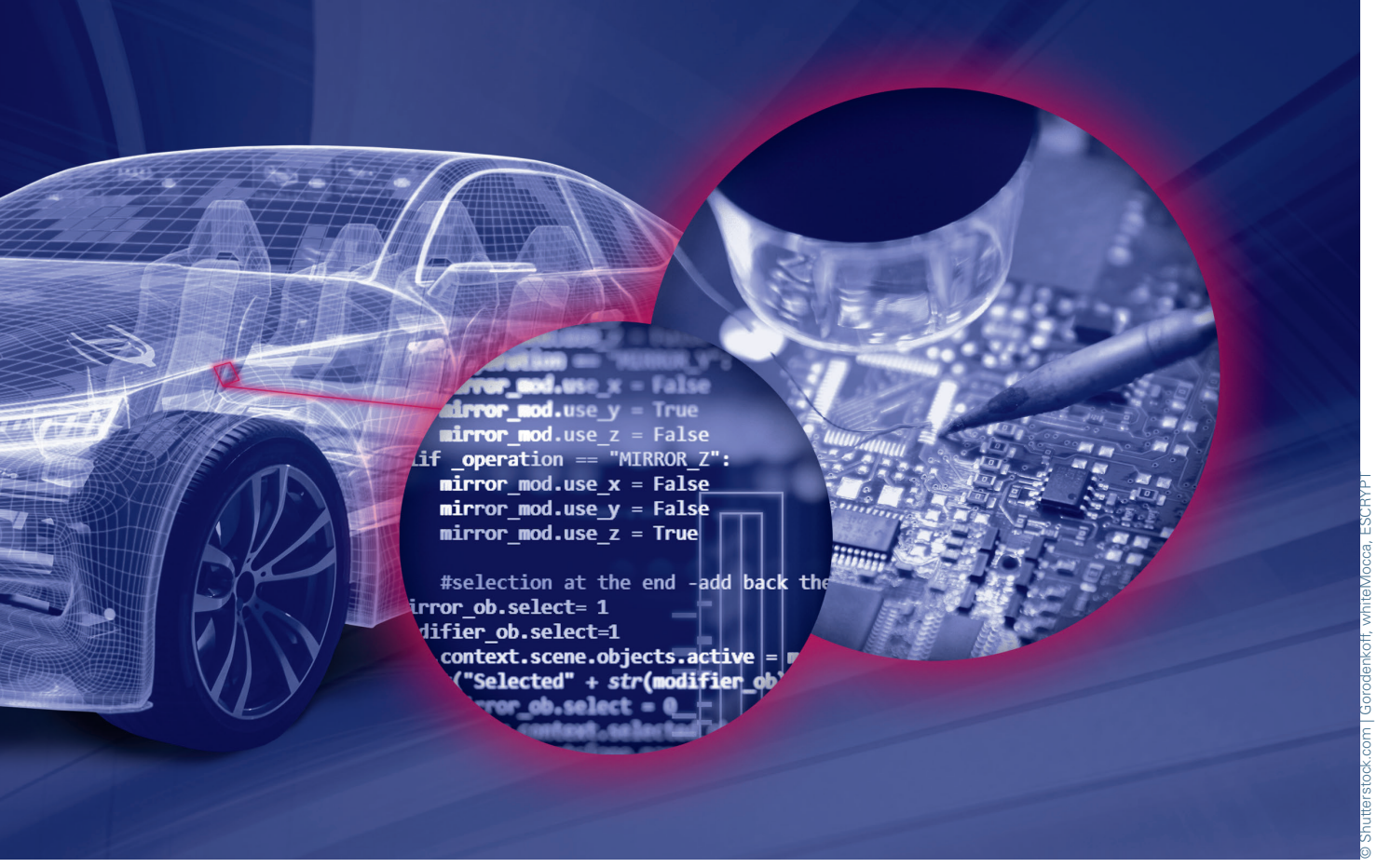

PENETRATION TESTING

Die Cybersecurity-Probe aufs **Exempel**

Vernetzte, hochautomatisierte Mobilität setzt die Messlatte für die IT-Sicherheit im Fahrzeug hoch. Penetrationstests für fahrzeuginterne Netzwerke oder Steuergeräte gelten hier als ultimative Eignungsprüfung. Doch wann und wie werden sie durchgeführt? Und was sind typische Ergebnisse solcher Automotive Pentests?

Mit zunehmender Konnektivität und Automatisierung der Fahrzeuge sind IT-Sicherheitsfunktionen für OEMs und Zulieferern längst unverzichtbarer Bestandteil des internen Fahrzeugnetzes und seiner Komponenten. Doch wie sicher sind Fahrzeugsteuergerät und -netzwerk am Ende wirklich vor unerlaubten Zugriffen und Manipulation? Eine der gängigsten und wirkungsvollsten Methoden, das herauszufinden, ist das Penetration Testing, kurz Pentesting. Der Pentester simuliert Cyberangriffe auf das System – und findet mögliche Sicherheitslücken bevor es ein anderer tut.

Wirklichkeitsgerechte Testumgebung

Ziel ist es also herauszufinden, ob und wie etwaige Angreifer Zugriff auf Systemfunktionen oder Daten erlangen können, um die identifizierten Schwachstellen in der Folge zu schließen. Typische Testfälle fürs Pentesting im Automotive-Bereich sind das Testen einzelner Steuergeräte, das Testen mehrerer ECUs im Verbund oder gar vollständiger Fahrzeugplattformen.

Wichtig dabei: Auch beim Security-Pentest eines einzelnen Steuergerätes muss dieses sinnvollerweise in eine Umgebung eingebunden sein, die den Betrieb des Testsystems ermöglicht. Das kann etwa eine angeschlossene Restbus-Simulation sein, die dem Steuergerät die nötigen Signale zur Verfügung stellt. Das können aber auch andere Steuergeräte oder spezielle Testeinheiten mit einzelnen Sensoren und Aktoren sein, mit denen sich das Steuergerät in einen wirklichkeitsgerechten Betriebszustand versetzen lässt. In Einzelfällen brauchen Steuergeräte auch eine reale Umgebung, beispielsweise die Anbindung an echte Lenk- oder Differentialgetriebe, um nicht im Fehlerzustand zu verharren und ein zielführendes Security Testing zu ermöglichen.

Pentesting als Nagelprobe

Für die Identifikation von Schwachstellen mittels Security-Tests unterscheidet man grundsätzlich zwei Vorgehensweisen: Das automatisierte softwarebasierte Testen, bei dem dedizierte Testing-Tools Verwundbarkeiten im System aufspüren. Und die manuelle Herangehensweise, bei dem ein Security-Experte eigenhändig versucht, mögliche Schwachstellen aufzuspüren und ins System einzudringen.

So lassen sich automatisierte Tests – etwa das Security-Testing-in the-Loop (Security XiL) – als feste, beliebig oft wiederholbare Routine hervorragend in den Entwicklungsprozess eines Systems integrieren, um bekannte Schwachstellen möglichst frühzeitig zu identifizieren und abzustellen.

Um jedoch auch die Sicherheitslücken zu finden, die nicht im Netz automatisierter Testverfahren hängen bleiben, sind Pentests wichtiger Bestandteil einer umsichtigen Security-Strategie. Penetrationstests setzen daher häufig zu einem späteren Zeitpunkt im Entwicklungsprozess an, wenn mindestens ein Teil der für den Test wesentlichen Funktionen vollständig ist. Im Gegensatz zu den rein automatisierten Testverfahren beinhalten sie eine eingehende individuelle Prüfung des Systems durch einen Security-Experten unter den erwähnten wirklichkeitsnahen Bedingungen. Nicht selten sind die Ergebnisse solcher Penetrationstests ausschlaggebend für die Marktfähigkeit und -zulassung des Systems in punkto IT-Sicherheit.

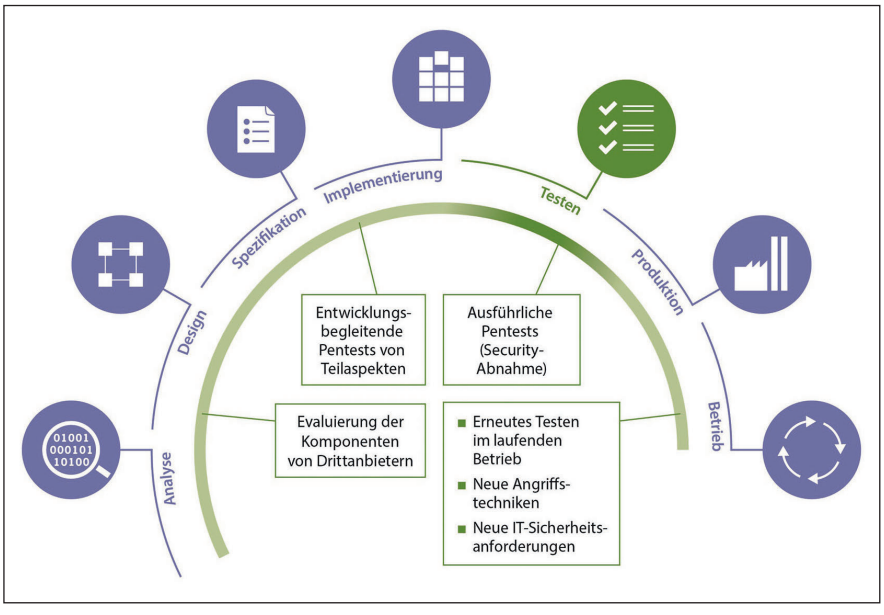

Bild 1: Penetrationstests werden häufig gegen Ende des Entwicklungsprozesses durchgeführt, sind aber anlassgebunden auch in anderen Phasen des Produktlebenszyklus sinnvoll. © Escrypt

Allerdings können Penetrationstests mitunter selbst nach Start-of-Production sinnvoll und notwendig sein – etwa wenn sich im Feld unerwartet Sicherheitslücken auftun, erforderliche Tests versäumt wurden oder wenn neue Angriffstechniken bekannt werden (Bild 1).

Umfassende Testing-Expertise erforderlich

Natürlich setzt auch der Pentester zunächst automatisierte Testing-Tools ein, dies jedoch in vollem Wissen um die Möglichkeiten und Grenzen solcher Tools. Denn so schnell automatisierte Testing-Tools bekannte Schwachstellen in den geläufigen Services und Protokollen finden, so „blind" sind sie oft gegenüber logischen Fehlern oder individuellen kundenspezifischen Anpassungen. Gerade im Automotive-Bereich aber

Bild 2: Für das Automotive Pentesting stehen eine Vielzahl von Techniken und Testmethoden zur Verfügung. Der Penetration-Tester entscheidet, auf welche er im Einzelfall zurückgreift. © Escrypt

sind solch spezifische Adaptionen wegen der limitierten Ressourcen und speziellen Anwendungsfälle der Systeme keine Seltenheit. Hier zahlt sich der Rückgriff auf Testing-Tools aus, die individuell auf die Spezifikationen der Fahrzeugplattformen bestimmter OEMs oder gar auf einzelne Steuergeräte angepasst werden können. So verwenden die Security-Tester bei ESCRYPT beispielsweise spezifisch auf die Anforderungen bestimmter Hersteller ausgelegte Fuzz-Testing-Tools und gelangen auf diese Weise sehr viel schneller zu sehr viel genaueren Testresultaten.

Dem Penetration-Tester indes liefern solch automatisierte Tools, so präzise sie auch arbeiten, nicht das Endergebnis, sondern vielmehr die Ausgangsbasis seiner eigentlichen Arbeit. Denn erst der Pentester als Security-Experte ist in der Lage, die Auffälligkeiten, die ihm das Testing-Tool aufzeigt, im Detail zu analysieren, indem er die zugrundeliegenden Daten, Datenverkehr, Prozesse oder den Quellcode eingehend untersucht. Auch kann er andere Experten konsultieren oder Datenbanken nach geeigneten Angriffen auf ähnliche Ziele durchforsten. Und er kann mehrere Fehler, die einzeln harmlos erscheinen, zu komplexen und potenziell gefährlichen Angriffsketten kombinieren. So wertvoll der Einsatz geeigneter Testing-Tools auch ist – letztlich liefert erst das manuelle Pentesting durch die Security Experten ein umfassendes Bild der IT-Sicherheit des zu testenden Systems im Falle einer gezielten elaborierten Cyberattacke.

Whitebox: Je mehr Informationen desto effizienter der Test

Grundsätzlich also bedient sich das Pentesting im Automotive-Bereich aller verfügbaren Techniken und Testmethoden. Je nach zu testendem System und Aufgabenstellung greift das Penetration Testing auf Fuzzing-Tests, Vulnerability Scans oder Seitenkanalanalysen zurück, kann sowohl ein Code Audit als auch den Test des physischen Systems und seiner Security-Funktionen beinhalten. Am Anfang steht jedoch stets die gemeinsame Risikoanalyse des Zielsystems durch den Kunden und das Testing-Team. Eine Headunit mit zahlreichen unterschiedlichen Schnittstellen bis hin zu WiFi und Bluetooth verfügt über mehr und andere Angriffsvektoren als ein tief verbautes Steuergerät mit einzelnem CAN-Bus (Bild 2).

Grundsätzlich gilt: Je mehr der Pentester über das System weiß, desto zielführender und effektiver kann er dessen Cybersicherheitslevel testen, Schwachstellen ermitteln und Verbesserungsvorschläge unterbreiten. Vorzugsweise werden Pentests daher als Whitebox-Tests durchgeführt, d. h. der Auftraggeber liefert möglichst alle verfügbaren Informationen: Quellcode, Rest-Bus-Simulation, vollständige Dokumentation (auch der vernetzten Security-Funktionen), Entwicklungs- und Diagnosetools, Zugänge zum Backend etc..
Blackbox- oder Greybox-Tests, bei denen diese Informationen nicht oder nur teilweise zur Verfügung stehen, sind deutlich ineffizienter und kostenintensiver, da viel Aufwand ins Reverse Engineering fließt, d.h. der Pentester verbringt viel Zeit damit, Dinge herauszufinden, die auf Seiten des Auftraggebers schon bekannt sind. Blackbox- oder Greybox-Tests sind nur dann zu empfehlen, wenn dem Tester, beispielsweise aus juristischen Gründen, die Informationen nicht zugänglich gemacht werden dürfen oder etwa der Quellcode dritter Zulieferer nicht zur Verfügung steht.

Iterative Herangehensweise

Demnach ist Pentesting zwar immer ein individuelles Geschehen je nach zu testendem System und Kundenanforderung. Es folgt aber natürlich dennoch einem festen und zielführenden Ablauf – angefangen bei der Inbetriebnahme des Systems sowie einer eingehenden Betrachtung aller Systemkomponenten, Schnittstellen, Funktionalitäten, Diagnosefunktionen etc. und der Erstellung eines ersten Testplans. Im folgenden Schritt fördern Fuzztests, ein Vulnerability Scan oder auch die genaue Beobachtung der Systemkommunikation erste Auffälligkeiten und mögliche Angriffspunkte zutage. Diese potentiellen Schwachstellen wird der Pentester dann genauer untersuchen und – im Stile eines Angreifers – versuchen sie auszunutzen, um ins System einzudringen oder es zu manipulieren (Bild 3).

Penetrationstests bedienen sich also einer iterativen Herangehensweise: Mögliche Angriffspunkte werden identifiziert, bestehende Schwachstellen ausfindig gemacht und ausgelotet. Daraufhin wird der Kunde vom Testing-Team mit den bestehenden Sicherheitslücken und den damit verbundenen Risiken konfrontiert sowie hinsichtlich wirksamer Lösungen und Gegenmaßnahmen beraten. Das kann von kleinen Implementierungsfehlern und der Änderung einzelner Codezeilen bis hin zu massiven IT-Sicherheitsmängeln und einer vollständigen Neukonzeption des getesteten Systems reichen. Im Regelfall werden die bestehenden Lücken in der Folge geschlossen und durch Re-Tests die erfolgreiche Absicherung des Systems bestätigt bzw. weitere nötige und mögliche Security-Maßnahmen anempfohlen. All das so lange, bis der vereinbarte Testumfang abgearbeitet ist.

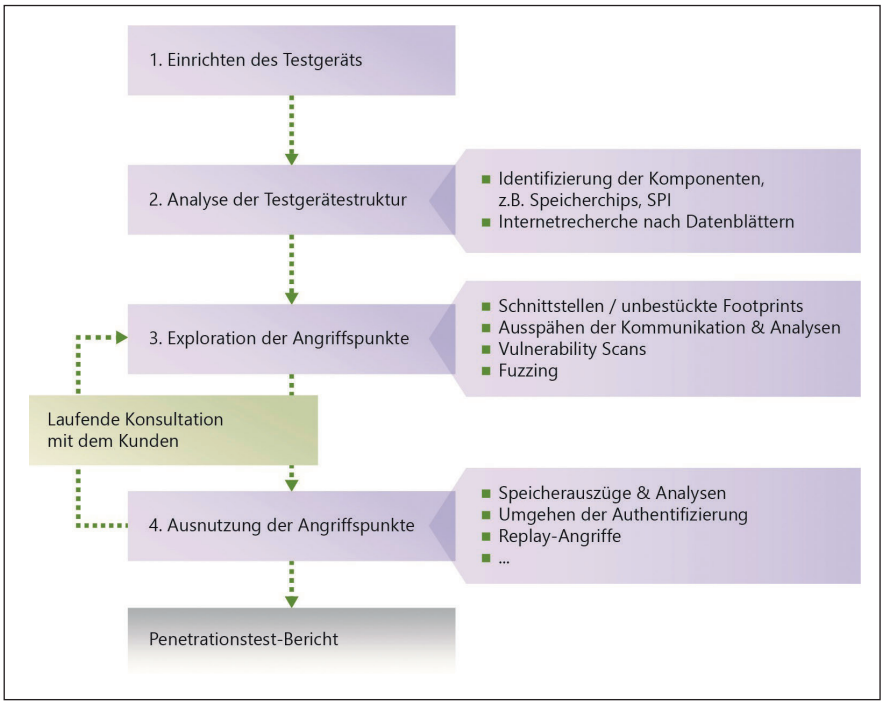

Bild 3: Operationaler Ablauf eines Automotive Pentests. © Escrypt

Typische Pentest-Findings

Das Pentesting von Fahrzeugsteuergeräten, Infotainmentsystemen oder auch mehrerer ECUs im Verbund folgt zwar immer einem dem Zielsystem gemäßen individuellem Vorgehen. Nichtsdestotrotz gibt es natürlich typische Angriffspunkte und Findings, die ein erfahrener Automotive Pentester stets in den Blick nimmt und – sofern vorhanden – gezielt überprüft. Solch typische Findings sind unter anderem:

- Offener Security Access: Über die Fahrzeugdiagnose ist ein Entwicklungsmodus zugänglich, der als „Hintertür" im finalen Produkt verblieben ist.
- Offener Hardware-Debugging-Port: Ein nicht gesperrter Debugging-Port (z. B. JTAG) ermöglicht den Lese-/Schreibzugriff auf den Speicher des Steuergeräts sowie die Analyse und Modifizierung der ausgeführten Programme.
- Mangelnde Robustheit / veraltete Software: Unzureichende Robustheit oder fehlende Einhaltung von Coding Standards (z. B. MISRA-C, CERT-C) kann zu schwerwiegenden Fehlern führen. Wenn beispielsweise beim Empfang eines fehlerhaften CAN-Frames das Zielsystem komplett abstürzt, kann ein Angreifer den Fehler näher untersuchen und das zu Grunde liegende Problem (beispielsweise Buffer Overflow) für einen Angriff ausnutzen. Dazu kommt, dass eingebundene Softwaremodule von Drittanbietern zwar über die Zeit Korrekturen erhalten haben, diese aber im vorliegenden Gerät nicht eingespielt wurden. Ein gutes Beispiel ist der Heartbleed Bug in der OpenSSL-Kryptobibliothek bei Versionen älter als 1.0.1f.
- Fehlerhafte Security-Funktionen: Die Security-Mechanismen sind nicht korrekt implementiert oder konfiguriert, z. B. liefert eine Zertifikatsprüfung wegen eines logischen Fehlers bei der Implementierung immer nur „True" zurück oder eine Firewall lässt sämtlichen Traffic passieren.
- Fehlerhafter Zufallsgenerator: Durch einen fehlerhaften Zufallsgenerator (RNG) wird die Zugriffskontrolle beeinträchtigt, und ein Angreifer könnte aufgezeichnete Authentisierungsdurchgänge wiederverwenden.
- GPS-Zeit zur Validierung von Zertifikaten: Durch Senden eines falschen GPS-Signals und Manipulieren der GPS-Zeit auf ein bestimmtes Datum werden Zertifikate ungültig, was im Zweifelsfall eine dauerhafte Funktionseinschränkung bewirkt.
- USB-Unterstützung: Das System unterstützt USB-Funktionalitäten, die nicht benötigt werden, wodurch die Angriffsfläche unnötig vergrößert wird.
- Bluetooth-Probleme: Aufgrund der historisch gewachsenen Komplexität des Bluetooth-Protokolls gibt es viele Angriffspunkte.
- Verwendung ungesicherter Protokolle
- Mobilfunkverbindungen können angegriffen werden, indem ein Fallback zum veralteten GSM erzwungen wird; dadurch kann der Angreifer mit einer gefälschten Basisstation das Anwendungsprotokoll manipulieren.

Folgerichtig steht am Ende jedes Penetrationstests ein ausführlicher Testbericht, der dediziert auch Lösungsmöglichkeiten zum Schließen der gefunden IT-Sicherheitslücken benennt.

Fazit

Die IT-Sicherheit der Fahrzeuge, des Fahrzeugnetzwerkes und seiner Systemkomponenten ist längst zum kritischen Erfolgsfaktor einer vernetzten, hochautomatisierten Mobilität geworden. Zudem wird mit der neuen ISO/SAE 21434 und den UNECE WP.29-Regularien zertifiziertes Cybersecurity-Management sehr bald schon zur Voraussetzung für die Typgenehmigung.

Eine Security-Teststrategie, die die Fahrzeugplattformen vollumfänglich, bis in die Tiefe durchdringt sowie die Lieferkette und den Fahrzeuglebenszyklus mit einbezieht, ist dabei ein zentraler Bestandteil. Das Pentesting wiederum ist hier die „Königsdisziplin": Es liefert die Probe aufs Exempel, indem es das zu testende System sowohl im Detail als auch in seiner Gesamtheit betrachtet – und IT-Sicherheitslücken aufdeckt, bevor ein Angreifer die Gelegenheit dazu hat.

■ (oe)

Dr. Martin Moser ist Head of Consulting & Testing bei ESCRYPT am Standort München.

Dipl.-Math. Thomas Enderle ist Security Consultant im Bereich Security Testing bei ESCRYPT am Standort München.