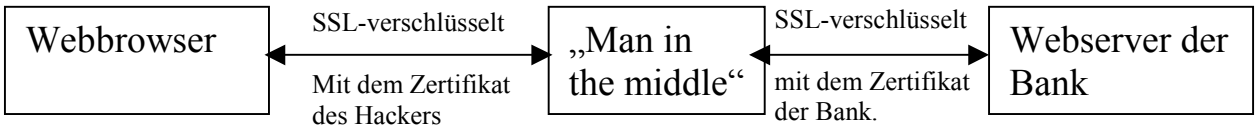
Raffinierte Attacken auf SSL-Verbindungen – nachgestellt im SySS-Labor (Sebastian Schreiber Schreiber@SySS.de)

Angriffe vom Typ "Man-in-the-middle"

Stellen Sie sich vor, Sie starten Ihren Webbrowser um Internet-Banking zu betreiben. Wie es sich gehört[1], wird für die Übertragung der sensiblen Daten eine verschlüsselte Verbindung (HTTPS, SSL) aufgebaut, die gewährleistet, daß Ihre Daten während des Transports nicht mitgelesen werden können. Findige Hacker haben dennoch Zugriff auf Ihre Daten: sie sorgen mittels einer der bekannten Spoofing-Attacken[2] dafür, daß Ihr Webbrowser nicht den gewünschten Internetbanking-Server kontaktiert, sondern den Server des Hackers. Auf Ihre http-Anfrage hin, kontaktiert der Server des Hackers den Server Ihrer Bank. Sie glauben also, mit Ihrer Bank zu kommunizieren - und der Server der Bank geht davon aus, einen Kunden zu bedienen. In Wirklichkeit kommunizieren beide legitimen Kommunikationspartner mit dem Server des Hackers:



Gegenmaßnahmen: das SSL-Zertifikat

HTTPS/SSL sehen Authentifikationsmechanismen vor, die solche M.i.M.-Attacken vereiteln. Ein vertrauenswürdiges Unternehmen („*Certification Authority*", CA) stellt Zertifikate aus, die dem Benutzer versichern, mit welchem Unternehmen er kommuniziert. Es stellen sich nun zwei Fragen:

1. Wie entscheidet der Benutzer, welche Unternehmen besonders vertrauenswürdig sind, und daher als CA taugen?
2. Wie prüft der Benutzer, ob das von der CA ausgestellte Zertifikat auch authentisch ist[3]?

Beide Fragen sind vom Endnutzer kaum zu beantworten. Als Beispiel soll hier gezeigt werden, wie sich der Versuch, Internet-Banking zu betreiben, einem Benutzer des Kreissparkassen-Bankingservers präsentiert: exemplarisch wird der Internetbankingservice der Kreissparkasse genutzt:

1. Wir wählen http://www.ksk-tuebingen.de im Browser an.

---

[1] Leider fordern unzählige Webseiten den Benutzer auf, Kreditkarteninformationen über unverschlüsselte Verbindungen zu übertragen. Bsp: www.eHotel.de.

[2] Unter Spoofing versteht man das Vorgeben einer falschen Identität. Gängige Attacken sind DNS-Spoofing, IP-Spoofing und Arp-Spoofing. IP- und MAC-Adressen sind leicht zu fälschen – dennoch werden sie bei Switches oder Firewalls oft als wichtigstes Filterkriterium eingesetzt.

[3] Microsoft stattet den Internet Explorer bereits mit einer Vielzahl von Zertifikaten von Root-CAs aus. Eine Garantie, daß die Zertifikate authentisch sind, oder daß die CAs vertrauenswürdig sind, übernimmt Microsoft allerdings nicht.

2. Wir klicken auf "Internet Banking", und werden zur uns völlig unbekannten Firma rwso.de umgeleitet. „Das ist ja seltsam“, denkt der Normalverbraucher. Da eine HTTPS/SSL-Verbindung etabliert werden soll, fordert der Internet Explorer das Zertifikat von RWSO und warnt, das Zertifikat sei „*fehlerhaft*“:
   “*Das Sicherheitszertifikat wurde von einer Firma ausgestellt, die Sie nicht als vertrauenswürdig eingestuft haben. Untersuchen Sie das Zertifikat um festzustellen, ob Sie der Institution vertrauen möchten*“.
3. Wie untersucht nun der „Ottonormalverbraucher“ ein SSL-Zertifikat? Bei der genauen Betrachtung wird ersichtlich, dass das Zertifikat von einer ebenfalls unbekannten Firma „Thawte“ aus dem gleichermaßen unbekannten Land mit der Länderkennung „ZA“ kommt. Eine genaue Recherche ergibt, daß die Firma Thawte ihren Sitz in Kapstadt in Südafrika hat. Der Anwender staunt: „Warum bemüht die Kreissparkasse ein Unternehmen in Südafrika um mir ihre Identität plausibel zu machen?“

Es überrascht, dass der Normalverbraucher diesen mit Ungereimtheiten gespickten Vorgang akzeptiert, und nach einigen Klicks auf diverse „Yes“- und „Accept“-Buttons vollen Zugriff auf sein Erspartes freigibt. Es ist davon auszugehen, dass weniger als 5% der Internetbanking-Anwender überhaupt verstehen, wie die Authentisierung des Bankingservers funktioniert.
Dennoch akzeptieren die Benutzer das Medium Internet - und klicken bei Rückfragen des Browsers fast automatisch auf „Ja“ oder „Akzeptieren“.

Technische Durchführung einer M.i.M.-Attacke gegen SSL

Man-in-the-middle-Attacken wurden lange Zeit als sehr unwahrscheinlich bewertet. Im Internet sind aber Tools verfügbar, die – geschickt kombiniert – eine schlagkräftige M.i.M-Attacke implementieren. Im SySS-Labor haben wir folgende Konfiguration aufgebaut:

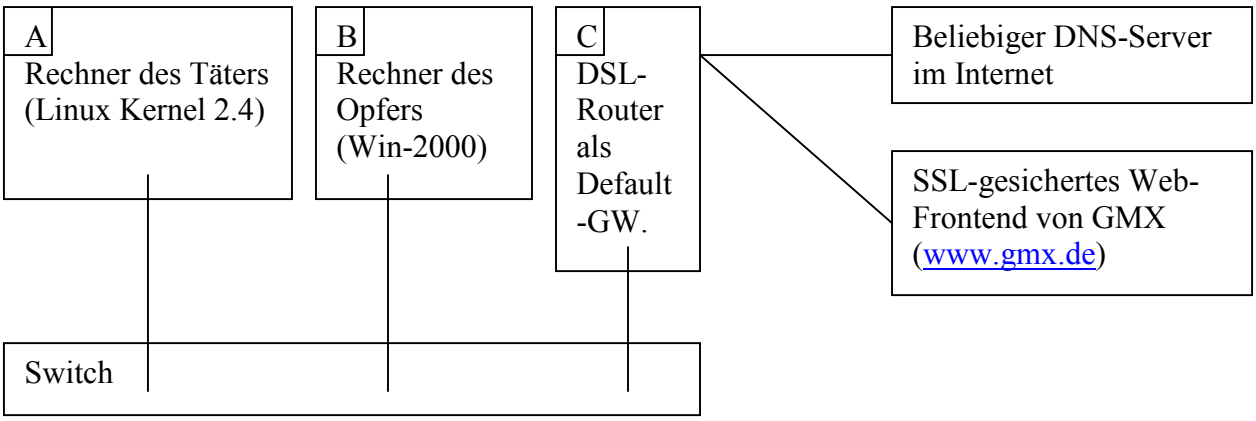


Abbildung 1: Testnetz im SySS-Labor

Das Opfer nutzt den Internet Explorer, um auf dem von GMX angebotenen Web-Interface E-Mail zu lesen. GMX schützt diesen Vorgang mit SSL. Es wird schnell klar, dass eine Hackertaktik allein nicht zum Ziel führt. Kombiniert man aber verschiedene Angriffsvarianten, so gelingt die Attacke. Aus der Perspektive des Hackers sollen hier die drei Schritte erläutert werden, aus denen sich der Angriff zusammensetzt.

1. Überlistung des Switchs: ARP-Spoofing

Im Gegensatz zu einem Hub hat ein Switch die Eigenschaft, dass die Kommunikation zwischen zwei Stationen von keiner der anderen Stationen abgehört werden kann. Zu diesem Zweck pflegt der Switch eine Tabelle, die die MAC-Adressen der beteiligten Stationen dem Port zuordnet, der zum betreffenden Rechner führt:

MAC-Adresse	**Port**
00:80:56.A3.03.CA	2 (der Rechner A)
00:50:53.B3.03.C1	3 (der Rechner B)
00:82:56.A3.03.CA	4 (der Rechner C)

Doch nicht nur der Switch pflegt eine ARP-Tabelle, sondern auch die beteiligten Rechner. Ziel des Hackers ist es nun, diese Tabellen zu manipulieren (*ARP-Cache-Poisoning)*: Dem Rechner B („Opfer“) wird suggeriert, dass der Rechner C („DSL-Router“) eine fiktive Mac-Adresse (MAC_H1) hat. Möchte B nun ein Paket zum (als Default-Gateway eingetragenen) DSL-Router schicken, so landet dieses Paket in Wirklichkeit beim Hacker-Rechner A, der so programmiert wurde, dass er Pakete mit der MAC-Adresse MAC_H1 akzeptiert. Auch der ARP-Cache von C wird so manipuliert, dass Ethernet-Pakete, die eigentlich zu B kommen sollten in Wirklichkeit bei A landen. Diese Manipulation lässt sich mit dem Tool „Hunt“ von Pavel Krauz[4] leicht durchführen: der durchgehende Pfeil stellt die normale Kommunikation dar, der gestrichelte Pfeil zeigt den Umweg über einen Man-in-the-middle. Dabei können für MAC_{H2} und MAC_{H1} jeweils frei erfundene MAC-Adressen verwendet werden.

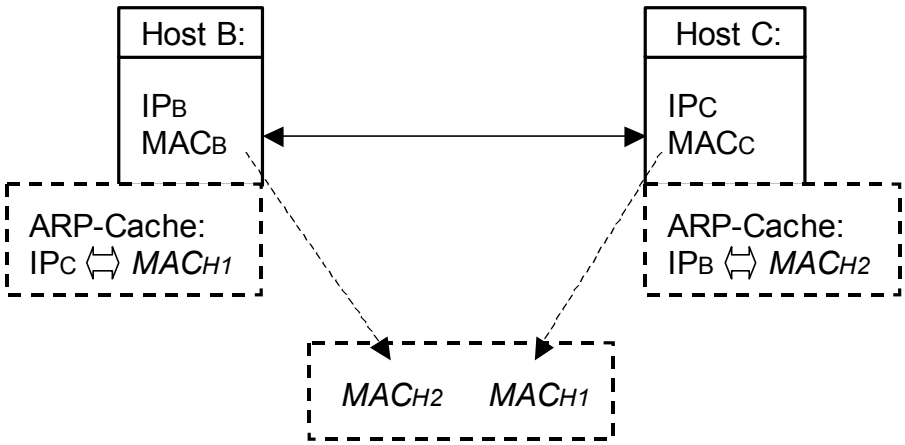


Abbildung 2: ARP-Spoofing mit Hunt

Nun landen sämtliche Pakte, die zwischen B und C ausgetauscht werden, beim Hacker A. Damit die Kommunikation nicht unterbrochen wird, muss auf A ein Relay-Deamon installiert werden, der die Pakete an ihren rechtmäßigen Empfänger weiterleitet. Nun haben wir erreicht, dass die Kommunikation zwischen dem Opfer und dem DSL-Router über den unseren Hacker-Rechner läuft.

[4] Siehe: http://lin.fsid.cvut.cz/~kra/index.html. Hunt lässt sich übrigens auch hervorragend zum Hijacking von Telnet-Verbindungen oder zum Sabotieren beliebiger TCP-Verbindungen nutzen.

2. DNS-Spoofing

Möchte nun der Benutzer B eine Verbindung zu www.gmx.de aufbauen, so wird zunächst der DNS Server nach der IP-Adresse von www.gmx.de gefragt. Unter Anwendung von Dug Songs[5] Tool Dnsspoof wird dafür gesorgt, dass der DNS-Auflösungs-Query des Browsers von B falsch beantwortet wird: der Browser kommuniziert darauf nicht mit www.gmx.de, sondern mit dem Webserver des Hackers.

3. SSL-Spoofing

Nun wird mit dem Tool „webmitm“ ein Zertifikat erzeugt, das, um vom Benutzer akzeptiert zu werden, dem Originalzertifikat von GMX möglichst ähnlich sein sollte. Dieses wird dem Client an Stelle des Originalzertifikats übermittelt und dient zur Etablierung einer verschlüsselten Verbindung.

Die Warnung, dass dieses Zertifikat unbekannt ist, die Signatur also nicht von einer im Browser hinterlegten Root-CA (Certificate Authority) bestätigt wurde, wird in der Regel mit einem reflexartigen Klick auf den „Ja“-Button des entsprechenden Dialogs akzeptiert.

Webmitm reicht die vom Client übermittelten Daten an den korrekten Server weiter, nicht ohne die Daten zuvor unverschlüsselt in Form einer Logdatei dem Angreifer zur Verfügung zu stellen. Wenn sich der Benutzer bei GMX angemeldet hat, steht in dieser Datei die LoginID und das Passwort des Benutzers, das über die – vermeintlich – sichere SSL-Verbindung transportiert wurde.

Gegenmaßnahmen

Es existieren Maßnahmen, die solche Attacken unmöglich machen – in der Praxis sind sie aber schwer zu realisieren:

Wirksame Maßnahme	**Problem/Nachteil**
Die MAC-Adressen der Rechner sind im Switch fest einzuprogrammieren. ARP Spoofing wird so unmöglich.	Die Wartung der Switches wird aufwendig.
Die übermittelten Zertifikate sind zu überprüfen. Dies kann beispielsweise durch einen Telefonanruf beim Zertifikatsinhaber und dem Abgleich der Fingerprints des Keys durchgeführt werden.	Momentan haben Webseitenbetreiber keine Geschäftsprozesse zur Überprüfung von Key-Fingerprints implementiert.
Die DNS-Infrastruktur auf die sich das Internet stützt, ist durch eine sichere Infrastruktur zu ersetzen.	Aufgrund der enormen Größe des Internets und der riesigen Installationsbasis der Clients ist dies schlichtweg unmöglich.
Schulung der Internet-Benutzer	Hohe Komplexität der Thematik

Bemerkenswert bei der beschriebenen Attacke ist, daß keinerlei Softwarebugs ausgenutzt werden: Sämtliche Angriffsschritte basieren auf systemimmanenten Schwächen von Ethernet, IP, TCP und DNS. E-Commerce ist und bleibt unsicher – das wird den Anwender aber nicht daran hindern, daß Internet mehr denn je zu geschäftlichen wie privaten Zwecken einzusetzen.

[5] http://www.monkey.org/~dugsong/