escrypt
SECURITY. TRUST. SUCCESS.

Whitepaper

Automotive Security auf dem Prüfstand

Security Testing als Nagelprobe für vernetzte Fahrzeuge

Die zunehmende Konnektivität moderner Fahrzeuge birgt aus Security-Sicht neue Risiken. Um vernetzte Systeme gegen unbefugte Zugriffe und Angriffe von Cyberkriminellen abzusichern, verankert der neue Standard ISO/SAE 21434 „Road vehicles – Cybersecurity engineering“ einen Security-by-Design-basierten Ansatz. Dieser sieht ausführliche Tests auf der Komponenten- und Systemebene vor. Damit wird Security Testing zur Nagelprobe für die Security-Maßnahmen und künftiger Bestandteil der Typzulassung. Die Durchführung und Auswertung setzt spezifisches Knowhow voraus. Dieses Whitepaper stellt die wichtigsten Testverfahren vor, weist auf typische Findings und Angriffspunkte hin und zeigt die Notwendigkeit einer holistischen Security-Strategie auf.

Inhaltsübersicht

Security bewegt sich in einem hochdynamischen Umfeld

2021 kam es zu der Veröffentlichung von zwei bedeutenden automobilen Standards. Dies war zuerst im Juni die UNECE WP.29, die auf optimierte Fahrzeugsicherheit und wirksamen Diebstahlschutz abzielt. Im August folgte die ISO/SAE 21434 „Road Vehicles – Cybersecurity Engineering", welche Prozesse und Anforderungen für die Entwicklung sicherer Fahrzeuge definiert. Ein zentraler Treiber beider Regelwerke ist die zunehmende Konnektivität moderner Fahrzeuge: Die Vernetzung mit der Außenwelt steigert nicht nur den Komfort und die Sicherheit, sondern zugleich droht eine erhöhte Angreifbarkeit der vernetzten Systeme. Die technischen Regelwerke bieten Herstellern Leitlinien für eine systematische und wirksame Absicherung der Fahrzeugsysteme gegen unbefugte Zugriffe, Manipulationsversuche und Cyberangriffe.

Doch wie gelingt die Risikominimierung, ohne den Rahmen einer Fahrzeugentwicklung zeitlich und finanziell zu sprengen? Und wie lässt sich die Wirksamkeit der Security-Maßnahmen prüfen? Zumal es auch um Angriffsszenarien und -strategien geht, die zum Zeitpunkt der Fahrzeugentwicklung noch nicht bekannt sind und teils weit in die Zukunft weisen. Lebenszyklen moderner Fahrzeuge können durchaus zwei Jahrzehnte umfassen. Ein Blick zurück verdeutlicht die Dimension: Im Jahr 2000 hatten erst fünf Prozent der Weltbevölkerung Zugang zum Internet, es gab noch keine Smartphones – und erst recht keine vernetzten Fahrzeuge.

Damit wird deutlich: Die Anforderungen an die IT-Security entwickeln sich hochdynamisch – gerade auch im Automotive-Bereich, der seine Fahrzeugnetze für Infotainment, für Updates und Upgrades Over the Air (OTA) oder für die Car-to-X-Kommunikation öffnet. Diese digitale Anbindung macht den Schutz der Systeme zu einer anspruchsvollen Aufgabe. Dies nicht nur, weil ebenfalls vernetzte Täter auf immer leistungsfähigere Rechner und immer neue, häufig im Internet verbreitete Angriffsmuster zugreifen können. Sondern auch, weil die Security organisatorische Herausforderungen bewältigen muss. So können Kryptoverfahren und Algorithmen während der mehrjährigen Entwicklungsphase und erst recht im weiteren Fahrzeuglebenszyklus veralten. Auch die Kommunikation erweist sich in Entwicklungsprozessen, an denen oft international verteilte Teams unterschiedlicher Unternehmen beteiligt sind, als problematisch. Werden Security-Anforderungen unzureichend kommuniziert, dann drohen Probleme bei deren Implementierung und Überprüfung. Die erschwerte Kommunikation hat teils auch zur Folge, dass Systemkomponenten Dritter für die anderen Akteure „Black Boxes" bleiben. Sie wissen mitunter nicht, welche Security-Maßnahmen bereits in den Bauteilen implementiert sind und wie robust diese Implementierungen sind.

Testing wird zur Nagelprobe

Der Blick auf den Standard ISO/SAE 21434 zeigt, mit welchen Methoden ein langfristig wirksames Cybersecurity-Engineering erreicht werden soll. Die Standardisierungsgremien setzen auf Security-by-Design. Analog zur ISO 26262 für funktionale Sicherheit soll auch das Security-Engineering in Zukunft einem V-Modell folgen. Die Arbeiten im linken Schenkel umfassen neben der Definition der Systemanforderungen und dem Erarbeiten des Systemdesigns die daraus abzuleitende Festlegung der Security-Spezifikationen. Darauf folgt im rechten, aufsteigenden Schenkel des Vs vor allem ausführliches Testing. Es beginnt auf der Komponentenebene und geht anschließend mit Tests auf der Systemebene weiter. Erst nach Auswertung dieser Tests und einer abschließenden Abnahme der vernetzten Systeme soll und kann deren operativer Betrieb beginnen.

Das Testing wird so zur entscheidenden Nagelprobe. Seit der Veröffentlichung des neuen Standards steht und fällt die Typzulassung neuer Modelle mit dem Testing. Denn das Befolgen der darin verankerten Methoden ist verpflichtend. Zugleich dient entsprechende Test-Dokumentation den Herstellern im Zulassungsprozess als Nachweis, dass sie die technischen Regelungen eingehalten und den Security-by-Design-Ansatz befolgt haben. Mit dem Security Testing liefern sie den Beleg für die Compliance ihres Entwicklungsprozesses mit den Sicherheitsregularien der Typzulassung.

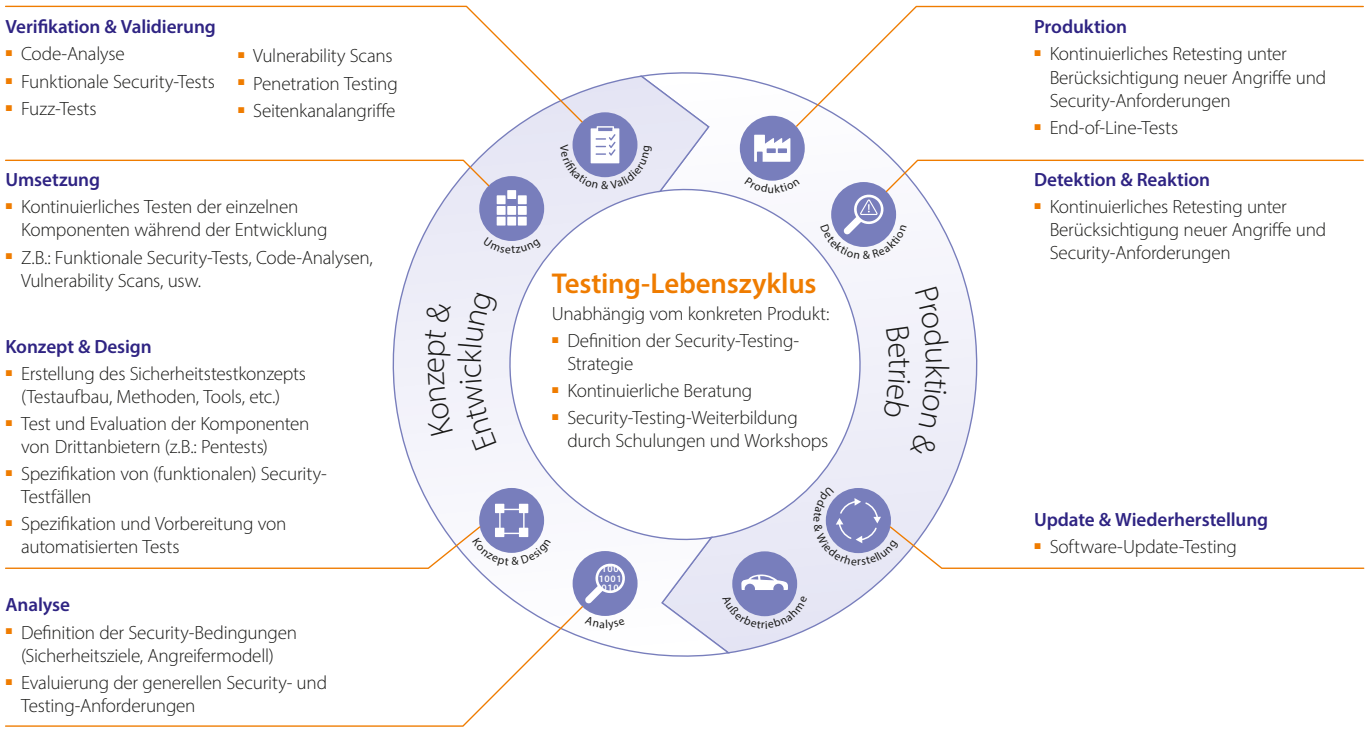

Abbildung 1: Security-Testing-Lebenszyklus

Testing wird zu einer Frage der Organisation

Angesichts dieser Bedeutung und vor dem Hintergrund der organisatorischen Herausforderungen der Security in verteilten Entwicklungsprozessen wirft die konkrete Umsetzung des Security Testings eine Reihe drängender Fragen auf: Welche Tests auf Komponenten- und Systemebene sind erforderlich? Welche Testverfahren sind geeignet? Welcher Kompetenzen bedarf es für die Planung, Durchführung und Auswertung der Tests? Und wie lässt sich eine entsprechende Teststrategie sinnvoll in ein über den Start-of-Production (SoP) hinausweisendes, holistisches Security-Konzept einbetten? Am Anfang dieser Überlegungen steht das Erarbeiten einer fundierten Teststrategie. Denn es gilt, die geeigneten Testverfahren und den jeweiligen Umfang der Tests zu ermitteln. Erst dann lässt es sich sinnvoll abschätzen, für welche Tests zu welchem Zeitpunkt welche Infrastruktur bereitstehen muss. Zugleich muss beantwortet werden, ob Spezialisten mit entsprechendem Security-Test-Knowhow im eigenen Haus verfügbar sind. Und weil Security Testing nicht mit dem SoP endet, muss die Teststrategie den ganzen Fahrzeuglebenszyklus in den Blick nehmen. Es bedarf einer verstetigten Auseinandersetzung mit Angriffsszenarien und -strategien, die in langfristig wirksame Schutzmaßnahmen münden muss.

Schon die grobe Bestandsaufnahme der für die Teststrategie grundlegenden Themen und Aufgaben verdeutlicht, dass dafür spezifische Erfahrungen, strategischer Weitblick und fundierte Analysen der vorhandenen personellen Ressourcen und technischen Infrastrukturen erforderlich sind. Für OEMs und Zulieferer stellt sich die Frage, inwieweit sie den Testaufwand inhouse mit eigenen Experten und eigener Infrastruktur bewältigen können. Andernfalls gilt es, schon im Zuge dieser strategischen Vorarbeiten zu ermitteln, in welchem Umfang und für welche spezifischen Aufgabenstellungen des Security Testings sich externes Knowhow hinzuziehen lässt. Das grundlegende Ziel: Das Fahrzeug und alle mit ihm vernetzten Komponenten so effizient und zuverlässig wie möglich und so lange wie nötig vor unerlaubten Zugriffen und Manipulationen zu schützen. Testing dient der Überprüfung, dass die ergriffenen Schutzmaßnahmen die vernetzen Fahrzeugsteuergeräte, Vehicle Computer und weiteren Komponenten wirksam schützen – und im Vorfeld des SoP alle bis dahin bekannten Sicherheitslücken geschlossen wurden. Um danach auftretende Sicherheitsrisiken zu adressieren, ist ein holistisches Security-Konzept erforderlich, das in einem späteren Abschnitt behandelt wird.

Security-Testing-Methoden

Probe aufs Exempel: Penetration Testing

Eine der gängigsten und wirkungsvollsten Methoden, um die Wirksamkeit der Schutzmaßnahmen zu überprüfen und unentdeckte Sicherheitslücken zu identifizieren, ist das Penetration Testing; geläufig auch unter dem Kurzwort Pentesting.

Pentester nutzen ein breites Arsenal an Angriffsstrategien, um Cyberangriffe zu simulieren und sich unberechtigten Zugang zum System zu verschaffen. Sie betreiben also „Hacking im Auftrag der guten Sache". So decken sie Schwachstellen und etwaige Implementierungsfehler auf, die ihre Ursache in fehlerhafter technischer Umsetzung, Abweichungen vom Security-Konzept, in problematischer Interaktion von Systemkomponenten oder Drittanbieterkomponenten haben können.

Grundsätzlich sind solche Pentests als Whitebox-, Greybox- oder Blackbox-Tests durchführbar. Whitebox-Tests sind im Sinne maximaler Sicherheit die erste Wahl. Hier liegen den Testern die internen Systeminformationen bis hin zum Source Code vollständig vor. Dagegen sind diese Informationen bei Greybox-Tests nur teilweise bekannt. Beim Blackbox-Testing haben die Experten ähnlich wie echte Angreifer lediglich Zugriff auf das zu testende System und öffentlich zugängliche Informationen. Es ist evident, dass mit abnehmender Information die Dauer der Tests steigt, da fehlende Informationen zunächst durch Reverse-Engineering erabeitet werden müssen. Dadurch erhöht sich der Zeitaufwand, die Systematik leidet und die Möglichkeit automatisierter Angriffsmethoden nimmt ab. Auf der anderen Seite ergibt sich aus dem höheren Aufwand beim Blackbox-Test kein Sicherheitsgewinn. Es gilt die Faustregel: Je tiefer der Einblick des Testers in das System, desto eher lassen sich mögliche Sicherheitslücken mit gleichzeitig begrenztem Budget aufdecken.

Der Pentest selbst umfasst einen mehrstufigen Prozess, angefangen beim Setup des Systems über die Analyse und das Erkunden potenzieller Angriffspunkte bis hin zu den eigentlichen Eindringversuchen. Identifizierte Schwachstellen werden mit den Entwicklern erörtert, um sie zu beheben und das System bei Bedarf erneut zu testen. Basierend auf der Gesamtsystementwicklung hat sich ein Vorgehen etabliert, bei dem Pentester typischerweise erst einzelne Steuergeräte, Netzwerkkomponenten, Schnittstellen, Apps und Services auf die Probe stellen, dann Steuergeräte-Verbünde und erst zuletzt die IT-Sicherheit des Gesamtfahrzeugs.

Abbildung 2: Beim Penetrationtest schlüpft der Tester in die Angreiferrolle und versucht, in das Zielsystem einzudringen.

Blackbox 	Greybox 	Whitebox 
Öffentlich zugängliche Informationen	Partielle Spezifikation und Dokumentation	Spezifikationen, Dokumentation, Konfiguration, Quellcode, Shell-Zugriff
Geringe Effizienz Aufwand fließt mehr ins Reverse Engineering als ins Testing ► **Tieferliegende Angriffspfade werden ggf. nicht erkannt**	Mittlere Effizienz	Hohe Effizienz Aufwand fließt ins Testing, nicht ins Reverse Engineering

Abbildung 3: Je mehr über das zu testende System bekannt ist, desto effizienter ist das Testing

Code-Analyse

Beim Code-Audit suchen Security-Tester auf Quellcode-Ebene nach Programmierfehlern oder Sicherheitslücken, die Angreifer nutzen könnten. Besonderes Augenmerk gilt dem korrekten Verhalten von implementierten Security-Maßnahmen sowie Code, der mögliche feindliche Eingaben potenzieller Angreifer verarbeitet – beispielsweise Parser, Krypto-Implementierungen oder Communication Stacks (z.B. für Netzwerk, Funk, User Interface). Zudem lassen sich mit Code-Audits Implementierungsfehler erkennen, wie etwa die unsachgemäße Validierung von Eingaben oder Speicherprobleme (z.B. Pufferüberläufe).

Im Sinne hoher Effizienz und Abdeckung sind automatisierte Code-Audits beispielsweise mithilfe statischer Code-Analyse empfehlenswert. Allerdings liegt der Fokus hierbei eher auf der Robustheit des Codes und der Einhaltung von Best Practices (z.B. MISRA-C, CERT-C), als auf logischen und funktionalen Problemen bei der Implementierung. Im Automotive-Umfeld sind Code-Audits insbesondere für die Bereiche empfehlenswert, die besonders hohe IT-Sicherheitsrisiken beinhalten. Überprüft werden damit häufig Kommunikationsschnittstellen sowie die Security-Funktionen selbst.

Funktionaler Security-Test

Funktionale Security-Tests dienen der Überprüfung, ob die verwendeten Security-Mechanismen korrekt funktionieren und vollständig umgesetzt sind. Auf dem Prüfstand stehen Mechanismen wie kryptografische Verfahren, Secure Boot, Firewalls, Kommunikationsprotokolle, sichere Software Updates und vieles mehr. Dafür definieren die Tester im Vorfeld Positiv- und Negativfälle, um das korrekte Verhalten der Funktion zu überprüfen. Die anschließende Durchführung der Tests erfolgt in der Regel automatisiert. Zentrale Voraussetzung: In die Definition der Testfälle muss ein hohes Maß an Security-Knowhow einfließen, um Angriffsszenarien realistisch durchspielen zu können. Knowhow ist auch gefragt, um die korrekte Integration auf die Zielplattform zu validieren. Denn diese verhält sich oft anders als das Entwicklungssystem. Die Integrationstests im Fahrzeugumfeld sind anspruchsvoll. So werden bei typischen Busprotokollen wie CAN mitunter keine direkten Antwortnachrichten verschickt, wodurch schwer nachvollziehbar ist, ob die Testnachrichten korrekt verarbeitet werden. Oft müssen daher gleichzeitig mehrere Signale in bestimmter Reihenfolge auf verschiedenen Fahrzeugbussen erstellt und überwacht werden, um beispielsweise die Wirksamkeit von Security-Protokollen oder Gateway-Funktionen zu überprüfen.

Fuzz-Test

Fuzzing ist ein leistungsstarkes Testverfahren, um die Robustheit getesteter Systeme zu prüfen. Bei Fuzzern handelt es sich um Testsoftware, mit deren Hilfe eine hohe Zahl untypischer oder ungültiger Eingaben generiert wird. So lassen sich nicht nur verschiedenste interne Systemzustände im Zeitraffer durchlaufen – sondern auch Fehlverhalten, Anomalien und nicht vorgesehene Informationspreisgaben provozieren, die Angreifer für einen Cyber-Angriff nutzen könnten. Gute Fuzzer stellen sich auf das vom Zielsystem verwendete Protokoll ein und modifizieren nur einzelne Aspekte des Protokolls, wie etwa einzelne Datenfelder oder Komponenten des Datenflusses. Dieses „Smart Fuzzing" ist besonders effizient, weil es sich auf die feine Linie zwischen ungültiger und gültiger Eingabe konzentriert (siehe Abb. 4).

Bei Fuzz-Tests ist umfassendes Monitoring gefragt, um Fehlverhalten des Testobjekts erkennen zu können. Das ist keineswegs trivial, weil dessen Reaktionen nicht immer nur auf der zu testenden Schnittstelle erkennbar sind, sondern sich mitunter nur im Verhalten des getesteten Objekts oder über Ausgaben auf anderen Schnittstellen zeigen.

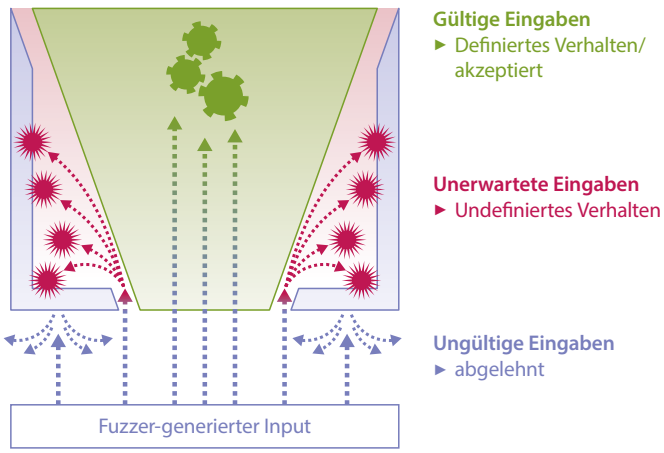

Abbildung 4: Fuzzing zielt auf die feine Linie zwischen ungültiger und gültiger Eingabe

Gute Fuzzing-Tools decken mittlerweile nahezu alle Automotive-relevanten Protokolle ab: CAN, ISO-TP, UDS, USB, Bluetooth, WiFi und viele Ethernet-basierte Protokolle (IP, TCP, UDP, FTP, TLS etc.).

Das Verfahren gilt als hocheffektives Werkzeug, um verschiedenste Schwachstellen und ihre Folgen aufzudecken – von subtilen Bugs wie abweichendem Zeitverhalten oder Speicherlecks bis zu dem Komplettabsturz des Systems.

Vulnerability Scan

Beim Vulnerability Scanning werden die Zielsysteme auf bekannte Verwundbarkeiten, Einfallstore und Sicherheitslücken hin untersucht. In aller Regel wird dabei auf eine Datenbank zurückgegriffen, in der die aktuell für das Testobjekt bekannten Schwachstellen hinterlegt sind. Mit dem Input der Datenbank „tastet" der Scanner das System ab und scannt im Steuergeräteumfeld beispielsweise das Unified Diagnostic Services-Protokoll (UDS) nach typischen Schwachstellen, darunter zu kurze Seed-Werte oder schwache Schlüssel-Berechnungsalgorithmen. Die Qualität solcher Vulnerability Scans hängt insbesondere vom Umfang und der Aktualität der genutzten Datenbank ab. Während es für klassische IT-Verwundbarkeiten zahlreiche umfangreiche und aktuelle Sammlungen gibt (z.B. die Mitre-CVE-Datenbank), befinden sich Automotive-Security-spezifische Datenbanken mit den hier relevanten Protokollen und Technologien meist noch im Aufbau. Ihr Nutzen steht und fällt mit der stetigen Aktualisierung. Zusätzlich erschwert wird das Vulnerability Scanning durch die geringe Standardisierung im Automotive-Bereich. Aus den genannten Gründen ist für dieses Verfahren ein hohes Maß an Spezialisierung und an Ressourcen für die Datenbankpflege erforderlich.

Seitenkanalangriffe

Bei Seitenkanalangriffen handelt es sich um ein Verfahren, bei dem Angriffe auf Komponenten der physischen Implementierung des Systems ausgeführt werden. Wir unterscheiden hier zwischen passiven und aktiven Seitenkanalangriffen.

Beim passiven Seitenkanalangriff (auch Seitenkanalanalyse) wird versucht über die Messung der sogenannten Seitenkanäle, d.h. physikalischer Eigenschaften des Zielsystems (z.B. Zeitverhalten, Stromverbrauch, elektromagnetische Abstrahlung) Rückschlüsse auf die interne Datenverarbeitung zu ziehen. So lässt sich beispielsweise anhand der Schwankungen in der Stromversorgung eines Mikroprozessors bei der Verarbeitung eines kryptographischen Algorithmus auf die verwendeten Schlüssel schließen.

Beim aktiven Seitenkanalangriff geht es darum, das System gezielt zu manipulieren. Ein typisches Beispiel: Fault-Injection-Angriffe, bei denen Tester versuchen, durch kurzfristige Unterbrechung der Stromversorgung oder gezielte elektromagnetische Störimpulse Verarbeitungsfehler eines Mikroprozessors zu provozieren – und so das von einem potenziellen Angreifer gewünschte Verhalten auszulösen.

Spezialisierung ist das A und O im Security Testing

Der Überblick über die gängigen Testverfahren zeigt einerseits, dass automatisierte und manuelle Ansätze in vielen Fällen unmittelbar ineinandergreifen. Dabei haben automatisierte Tests den Vorteil schneller Durchführbarkeit, hoher Reproduzierbarkeit, guter Integrierbarkeit in Test-Routinen – und nicht zuletzt hoher Kosteneffizienz. Dies gilt erst recht, wenn sie Schwachstellen bereits in der Frühphase des Entwicklungsprozesses identifizieren. Je früher Sicherheitslücken auffallen, desto einfacher und kostengünstiger lassen sie sich schließen. Dieser Vorteil lässt sich durch Security Testing in-the-Loop (Security XiL) nutzen. Dabei erfolgen automatisierte Security-Tests (Funktionale Security, Fuzzing und Vulnerability Scans) bereits während der Steuergeräteentwicklung, um etwaige Sicherheitslücken zu identifizieren.

Doch – und hier folgt das Andererseits – trotz der Automatisierungsmöglichkeiten bleibt das Security Testing eine Aufgabe für Spezialisten. Schon im Kurzüberblick der Testverfahren ist ersichtlich, dass Planung, Durchführung und Monitoring der unterschiedlichen Tests umfassendes Knowhow, genaue Kenntnis der Angriffsstrategien und in vielen Fällen auch spezifische Infrastrukturen erforderlich machen; seien es stetig aktualisierte Datenbanken oder Testequipment für virtualisiertes Security-XiL-Testing. Auch ein weiterer Aspekt spricht für das Hinzuziehen von Security-Spezialisten: Automatisierte Tests finden zwar schnell bekannte Schwachstellen in geläufigen Services und Protokollen, doch sobald es individuelle kundenspezifische Anpassungen gibt, stoßen sie schnell an Grenzen. Letztere sind gerade im Automotive-Bereich keine Seltenheit. Zudem sind automatisierte Testing-Tools nicht dazu in der Lage, logische Fehler in getestetem Programmcode oder auch die Verkettung mehrerer, alleine betrachtet harmloser Schwachstellen zu komplexen Angriffspfaden zu erkennen. Daher bedarf es in Ergänzung der automatisierten Methoden manuelle Tests durch Security-Experten. Dies gilt auch dann, wenn der Zugang zu Informationen über Systemkomponenten und Quellcode aus juristischen oder anderen Gründen versperrt ist – und die Tester zum „Blackbox-Testing" gezwungen sind.

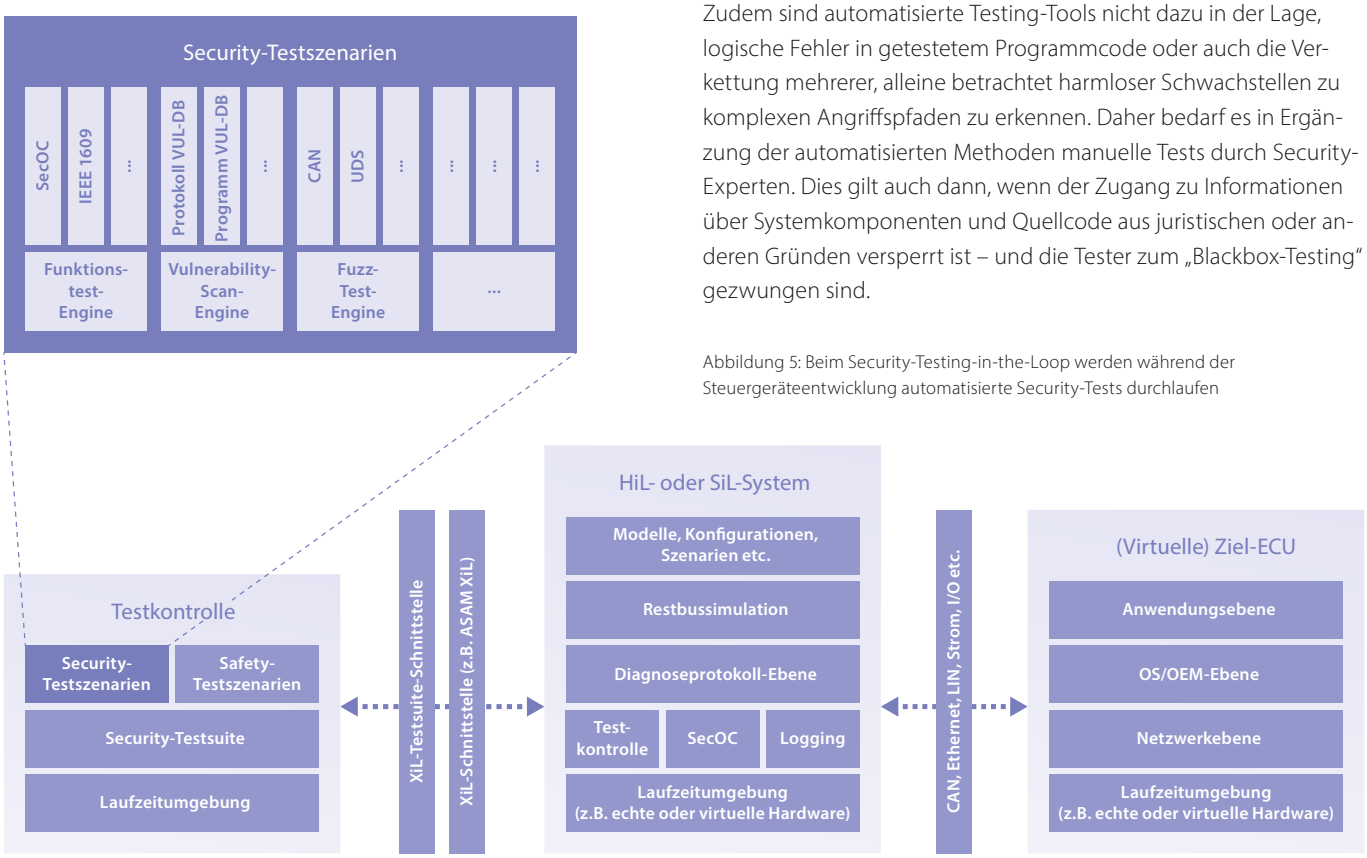

Abbildung 5: Beim Security-Testing-in-the-Loop werden während der Steuergeräteentwicklung automatisierte Security-Tests durchlaufen

Zusammenspiel individuell angepasster Testing Tools und manueller Pentest-Verfahren

Es bedarf also neben der automatisierten Prüfung mit grobem Korn sehr gezielter manueller Tests, um tatsächlich alle Testfälle abdecken zu können. Dabei zahlt sich der Rückgriff auf Testing-Tools aus, die sich individuell an die Spezifikationen der Fahrzeugplattformen von bestimmten OEMs oder sogar auf einzelne Steuergeräte anpassen lassen. Beispielsweise verwenden Security-Tester von ESCRYPT spezifisch auf die Anforderungen bestimmter Hersteller ausgelegte Fuzz Testing Tools und gelangen auf diese Weise sehr viel schneller zu sehr viel genaueren Testresultaten. Anhand der Auffälligkeiten, die dabei zu Tage treten, beginnt dann die eigentliche Analyse. Hierfür bringen Security-Experten ihr ganzes Knowhow und ihre gesammelten Erfahrungen ein, um die Befunde in den zugrundeliegenden Daten, im Datenverkehr, den untersuchten Prozessen oder im Quellcode zu deuten. Im Team können sie Datenbanken nach Angriffen auf ähnliche Ziele durchforsten, Vergleiche anstellen und optimierte Schutzstrategien entwickeln. Auch sind erfahrene Spezialisten in der Lage, in kleinen Fehlern, die für sich genommen harmlos erscheinen, komplexe potenziell gefährliche Angriffsketten zu erkennen. So wertvoll der Einsatz geeigneter Testing-Tools auch ist – letztlich liefert erst die Kombination mit manuellen Tests durch Security-Experten jenes umfassende Bild der IT-Sicherheit, das für eine wirksame Absicherung der Systeme gegen gezielte und elaborierte Cyberattacken unverzichtbar ist.

Die Testexperten gehen dabei in enger Kooperation mit den Herstellern iterativ vor. Zunächst identifizieren sie mögliche Angriffspunkte und Schwachstellen. Anschließend konfrontieren sie die Hersteller mit diesen Sicherheitslücken und den dadurch ausgelösten Risiken und erläutern ihnen mögliche Lösungen und wirksame Gegenmaßnahmen. Das Spektrum reicht von Implementierungsfehlern, die mit Korrekturen in einzelnen Codezeilen behoben werden können, bis zu massiven IT-Sicherheitsmängeln, die im Worst Case eine vollständige Neukonzeption des getesteten Systems erfordern. Sobald die bestehenden Lücken geschlossen sind erfolgen Re-Tests, um den Erfolg der Maßnahmen zu prüfen und um zusätzlichen Korrekturbedarf zu identifizieren. Die Iterationsschleifen setzen sich fort, bis der vereinbarte Testumfang abgearbeitet ist. Dabei können die Tests sowohl auf Komponentenebene als auch auf Systemebene erfolgen.

Typische Pentest Findings

Tatsächlich finden sich in Entwicklungsprojekten regelmäßig Schwachstellen, die Angreifer potentiell für Manipulationen und Cyberangriffe nutzen könnten. Obgleich das Pentesting von Steuergeräten, Infotainmentsystemen oder Steuergeräte-Verbunden stets einer individuell angepassten Strategie folgen, nehmen erfahrene Automotive Pentester zunächst die typischen Angriffspunkte und Findings in den Fokus:

- **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 Sofware:** 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 (z.B. Buffer Overflow) für einen Angriff ausnutzen. Dazu kann es sein, 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. Noch weitreichendere Folgen hat es, wenn ein solcher RNG für kryptografische Zwecke verwendet wird; im Extremfall kann ein Angreifer dann geheime Schlüssel errechnen.

- **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. Beispielsweise können dann eine Computer-Tastatur oder ein zusätzlicher Netzwerkadapter an einem Steuergerät verwendet werden.

- **Bluetooth-Probleme:** Aufgrund der historisch gewachsenen Komplexität des Bluetooth-Protokolls gibt es viele Angriffspunkte, beispielweise eine Verwendung von statischen Default-PINs wie „1234", oder den "BlueSmack"-Denial-of-Service-Angriff. Bei letzterem wird das Zielsystem durch große L2CAP-Echo-Request-Pakete zum Absturz gebracht und benötigt einen Power-Cycle, um seine Arbeit fortzusetzen. Ein solcher Angriff kann ohne großen Hard- und Softwareaufwand ausgeführt werden und zu gefährlichen Verkehrssituationen führen.

- **Verwendung unsicherer Protokolle:** Bei Protokollen mit veralteten Security-Mechanismen lassen sich durch das Brechen des Verschlüsselungsalgorithmus Inhalte der Nachrichten lesen oder manipulieren.

- **Mobilfunkverbindungen** können angegriffen werden, indem ein Fallback zum veralteten GSM erzwungen wird; dadurch kann der Angreifer mit einer gefälschten Basisstation das Anwendungsprotokoll manipulieren. Ist die Anwendung dann nicht selbst abgesichert, lassen sich vorgesehene Fernbedienungsfunktionen (z.B. Türöffnung) missbrauchen, oder es kann über weitere Lücken Schadsoftware aufgespielt werden.

Als Resultat des durchgeführten Tests halten Kunden einen ausführlichen Testbericht in den Händen, der nicht nur die identifizierten Angriffspunkte, die angewandten Testverfahren sowie alle Findings im Detail auflistet, sondern auch die jeweiligen Lösungsmöglichkeiten zum Schließen der gefunden IT-Sicherheitslücken benennt. Diese Testberichte wurden durch die Veröffentlichung des Standards ISO/SAE 21434 zur unverzichtbaren Voraussetzung für die Typzulassung.

Security Testing als Teil einer holistischen Sicherheitsphilosophie

Schon mehrfach ist angeklungen, dass Security Testing nicht mit der Typzulassung und dem SoP endet. Vielmehr brauchen vernetzte Fahrzeuge über ihren Gesamtlebenszyklus hinweg holistischen Schutz. ESCRYPT bietet entsprechende Lösungen, die von der Absicherung der Entwicklungs- und Fertigungsprozesse über hardwarebasierte Security im Fahrzeug, kryptographische Lösungen bis hin zu einer intelligenten aktiven Angriffserkennung und Angriffsabwehr (Intrusion Detection and Prevention Systems, IDPS) für Fahrzeuge und Fahrzeugflotten im Feld reicht. Ein IDPS ermöglicht es, sämtliche erkannten Anomalien aller Fahrzeuge im Feld in einer zentralen cloudbasierten Event-Datenbank zu sammeln, auszuwerten und umgehend Gegenmaßnahmen einzuleiten, wenn die Befunde auf neue Risiken oder Angriffsmuster hinweisen. Mit jedem Fahrzeug, das dem Verbund angeschlossen ist, wird das IDPS intelligenter und abwehrfähiger. Bisher unsichtbare und von Firewalls geblockte Angriffe fließen in eine ständige Lageauswertung ein, wodurch sich die Security-Maßnahmen auch nach dem SoP jederzeit schnell und gezielt an die aktuelle Risikolage anpassen lassen. Das vernetzte Fahrzeug erhält so ein Immunsystem, dessen Abwehrkräfte durch jeden Angriffsversuch gestärkt werden – und das die IT-Sicherheit von modernen Fahrzeugen bis ans Ende ihrer Lebenszyklen gewährleisten kann.

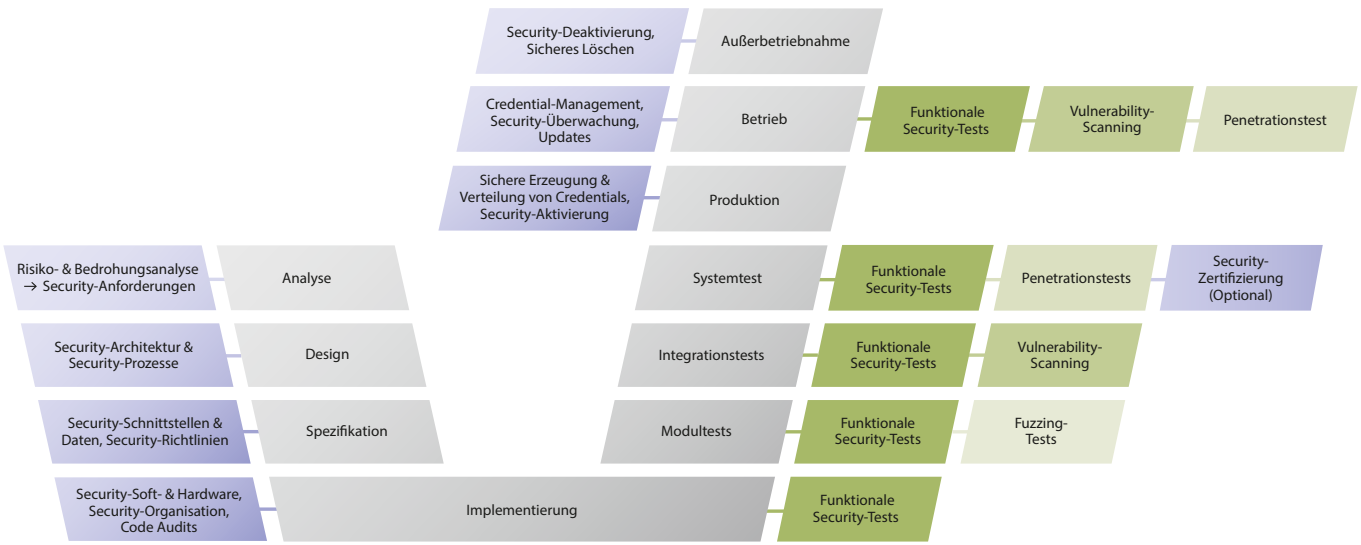

Abbildung 6: Security Testing während des gesamten Produktlebens

Fazit

Vernetzte Fahrzeuge brauchen einen systematischen Security-Rundumschutz, der von der Wiege bis zur Bahre reicht. Nicht zuletzt die Veröffentlichung des Standards ISO/SAE 21434 „Road Vehicles – Cybersecurity Engineering" erfordert fundierte und methodische Ansätze für ein wirklich umfassendes Security Testing auf der Komponenten- und Systemebene. Diese Tests und deren Dokumentation werden zur Voraussetzung der Typzulassung neuer Fahrzeugmodelle. Automatisierte und manuelle Testverfahren bieten im Zusammenspiel die Möglichkeit, Schwachstellen und mögliche Angriffspunkte auf Fahrzeugnetzwerke systematisch und frühzeitig zu schließen. Umsetzbar ist dieses umfassende Testing nur mit spezifischem Knowhow und entsprechender technischer Infrastruktur. Für Hersteller und Zulieferer stellt sich nun die Herausforderung Teststrategien zu erarbeiten, die personellen und technischen Voraussetzungen für die Tests zu schaffen – und diese in umfassendere holistische Schutzmaßnahmen für ihre zunehmend vernetzten Fahrzeuge einzubetten. ESCRYPT hat den Trend frühzeitig erkannt – und in den letzten Jahren entsprechendes Knowhow und ein Lösungsportfolio aufgebaut, mit dem sich IT-Sicherheit hochwirksam und langfristig sicherstellen lässt.

Autoren & Kontakt

Dr. Martin Moser
Head of Consulting & Testing Munich
Lead Service Owner Security Testing
Martin.Moser@escrypt.com
+49 (89) 35 64 78-191

M.Sc. Tobias Brennich
Security Consultant
Tobias.Brennich@escrypt.com
+49 (89) 35 64 78-166

Dipl.-Math. Thomas Enderle
Lead Security Specialist
Thomas.Enderle@escrypt.com
+49 (89) 35 64 78-133

ESCRYPT GmbH
Ridlerstraße 57
80339 München
Germany

www.escrypt.com

Stand: 1/2022