrekt konfiguriert sind und von allen Anwendern auch richtig verstanden und angewendet werden.

Eine Forschergruppe schlägt vor, das herkömmliche ARP durch eine sichere Version zu ersetzen **[7]**. Ihre Entwicklung S-ARP nutzt Kryptographie, eine CA (Certifcation Authority) und digital signierte ARP-Nachrichten. Es ist fraglich, ob der Aufwand lohnt: IPsec bietet einen wesentlich weiter reichenden Schutz bei ähnlichem Aufwand, während S-ARP nur ARP sichert. Allein die geringere Rechenlast auf den beteiligten Systemen spricht für S-ARP.

Hersteller von Firewalls oder Routern behaupten gern, ihre Produkte könnten ARP-Spoofing-Angriffe erkennen. Das trifft aber nur eingeschränkt zu, denn diese Systeme können höchstens die Änderung eines ARP-Eintrags in ihrer ARP-Tabelle erkennen und protokollieren. Sie können nicht wissen, aus welchem Grund die Änderung erfolgt. Mögliche Ursachen sind dynamische IP-Adressen, Adresskonflikte, Netzänderungen sowie Hochverfügbarkeits- und Lastverteilungslösungen (High Availability, HA, und Load Balancing, LB). Darüber hinaus bemerkt eine Firewall meist nur ARP-Angriffe gegen sie selbst. Attacken auf den Server direkt nebenan bleiben im Verborgenen.

Der Netzverkehr lässt sich auch mit managebaren Switches einschränken. Dieses Mittel hilft nicht nur gegen ARP-Angriffe, sondern eignet sich für Verkehrsbeziehungen im Netz generell. Damit laufen allerdings einige Anwendungen nicht mehr. VLANs (virtuelle LANs), Port Protection und Zugriffskontrolllisten von Layer-3/4/5/6/7-Switches zählen zu diesen Ansätzen. Neben der Einschränkung der Verkehrsbeziehungen sind diese Techniken teils recht teuer, ein Layer-3-Switch kostet etwa das Vierfache eines programmierbaren Layer-2-

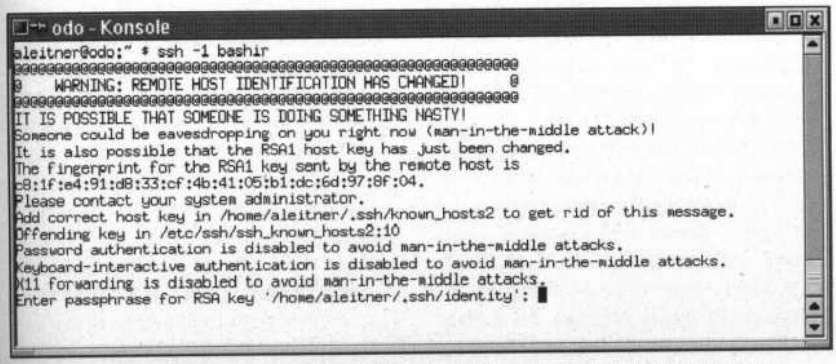

Abbildung 8: Während des Ettercap-Angriffs (Abbildung 7) erhält der Client (hier »odo«) einen falschen Host-Key vom Server (Name »bashir«, IP-Adresse 192.168.1.124). Der Key stammt vom Angreifer und nicht vom gewünschten Server (»bashir«). Wer diese Warnung ignoriert, ist abhörbar.

Switch. Zudem steigt der Aufwand beim Administrieren, da man die Switches individuell konfigurieren muss.

Denkbar wäre es auch, ein Netz in so viele Subnetze aufzuteilen, dass sich möglichst wenige Mitarbeiter ein Subnetz teilen. Ideal wäre eine einzelne IP pro Subnetz. Die Hersteller von Routern könnten diese Lösung sicherlich empfehlen, da hohe Investitionen in entsprechende Geräte nötig wären. Der Administrationsaufwand wäre ebenfalls unangemessen hoch.

Gegenmaßnahmen im Stapel

Einige Entwickler versuchen Schutzfunktionen in den IP-Stack zu integrieren. Mit dem Antidote-Patch [8] sendet Linux vor der Änderung eines ARP-Eintrags erst einen Request an die alte MAC-Adresse. Nur wenn dieser unbeantwortet bleibt, ändert das System den Eintrag. Die Schutzwirkung ist begrenzt: Der Angreifer muss nur dafür sorgen, dass die Änderung dann erfolgt, wenn der andere Rechner nicht aktiv oder nicht erreichbar ist. Bei vielen HA- oder LB-Lösungen kann dieses Patch zudem die Kommunikation mit den – meist sehr wichtigen – Systemen stören.

Ein weiterer Ansatz, sich vor ARP-Poisoning zu schützen, ist das Unterbinden jeder Änderung vorhandener MAC-IP-Zuordnungen. Das Anticap-Patch [9] implementiert dieses Verhalten für Linux, FreeBSD und NetBSD. Solaris verfügt über eine ähnliche Option, es erlaubt Änderungen erst nach Ablauf eines Timers. Das Verhalten lässt sich frei konfigurieren. Es schützt jedoch ebenfalls nur ständig aktive Systeme – nach dem Rausaltern des Eintrags können Angreifer neue Felder fälschen.

Seit Kernel 2.4 reagiert Linux nicht mehr auf unverlangt zugesandte ARP-Replys. Leider ist auch dieser Schutz leicht zu umgehen, wie die Readme-Datei von Ettercap erklärt (siehe **Kasten „ARP-Angriffssoftware"**). Der Hintergrund: Einen ARP-Request muss der Kernel immer verarbeiten. Da er dabei auch eine Kombination aus IP und MAC erfährt (die des Absenders), trägt er diese vorsorglich in seinen ARP-Cache ein. Der Angreifer muss also nur einen gefälschten ARP-Request senden. Ettercap schickt eine Kombination aus ARP-Request und -Reply: Auf einen der beiden Angriffe reagiert jedes System.

Netzlast oder Sicherheit

Ein Nachteil der meisten Schutzmechanismen ist die höhere Netzlast durch zusätzliche ARP-Pakete. Viele Mechanismen führen auch dazu, dass sich Änderungen (etwa ausgetauschte Ethernet-Karten) erst verspätet in allen ARP-Caches niederschlagen. Die IP-Stacks versuchen normalerweise möglichst schnell und lastschonend neue MAC-IP-Kombinationen zu lernen. Ein Verzicht darauf verhindert zwar einige ARP-Poisoning-Angriffe, geht aber auf Kosten der Netzwerk-Performance.

Gegen ARP-Spoofing sind alle Schutzmechanismen im IP-Stack machtlos. Wenn der Angreifer schneller auf einen ARP-Request reagiert als der Gesuchte, gewinnt er diese Race Condition und hat seine Adresse erfolgreich in den ARP-Cache eingetragen.

Kaum Schutz

Mit heutigen Techniken und Protokollen ist ein umfassender Schutz vor ARP-Angriffen kaum möglich. Mit IDS und spezialisierten ARP-Angriffserkennern fliegen viele Manipulationsversuche immerhin auf. Wer sichergehen will, muss im internen Netz konsequent IPsec einsetzen. Ungestraft das Problem ignorieren darf eigentlich nur, wer jedem Einzelnen vertraut, der Zugriff auf sein LAN hat – egal an welcher Stelle. ■

Infos

[1] KPMG-Studie: [http://www.kpmg.com/about/press.asp?cid=469]
[2] Address Resolution Protocol, RFC 826: [http://www.ietf.org/rfc/rfc826.txt]
[3] Reverse APR, RFC 903: [http://www.ietf.org/rfc/rfc903.txt]
[4] Arpwatch: [http://www-nrg.ee.lbl.gov] und [http://www.securityfocus.com/tools/142]
[5] ARP-Guard: [https://www.arp-guard.com]
[6] Snort: [http://www.snort.org]
[7] Secure ARP: [http://security.dico.unimi.it/research.en.html#sarpd] und [http://www.acsac.org/2003/papers/111.pdf]
[8] Antidote-Patch: [http://www.securityfocus.com/archive/1/299929]
[9] Anticap-Patch: [http://cvs.antifork.org/cvsweb.cgi/anticap/]

Snort und ARP

Snort [6] ist ein prominentes Beispiel für ein Netzwerk-IDS. Das Intrusion Detection System hilft dem Admin, um Angriffe im Netz frühzeitig zu erkennen und Gegenmaßnahmen einzuleiten. Snort verfügt über einen Arpspoof-Präprozessor mit vier Erkennungsmechanismen:

- In jedem ARP-Request vergleicht der Präprozessor die Absenderadresse im Ethernet-Frame mit der Absenderadresse im ARP-Paket. Falls sie nicht übereinstimmen, löst er eine Meldung aus. Für ARP-Poisoning ist es aber nicht nötig, in diesen Feldern unterschiedliche Adressen zu verwenden, daher bleiben Angriffe unbemerkt.
- Bei ARP-Replys betrifft der Vergleich sowohl Absender- als auch Empfängeradressen. Falls auch nur eines der Adresspaare ungleich ist, löst Snort eine Meldung aus. Auch hier bleibt ARP-Poisoning unbemerkt. Lediglich Proxy-ARP wird erkannt – allerdings ist dies Verfahren meist legitim. Eine Maschine beantwortet hier ARP-Requests stellvertretend für eine andere.
- ARP-Requests, die nicht an die Broadcast-Adresse, sondern an eine Unicast-Adresse gehen, meldet der Arpspoof-Präprozessor ebenfalls. Obwohl dieses Verhalten nicht dem (mittlerweile über 20 Jahre alten) Standard entspricht, gibt es durchaus gute Gründe für dieses Vorgehen. Bei echten ARP-Angriffen besteht aber keine Notwendigkeit dafür, ARP-Requests per Unicast abzusetzen. Daher erkennt auch dieser Mechanismus keine ARP-Poisoning-Attacken.
- Anhand einer Liste von IP- und MAC-Adressen, die der Admin selbst anlegen muss, kontrolliert Snort alle ARP-Pakete. Falls diese Absender-IP-Adresse in der Liste enthalten ist, nimmt das IDS die zugehörige MAC-Adresse aus der Liste und vergleicht sie mit den Absender-MAC-Adressen aus dem ARP-Paket und dem Ethernet-Frame. Bei Abweichungen schlägt Snort Alarm. Dieser Mechanismus eignet sich nur für kleine Netze, da der Konfigurationsaufwand andernfalls zu hoch wird. Bei dynamischen IP-Adressen (DHCP) ist ein sinnvoller Einsatz kaum möglich.

Snort ist also – wie bei Intrusion-Detection-Systemen üblich – nur eingeschränkt zur Erkennung von ARP-Poisoning einsetzbar.