testing experience de

Das Magazin für Tester, Entwickler und Manager

Nr. 2 | Juli 2013
kostenlose digitale Version

Der Weg zum agilen Testen

Díaz Hilterscheid

Vulnerability-Check von Internetroutern

Sicherheitsrisiko durch NAS-Funktionalitäten

Die Autoren

Valeri Milke

Nach abgeschlossenem B. Sc. Informatikstudium mit dem Schwerpunkt der Informationssicherheit arbeitet Valeri Milke bei der softScheck GmbH als IT Security Consultant. Kernkompetenzen sind neben klassischem Penetration Testing und Vulnerability Analysis auch Fuzzing als eine semi-automatisierte Methode für Softwaretests. Hauptanwendungsgebiete sind Webanwendungen, SCADA-Security und Mobile Security.

Prof. Dr. Hartmut Pohl

Professor für Informationssicherheit. Geschäftsführender Gesellschafter der IT-Sicherheitsberatung softScheck GmbH, Köln. Informationssicherheit mit dem Schwerpunkt ‚Security Testing': Erfolgreiche Identifizierung insbesondere bisher nicht erkannter Sicherheitslücken (Zero-Day Vulnerabilities) in Software (und Hardware/Firmware) mit den folgenden 4 Verfahren: Threat Modeling, Static Source, Code Analysis, Dynamic Analysis: Fuzzing, Penetration Testing. Erfahrungsgemäß werden bei Einsatz dieser 4 Verfahren mit jeweils mehreren (!) Tools alle Sicherheitslücken – auch alle Zero-Day-Vulnerabilities – identifiziert.

Viele Hersteller für WLAN-Router implementieren in ihren Produkten Network Attached Storage (NAS) Funktionalitäten, die – verbunden mit einer externen Festplatte – Netzwerkspeicher ersetzen können. In den Default-Konfigurationen sind NAS-Funktionalitäten zwar meist deaktiviert, lassen sich aber bei Bedarf einschalten; in diesen implementierten Funktionalitäten können Sicherheitslücken enthalten sein.

Network Attached Storage (NAS) stellt angeschlossenen Clients betriebssystemunabhängig Dateisysteme zur Verfügung. Das NAS kann dabei so verwendet werden, als würde sich die Festplatte direkt auf dem eigenen Rechner befinden. Hierfür wird die SMB-Schnittstelle (Server Message Block) verwendet, die die Open Source Suite „Samba" betriebssystemunabhängig implementieren kann. SMB ist ein auf NetBIOS-basierendes Netzwerkprotokoll mit den Kernfunktionalitäten Dateiübertragung und Kommunikation mit dem Drucker. NetBIOS wiederum basiert auf TCP und nutzt in der Default-Konfiguration den Port 139. Die SMB-Schnittstelle wird in der Regel durch die integrierte Firewall des Routers nach außen gesperrt, sodass nur die Clients Zugriff auf das NAS haben, die sich im lokalen Netz befinden. 1996 wurde das SMB-Protokoll von Microsoft unter der Bezeichnung Common Internet File System (CIFS) weiterentwickelt. Neue Features waren z.B. die Unterstützung von größeren Dateien, Soft- und Hardlinks. Außerdem wurde eine neue direkte SMB-Verbindung ohne NetBIOS als darunterliegendes Kommunikationsprotokoll implementiert, die den TCP Port 445 nutzt.

Viele Router ermöglichen zudem die Nutzung des NAS über das Internet. Es entstehen so neue Nutzungsszenarien, jedoch auf Kosten der Sicherheit. Die integrierte Firewall öffnet für diesen Zweck weitere Ports, z. B. FTP (TCP Port 21) über die der Angreifer in das System gelangen kann und im Worst Case die Daten lesen und manipulieren kann. Das Risiko eines erfolgreichen Angriffs ist besonders dann hoch, wenn die FTP-Kommunikation inkl. Authentifizierung in Klartext übertragen wird. Die Log-in-Daten können dann durch Man-in-the-middle-Angriffe (MITM-Angriffe) mitgeschnitten werden, wie in Abbildung 1 zu sehen.

In diesem Beispiel wurde mit dem Tool „Cain & Abel" ein MITM-Angriff durchgeführt. Cain & Abel sendet dabei gefälschte ARP-Pakete, um die Kommunikation zwischen beteiligten Servern und Clients mit zu verfolgen. Diese Art von MITM-Angriffen wird ARP-Spoofing genannt und wird z.B. auch verwendet, um eine VoIP-Kommunikation mitzuschneiden. ARP befindet sich im OSI-Modell zwischen TCP und Ethernet und dient dazu anhand einer IP die MAC-Adresse des Netzwerkadapters herauszufinden, daher können alle Kommunikationspakete, die sich über dieser Schicht befinden, durch ARP-Spoofing mitgeschnitten werden.

Es wird daher empfohlen, bei Nutzung des NAS über das Internet Verschlüsselung einzusetzen, das Risiko wird dadurch signifikant gesenkt. Um eine FTP-Kommunikation zu verschlüsseln gibt es mehrere Alternativen.

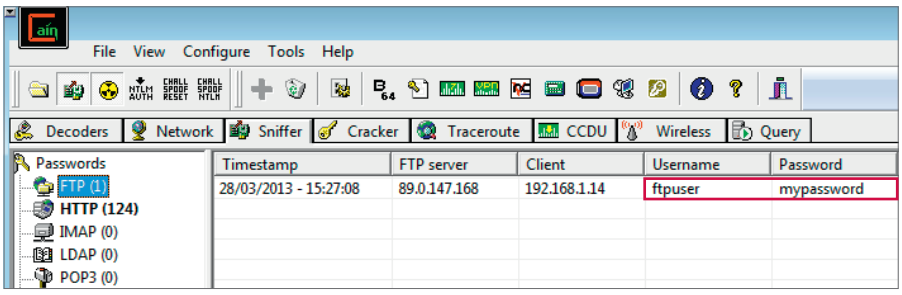

Abbildung 1

FTPS als Alternative nutzt in Kombination eine SSL bzw. TLS-Verschlüsselung. Diese kann explizit (auch bekannt als FTPES) oder implizit sein. Es wird eine implizite Verschlüsselung empfohlen, da hierbei alle Anfrage ohne SSL-/TLS-Verschlüsselung vom FTP-Server (oder des Routers als FTP-Server) abgelehnt werden und somit Log-in-Daten durch MITM-Angriffe nicht mitgeschnitten werden können. Verschlüsselungen nützen nichts, wenn der Angreifer über Library- oder Bruteforce-Angriffe die Log-in-Daten errät. Bei der Erstkonfiguration sollten daher unbedingt evtl. vorhandene Default-User-Accounts wie „admin“ und „ftpuser“ deaktiviert werden.

Die zweite Alternative für eine sichere FTP-Kommunikation ist das SSH File Transfer Protocol (SFTP). SFTP ist eine Erweiterung der Secure Shell (SSH) um Datenübertragungsfunktionen und ist seit der aktuellen Version SSH-2 fester Bestandteil dieses Kommunikationsprotokolls. Mit dem Open Source Projekt OpenSSH lässt sich SFTP implementieren. In der Default-Konfiguration nutzt SFTP den TCP Port 22. Eine Änderung des Ports für SSH erhöht die Sicherheit des Systems, da der Angreifer für das Protokoll SSH den TCP Port 22 erwartet und damit zielgerichteter angreifen kann.

Zusätzlich zur verschlüsselten Dateiübertragung, lässt sich die Sicherheit weiterhin erhöhen, wenn ergänzend eine VPN-Verbindung eingesetzt wird, sodass das NAS über das Internet zunächst gar nicht erreichbar ist. Ein sicherer VPN-Tunnel alias SSL-VPN lässt sich via SSL- oder TLS-Protokoll umsetzen. Einige Router bieten bereits integrierte VPN-Lösungen an.

Abschließend ist zu bemerken, dass alle Sicherheitsmechanismen die Wahrscheinlichkeit für einen erfolgreichen Angriff nur verringern können. Das Restrisiko mit Library- und Brute-Force-Angriffen in das System einzudringen bleibt – der Aufwand ist allerdings ungleich höher. Das höchstmögliche Sicherheitsniveau lässt sich nur erreichen, wenn die Angriffsfläche reduziert wird, offene und überflüssige Ports also geschlossen werden.

Literaturverzeichnis

- Ford-Hutchinson, P. (Oktober 2005). *Securing FTP with TLS*. Abgerufen am 15. April 2013 von *http://tools.ietf.org/html/rfc4217*
- J. Galbraith, O. (10. Juli 2006). *SSH File Transfer Protocol*. Abgerufen am 15. April 2013 von *http://tools.ietf.org/html/draft-ietf-secsh-filexfer-13*
- OpenBSD Foundation. (22. März 2013). *OpenSSH*. Abgerufen am 15. April 2013 von *http://www.openssh.org/de/*
- Sons, J. W. (2012). *Information Sorage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments*. EMC².