Piratenpartei Rheinland-Pfalz • Postfach 3309 • 55023 Mainz

Remote-Zugang zur Piratendatei

Nachstehend wird der Zugang zur Piratendatei beschrieben, wie er mittels eines Browsers über eine Internet-Verbindung in gesicherter Form erfolgt. Es kommen mehrere anerkannte Verfahren zum Einsatz um eine mehrfach gestufte Sicherheit zu realisieren. Bei den Verfahren handelt es sich im Wesentlichen um:

- Firewall mit NAT (network address translation)
- SSH (secure shell) Verbindungen
- Reverse tunnel Verbindung
- SSL (secure socket layer) Protokoll für https-Verbindungen
- SSL Client Zertifikate zur Authentifizierung
- Überwachunsprogramme

Die Beschreibung ist gegliedert in:

1 Remote-Verbindung

1.1 Datenserver

Der Datenserver wird mit dem open source Überwachungsprogramm "Samhain" ausgestattet, welches mittels Prüfsummen Änderungen an den Dateien des Betriebssystems überwacht und gegebenenfalls einen Alarm generiert, der auch per Email, oder SMS erfolgen kann. Es wird geprüft, ob es Sinn macht, das open source Programm "Fail2ban" zur Überwachung der Netzwerkschnittstellen, insbesondere des Ports 22 für den SSH-Zugang einzusetzen. Da ein "reverse tunnel" zum Einsatz kommt, würde "Fail2ban" möglicherweise keine zusätzliche Sicherheit bringen.

1.2 Router

Der Datenserver für die Piratendatei ist mittels eines Routers Fritz!Box 7490 über einen VDSL2-Anschluss mit 42 MB Upstream und 100 MB Downstream an das Internet angebunden. Die Firewall-Funktionen mit network address translation (NAT) stellen sicher, dass kein unautorisierter Zugriff auf den Server erfolgen kann. Ebenso stellt der deutsche Hersteller AVM des Routers durch seine kostenlose, transparente und zuverlässige Produktpflege mit regelmäßigen Firmware-Updates sicher, dass diese konform zum Bundesdatenschutzgesetz sind. Der Router ist sorgfältig mit restriktiven Einstellungen konfiguriert.

1.3 Verschlüsselte Verbindungen

Die Verbindungen mit dem Datenserver erfolgen mittels verschlüsselter Verbindungen unter Einsatz des Programms „secure shell“ (SSH). Das ist ein als sicher anerkanntes und vielfach eingesetztes Verfahren. Alternativ könnte auch „openVPN“ verwendet werden, welches den gleichen Sicherheitsstandard liefert. Dieses Programm ist jedoch konzeptionell dafür vorgesehen, ganze Netzwerksegmente miteinander zu verbinden. SSH ermöglicht prinzipiell eine Punkt-zu-Punkt-Verbindung, also PC zu Datenserver, sodass SSH hier der Vorzug gegeben wird, um möglichen Angriffsszenarien auf Netzwerke vorzubeugen.

Die Authentifizierung zum Verbindungsaufbau erfolgt ausschließlich über asymmetrische Verschlüsselung mittels privater und öffentlicher Schlüsseldateien. Nutzername und Kennwort werden grundsätzlich nicht akzeptiert. Somit sind selbst „brute-force“ Angriffe auf Kennwörter in jedem Fall erfolglos. Die „secure shell“ kann natürlich auch eine interaktive Oberfläche auf dem entfernten Rechner bereit stellen, auf der jener Rechner ferngesteuert werden kann. Da diese Funktion nicht benötigt wird, ist sie konsequent an allen Stellen für die Tunnel abgeschaltet.

Üblicherweise baut der Rechner, der sich mit dem Server verbinden möchte, den Datentunnel auf. In unserer Konfiguration ist es aber so, dass der Datenserver diesen Tunnel zu einem Gate-

way-Server aufbaut und dieser dann zur Verfügung steht, um Daten zu empfangen, also umgekehrter Verbindungsaufbau ("reverse"). Da sich der Datenserver hinter einer Firewall befindet, dessen IP-Adresse sich zudem wegen der DSL-Vorgaben auch noch täglich ändert, ist der "reverse tunnel" in diesem Falle das bevorzugte Verfahren, da der Datenserver naturbedingt einfach eine Verbindung ins Internet aufbauen kann. Außerdem steht die Verbindung vollständig unter Kontrolle des Datenservers. Bei Unstimmigkeiten bricht er den Datentunnel ab und mögliche Angreifer haben keine Möglichkeit mehr, ihn zu kompromittieren. In Richtung zum Datenserver ist kein Verbindungsaufbau möglich.

Auf die alternative Verwendung eines Providers für einen DDNS-Dienst (dynamic DNS) zur Ermittlung der ständig wechselnden Internet IP-Adresse des Routers als zusätzliche Vertrauensinstanz zum Stichwort „DNS spoofing“ möchte ich hier nicht weiter eingehen. Jedenfalls ist das daraus resultierende „man in the middle“ Angriffsszenario ein weiteres Argument für das in dieser Konfiguration gewählte Verfahren.

1.4 Gateway-Server

Bei einem Internet-Provider ist ein V-Server (virtueller Server) mit root-Zugang für jährlich 72 € angemietet und über den registrierten DNS-Namen "piratendatei.net" allgemein erreichbar. Dieser Server hat eine feste IP-Adresse und steht mitten im Internet. Er dient nebensächlich dazu, eine Demo-Version der Piratendatei mit unverfänglichen Testdaten allgemein zur Verfügung zu stellen, damit man mal "reinklicken" kann.

Hauptsächlich jedoch dient der Gateway-Server dazu, als Endpunkt des "reverse tunnels" zu dienen. Der Datenserver baut von sich aus einen Tunnel zum Gateway-Server auf. Der Gateway-Server stellt den Endpunkt des „reverse tunnels“ auf dem Port 22649 seines Internet-Anschlusses zur Verfügung. Dieser willkürliche Port bietet den Vorteil, dass er nicht von Angreifern gescannt wird, da es denen viel zu aufwändig ist, solche offenen Ports von 65535 möglichen zu suchen. Der Gateway-Server ist hinsichtlich von Angriffen sehr sorgfältig konfiguriert und gehärtet. Die firewall „iptables“ ist aktiv. Außerdem sind die Überwachungsprogramme „Samhain“ und „Fail2ban“ installiert. „Samhain“ überwacht mittels Prüfsummen Veränderungen an Betriebssystemdateien und generiert Alarme bei Änderungen. Die Alarme können auch per Email, oder SMS versendet werden. „Fail2ban“ überwacht die kritischen Ports und sperrt Zugriffe mittels der firewall „iptables“ bei brute-force Angriffen. Dieses ist besonders bei den täglich 1000-fachen scans an den Port 22 erforderlich. Dieser Port ist standardmäßig für den normalen SSH-Zugang festgelegt und wird auch für den Gateway-Server benötigt. Er dient ganz normal als remote shell, um den Gateway-Server zu administrieren. Er ermöglicht keine Verbindung zur Piratendatei.

Selbst wenn es einem Angreifer gelingt, Zugang zum Gateway-Server zu erhalten und dieses von den Überwachungsprogrammen unbemerkt bleibt, so findet er nichts relevantes. Es gibt keine Daten und es gibt nur öffentliche Schlüssel, da dieser Server keine Verbindungen aufbaut.

Der Gateway-Server leitet nun alle Daten, die er auf Port 22649, also piratendatei.net:22649, empfängt durch den „reverse tunnel“ an den Datenserver. Dieser akzeptiert aus diesem Kanal nur einen SSH-Tunnel mit Authentifizierung per Schlüsseldatei. Der entfernte PC baut also einen Datentunnel über den „reverse tunnel“ auf. Es findet demnach zwischen PC und Gateway-Server eine einfache Verschlüsselung statt, zwischen Gateway-Server und Datenserver eine doppelte Verschlüsselung. Über den Ende-zu-Ende Tunnel des PCs zum Datenserver (über den Gateway-Server) können nun Daten ausgetauscht werden.

Zudem können die web-Seiten von der Piratendatei an den Browser vom Nutzer-PC ohne Sicherheitseinbußen unverschlüsselt mittels http-Protokoll durch den Datentunnel zwischen PC und Datenserver gehen. Das ist Stand der Technik, wie sie auch für den Zugang zu SAGE der Bundes-IT realisiert ist.

2 Authentifizierung

Auf dem Datenserver ergibt sich nun die Problematik, dass er Daten benutzerspezifisch ausliefern soll, also dass er z.B. nur die Mitgliedsdaten aus einem Kreisverband ausliefert, wenn der Generalsekretär dieses Kreisverbandes die Anfrage stellt. Dazu muss zunächst ermittelt werden, wer angemeldet ist.

Auf dem Datenserver läuft „Apache“ als Web-Server und im Hintergrund dazu „postgres“ als Datenbankserver. Der Web-Server muss den Nutzernamen an den Datenbankserver übermitteln, damit dieser nur die relevanten Daten für die HTML-Seite an den Browser des Nutzers ausliefert. Der Nutzername lässt sich nun kaum sinnvoll aus den Verbindungsdaten der Datentunnel gewinnen und an den web-Server übermitteln.

Naheliegend ist, dass sich der Benutzer mittels Name und Kennwort im Browser beim web-Server authentifiziert. Das ist jetzt aber ein Standard-Verfahren, das nicht besonders hohen Sicherheitsanforderungen entspricht. Benutzername und Kennwort lassen sich z.B. sehr einfach an zweite Personen weitergeben. Deshalb wir hier die Authentifizierung mittels SSL Client-Zertifikate nach ITU-T X.509 gewählt. Das funktioniert als Verschlüsselungstechnik allerdings nur mit dem https-Protokoll, was ja aber kein Nachteil ist. Ein solches Zertifikat wird von der Verwaltung als Datei ausgestellt und es enthält den Namen des Benutzers. Auf diesem Wege ist eine gesicherte Authentifizierung gegeben. Der Benutzer importiert dazu das Zertifikat der CA (Certification Authority), seinen privaten Schlüssel und sein persönliches Zertifikat in den Browser. Für den Import muss er einmalig ein Kennwort eingeben und kann dann fortan mit diesem Browser ohne weitere Angaben seine Daten per https von der Piratendatei abrufen. Bei jeder Anfrage an den web-Server wird automatisch auch der Benutzername verschlüsselt an diesen übermittelt, der es dann an den Datenbankserver weitergeben kann. Das Verfahren selbst soll hier nicht Gegenstand der Beschreibung sein. Dazu gibt es zahlreiche Dokumentationen im Internet.

3 Autorisierung

Nachdem ein Benutzer eindeutig authentifiziert ist, muss nun entschieden werden, welche Daten er erhalten darf. Dieses geschieht ausschließlich im Datenbankserver mittels des dort vorhandenen Rollensystems. Jedem Nutzer werden entsprechende Rollen zugewiesen, z.B. Anwender (darf Daten lesen), oder Editor (darf Daten ändern). Die Berechtigungen können bis hinunter auf einzelne Felder vergeben werden, sodass z.B. Generalsekretäre und Schatzmeister zwar den gleichen Personal-Datensatz angezeigt bekommen können, der Schatzmeister aber nur mit einer eingeschränkten Zahl an Feldern. Die Pflege des Rollensystems ist naturbedingt ein dynamischer Prozess.

4 Kritik

4.1 Zertifizierung der X.509 Zertifikate

Vom prinzipiellen Verfahren her ist es vorgesehen, dass ein Nutzer seinen privaten Schlüssel für das X.509 Zertifikat selbst erstellt, daraus dann eine Zertifizierungsanforderung generiert und die Anforderung an die Zertifizierungsstelle (Verwaltung Piratenpartei) schickt. Die Zertifizierungsstelle unterschreibt die Zertifizierungsanforderung mit ihrer Zertifizierungsautorität und macht damit daraus das persönliche Zertifikat. Dieses schickt sie zurück an den Nutzer zur Verwendung.

Damit ist in aller Regel aber ein ungeübter Nutzer überfordert. Mir ist auch bis jetzt keine Organisation, oder Firma bekannt, die das so handhabt. Die organisatorische Lösung dieses Problems ist auch bei unserer Installation, dass die Verwaltung den privaten Schlüssel und die Zertifizierungsanforderung für den Nutzer stellvertretend erstellt und ihm dann gebrauchsfertig seinen privaten Schlüssel, das persönliche Zertifikat und das Zertifikat der CA übermittelt. Zwingende Vorschrift muss es dann aber für den Ersteller sein, dass er danach sofort mindestens den privaten Schlüssel des Nutzers sicher löscht, worauf er auch datenschutzrechtlich verpflichtet werden sollte.

4.2 Protokollierung von Verbindungsdaten

Sowohl auf dem Gateway-Server für die Demo-Installation, als auch für die Piratendatei auf dem Datenserver werden die Verbindungsdaten in der Standardkonfiguration von „Apache“ erfasst und gespeichert. Das sind im Wesentlichen IP-Adresse, Browser des Clients, Referer (vorher besuchte Website), Datum, Uhrzeit und ausgeführte Transaktion. Es ist noch festzulegen, in welchem Umfang und für welche Dauer diese Daten gespeichert werden sollen. Dabei soll abgewogen werden zwischen der Erfordernis zur Nachvollziehbarkeit von Angriffen, oder unautorisierten Zugriffen und der Verhinderung der Vorratsdatenspeicherung.

Ähnliche Daten werden auf dem Gateway-Server auch für die Zugriffe auf den SSH-Port 22 protokolliert. Dieses ist jedoch unkritisch. Zum einen werden diese Protokolle zur Abwehr von Angriffen durch „fail2ban“ ausgewertet, zum anderen ist die Verbindung über diesen Port nur einem ganz eng begrenzten Personenkreis von Administratoren erlaubt.

4.3 Nur https verwenden

Zunächst ist es das Ziel, die Anforderungen der Bundes-IT für eine Remote-Verbindung auch für die Piratendatei nachzubilden. Mit einer Tunnelverbindung vom Client zum Server, über welche mindestens per http kommuniziert wird, ist dieses gegeben. Dass dabei eine weitere Verschlüsselung im „reverse tunnel“ erfolgt, ist lediglich systembedingt.

Nun hat sich aber durch den Einsatz der X.509 Zertifikate quasi als Nebenbedingung ergeben, dass eine weitere Verschlüsselungsebene eingeführt ist. Aus Sicht der Datensicherheit bringt dieses so gut wie keinen zusätzlichen Gewinn. Allerdings erhöht es die Komplexität signifikant. Komplexe Systeme sind anfälliger und schwerer zu handhaben, vor allem für ungeübte Nutzer, sodass auch hier Unsicherheiten entstehen können.

Es ist deshalb zu überlegen, ob die Tunnelverbindung zwischen Client und Datenserver (über den Gateway-Server) entfallen kann und die Kommunikation allein per https erfolgt. Immerhin ist das ein Verfahren, das auch sämtliche Banken für all ihre Transaktionen als sicher genug ansehen.

11. September 2015, Ingo Höft

5 Übersicht

Remote-Zugang zur Piratendatei