A. Ruckelshausen et al. (Hrsg.): Digitale Marktplätze und Plattformen,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 91

d.fence - Sicherer Fernzugriff auf Agrarsteuerungen mit Hilfe eines SSH-Reverse-Proxys mit grafischer Bedien- und Verwaltungsoberfläche

Stefan Hagedorn[1]

Abstract: Für den Betreiber von Anlagen der Mess-, Steuer- und Regelungstechnik bietet sich mit dem System d.fence die Möglichkeit, sichere und robuste Fernzugänge als Standard für seine gesamte Produktpalette zu implementieren, wo er heute noch eine heterogene Landschaften vorfindet, welche maßgeblich von seinen Endkunden oder deren teils semiprofessionellen IT-Beauftragten mitgestaltet wird.

Keywords: Datensicherheit, Datenschutz, Remote Services, Diagnose, Internet der Dinge und Mobile Vernetzung

1 Ausgangslage - die "Pain-Points"

Die Produktionsprozesse in der Landwirtschaft finden oft an verschiedenen Orten statt. Digitalisierung heißt hier die Produktionsketten an verschiedenen Orten zu vernetzen und hierdurch zu integrieren und zu steuern. Oft mangelt es jedoch bei den Betreibern von Infrastruktur an der Kenntnis und Sensibilität für dieses Thema. In der Folge werden Visualisierungen, Datenaustauschverfahren und Datenkopplungen oft durch ungesicherte Verbindungen, das Öffnen von Hintertüren oder Portweiterleitungen bewerkstelligt. In Kombination mit leeren oder unveränderten Passworten für Administrationsoberflächen ergibt sich ein ungeschützter Zugang zu diesen Geräten. Mit Hilfe der IoT-Suchmaschine shodan.io\cite{shodan} lassen sich Zugänge zu diesen Geräten aufspüren und ausbeuten.

Weiterhin gilt gerade für Steuerungsrechner, dass Betriebssysteme für Prozessrechner häufig auf den Anwendungsfall zugeschnitten sind und somit Sicherheits-Patches manuell eingepflegt werden müssen. Darauf folgt ein aufwändiger Test des Prozessrechners auf uneingeschränkte Funktionalität und im weiteren Verlauf das Deployment, teils in persönlicher Abstimmung mit dem Endkunden um Produktionsausfälle zu vermeiden. Diese Aufwände führen dazu, dass die Hersteller von Prozessrechnern das Einpflegen von Patches des Betriebssystems oft als Aufwand wahrnehmen, den es zu vermeiden gilt.

[1] Hagedorn Software Engineering GmbH, Hesselstraße 2, 48231 Warendorf,
info@hagedorn-software.de

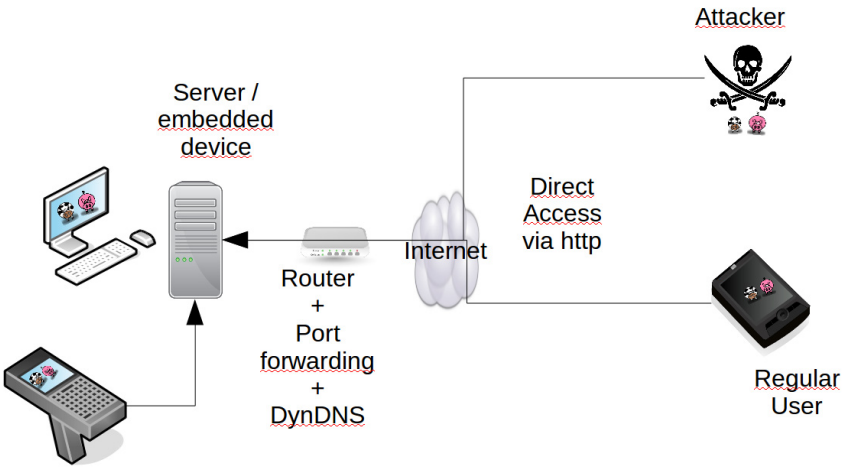

Abb. 1: Typische - unsichere - Vorgehensweise

2 Anforderungen für eine Lösung

Software und Geräte, welche auf den Austausch oder die Visualisierung von Daten angewiesen sind, müssen konsequent werksseitig mit starker Verschlüsselung mit Hilfe von vertrauenswürdigen Zertifikaten bzw. einer Private/Public Key Infrastructure sowie einem handhabbaren Mechanismus für den Verbindungsaufbau und Abbau ausgestattet sein. Die Lösung muss aus frei verfügbaren (Open-Source) Komponenten besteht, so dass gewährleistet ist, dass keine versteckten Hintertüren oder Sicherheitslücken in der verwendeten Software eingebaut sind\cite{patriotact}. Die verwendeten Komponenten sollten ein kostenfreies Verteilen ermöglichen. Der Verbindungsaufbau muss vom Prozessrechner zu einem Vertrauenswürdigen System im Internet erfolgen, anstatt einer Exposition eines Serverprogrammes in das Internet hinein. Die Betreuung der Infrastruktur muss in den Händen der Hersteller von Software und Geräten selbst liegen.

3 Lösungsansatz

3.1 Vorgehensweise auf dem Server

Auf dem Server legt der Bediener - meistens der Mitarbeiter im Service oder der Produktion - das Gerät an und ordnet es einem seiner Kundenbetriebe zu. Hierdurch können in der Folge die Nutzer des betreffenden Kundenbetriebes sowie die Service-Mitarbeiter des

Herstellers das Gerät verwalten. Dann trägt entweder der Service-Mitarbeiter oder der Endnutzer den Public Key des Gerätes mit Hilfe der grafischen Benutzeroberfläche ein. Die Vorgehensweise ist angelehnt an die Vorgehensweise der Einrichtung eines Nutzers z.B. bei GitHub.\cite{github}, GitLab\cite{gitlab} und ähnlichen, auf SSH basierten Diensten. In gleicher Weise können sowohl Nutzer als auch Servicemitarbeiter das Gerät sperren, löschen oder den Besitzer ändern, sofern es letzteren durch Weiterverkauf gewechselt hat. Der Server verteilt die aufgebauten verschlüsselten Verbindungen so, dass die Nutzer, die die für sie freigeschalteten Geräte sehen wollen, über die Verbindung zum Server auf ihr Endgerät geleitet werden.

3.2 Vorgehensweise auf dem Prozessrechner

Für die verschlüsselte Verbindung zum d.fence-System ist es erforderlich, dass ein Private/Public Key-Paar auf dem Prozessrechner existiert. Der Public-Key muss auf das System d.fence übertragen werden. Ideal hierfür ist es, wenn die Weboberfläche des Gerätes diesen Public Key in seiner Weboberfläche präsentiert, so dass der Nutzer des d.fence-Systems (Endnutzer oder Service-Mitarbeiter) diesen per copy&paste im System d.fence bei dem entsprechenden Gerät eintragen kann.

3.3 Vorgehensweise auf dem Bediengerät

Auf dem Bediengerät ist als Voraussetzung lediglich ein Browser erforderlich. Der Nutzer ruft die Adresse des Gerätes innerhalb des Systems d.fence - also im Webspace auf. Nach erfolgreicher Autorisierung bedient der Nutzer sein System, wie er es von der lokalen Bedienung gewohnt ist.

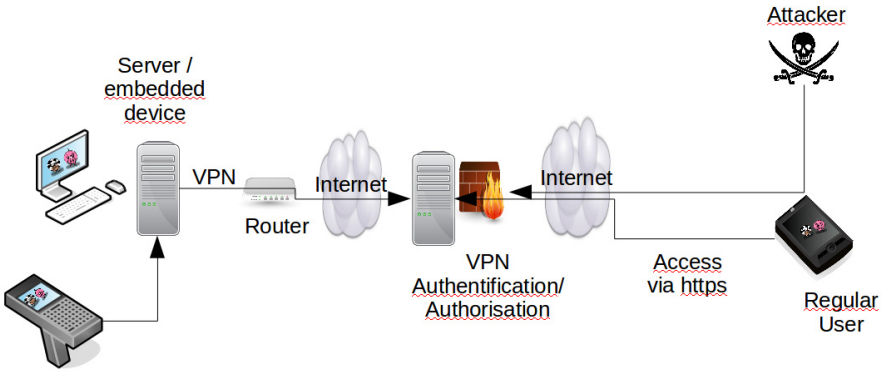

Abb. 2: Sichere Vorgehensweise durch SSH-Reverse-Proxy

4 Alternativen

Im Bereich von PC-Software und der Applikationen für Android/IoS bestehen bereits seit längerer Zeit Produkte, welche sich mit einer verschlüsselten Möglichkeit der Fernwartung beschäftigen. Anbieter solcher Software sind z.B. die Firma TeamViewer \cite{teamviewer} oder ähnliche. Für den Privatanwender sind diese Werkzeuge kostenlos, im geschäftlichen Umfeld dagegen kostenpflichtig. Hierdurch hat sich eine Grauzone verbreitet, in der gewerbliche Unternehmen private Zugänge nutzen, obwohl dies nicht zulässig ist. Diese Werkzeuge sind auf die Visualisierung von Bedienoberflächen zugeschnitten, die auf einem Monitor angezeigt werden. Fernwartungssysteme für embedded Geräte ohne Monitor inklusive einer Verwaltung für den Gerätehersteller gibt es derzeit nicht am Markt. Verfügbar sind verschiedene Verschlüsselungssoftware wie OpenSSL\cite{openssl} oder das darauf aufsetzende OpenSSH\cite{openssh}. Diese sind jedoch durch Konfigurationsdateien auf Kommandozeilenebene zu konfigurieren, was nur wenigen Experten vorbehalten bleibt.

5 Zusammenfassung

Das hier gezeigte System d.fence ist ein Good-Practice-Fall dafür, bekannte Softwaretechniken neu zu verbinden und mit überschaubarem Aufwand ein Produkt zu komponieren, welches alltägliche Probleme im Bereich Service und Support löst. Es soll den Leser ermutigen, die "`Pain-Points"' des alltäglichen Umgangs mit Elektronik und EDV kreativ durch neue Komposition einfacher und vorhandener Techniken zu lösen.

Literaturverzeichnis

[gita] Github. https://github.com 21.11.2017

[gitb] Gitlab. https://gitlab.com 21.11.2017

[myf] AVM MyFritz. https://www.myfritz.net/ 21.11.2017

[opea] OpenSSH. https://www.openssh.com/ 21.11.2017

[opeb] Openssl.- Cryptography and SSL/TLS Toolkit https://www.openssl.org/ 21.11.2017

[pat] PatriotAct. https://de.wikipedia.org/wiki/usa-patriot-act 21.11.2017

[sho] shodan - The search engine for the Internet of Things. https://shodan.io 21.11.2017

[tea] Teamviewer. http://www.teamviewer.com 21.11.2017